DESIGN AND IMPLEMENTATION OF A MICROCOMPUTER, CONTROLLER FOR STABILIZING AN INVERTED PENDULUM MOUNTED ON A CART

by

PARAG BRIJENDRA BHARGAVA

B.E., Bombay University, India, 1980

A THESIS

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1983

Approved by:

Major Professor

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

FIGURES PAGE

CHAPTER I

INTRODUCTION

In this advanced technological age there are many useful systems which have a natural instability; for example missiles, helicopters, aircrafts, torpedos, etc. . To stabilize these systems, for useful and safe operation, control systems are designed and built to suit their individual requirements.

In recent years, especially since the advent of the microcomputer, great advances have been made in the field of discrete control systems. These systems differ from conventional, continuous systems in that some of the signals are in the form of either discrete values or numerical codes.

With the development of sophisticated, fast, inexpensive computers and interfacing hardware, an impetus is given to the study and development of discrete control systems. The extensive arithmetical and logical capabilities of microcomputers make them suitable for executing complex and extensive control algorithms. Hence digital computers have gained significant importance as control elements in control systems.

The object of this thesis is to develop and implement a discrete control system for a naturally unstable system. An inverted pendulum mounted on a cart was chosen for this study. Controlling this system is similar to the problem of controlling the attitude of a rocket during the initial stages of the launch.

A d.c. motor was used to drive the cart. Power was transmitted from the d.c. motor to the wheels of the cart by a belt and pulley drive system. The movement of the cart is limited to the X - direction and the displacement of the pendulum is restricted in the vertical plane along the X - direction.

This thesis is divided into eight major parts.

To know the dynamics of the system to be controlled the mathematical model of the system is developed in Chapter II. The paramaters of the model were obtained by frequency response analysis.

The discrete data model was obtained by applying the Z-transform. The Z-transform analysis for the system is given in Chapter III.

The physical system has one mode of instability and one mode of neutral stability. The system is stabilized by a compensation technique. The compensation analysis is presented in Chapter IV.

The procedure and the results of a simulation of the system are given in Chapter V.

The system was controlled by a microcomputer. The program was written in assembly language. Emphasis was placed on making the program as efficient as possible due to the time constraint for the execution of the computation. The main program and its' subroutines are explained in Chapter VI. The programs are listed in Appendix E.

The experimental arrangement and procedure are given in Chapter VII.

The results of the experiment are given in Chapter VIII.

The conclusions and recommendations are presented in Chapter IX.

CHAPTER II

MATHEMATICAL MODELING

2.1 Introduction

The details of the system can be visualised from photographic Figure 2.1.

A d.c. motor is used to drive the cart which has seperate excitations for the armature and the field. The armature voltage was varied to control the speed of the cart while a constant voltage was applied to the field. Power from the d.c. motor was transmitted to the wheels of the cart by belt drives with a gear ratio of 19.7:1. The position of the pendulum was measured by a potentiometer which was mounted on the pivot of the pendulum. The speed of the cart was measured by a tachometer which was coupled to the shaft of the d.c. motor.

In this chapter a mathematical model of the system is developed. To obtain a mathematical model it is necessary to analyze the relationship between the system variables. Since the system is dynamic, the descriptive equations are differential equations. Necessary assumptions were made to linearize the system, in order to simplify the analysis. The parameters of the system were determined by frequency response analysis.

2.2 Mathematical Model of the D.C. Motor, Drive and Cart.

A schematic representation of the d.c. motor and the drive system is shown in Figure 2.2. An armature controlled d.c. motor

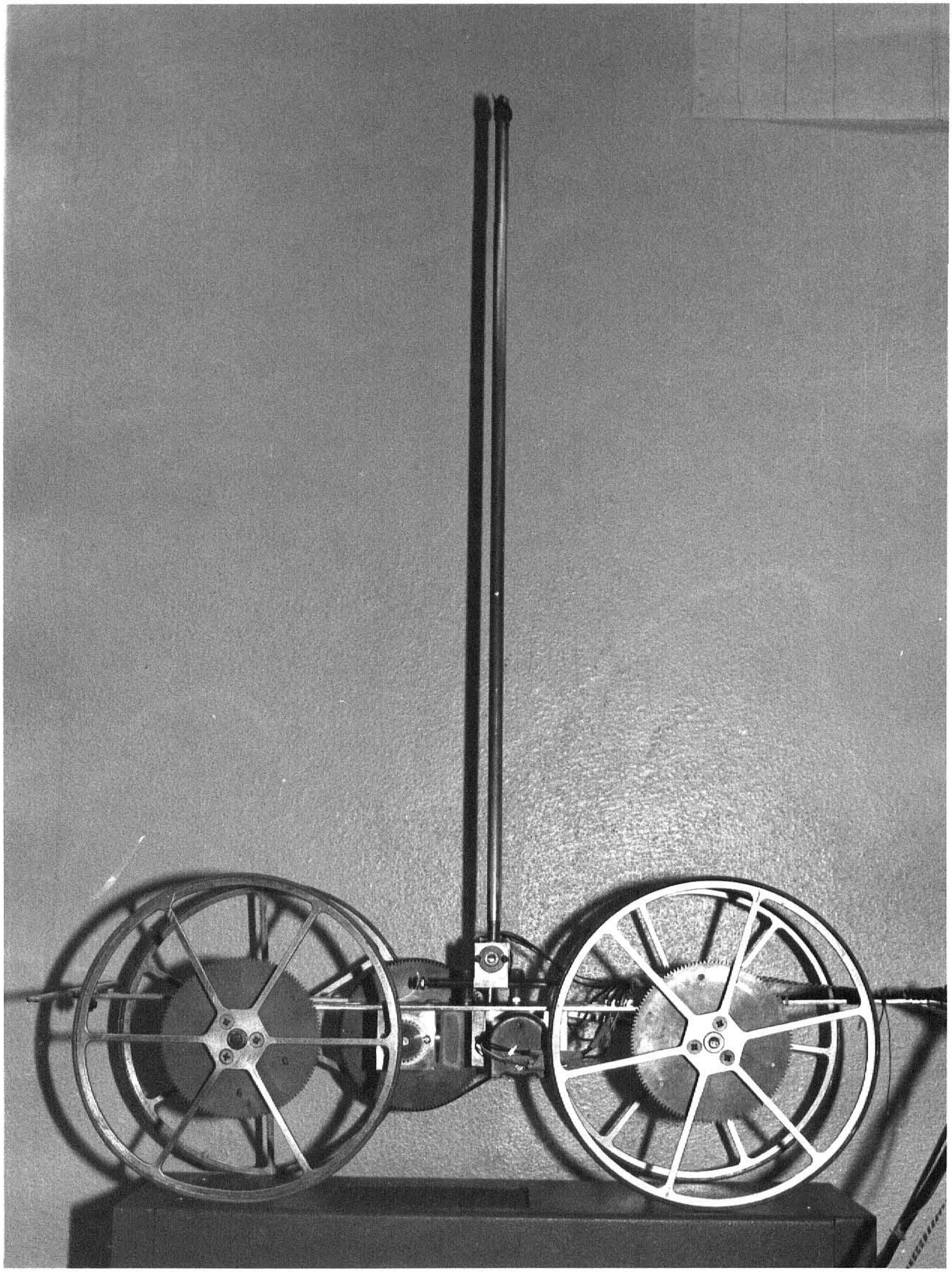

Figure 2.1 Photograph of the Cart Pendulum System

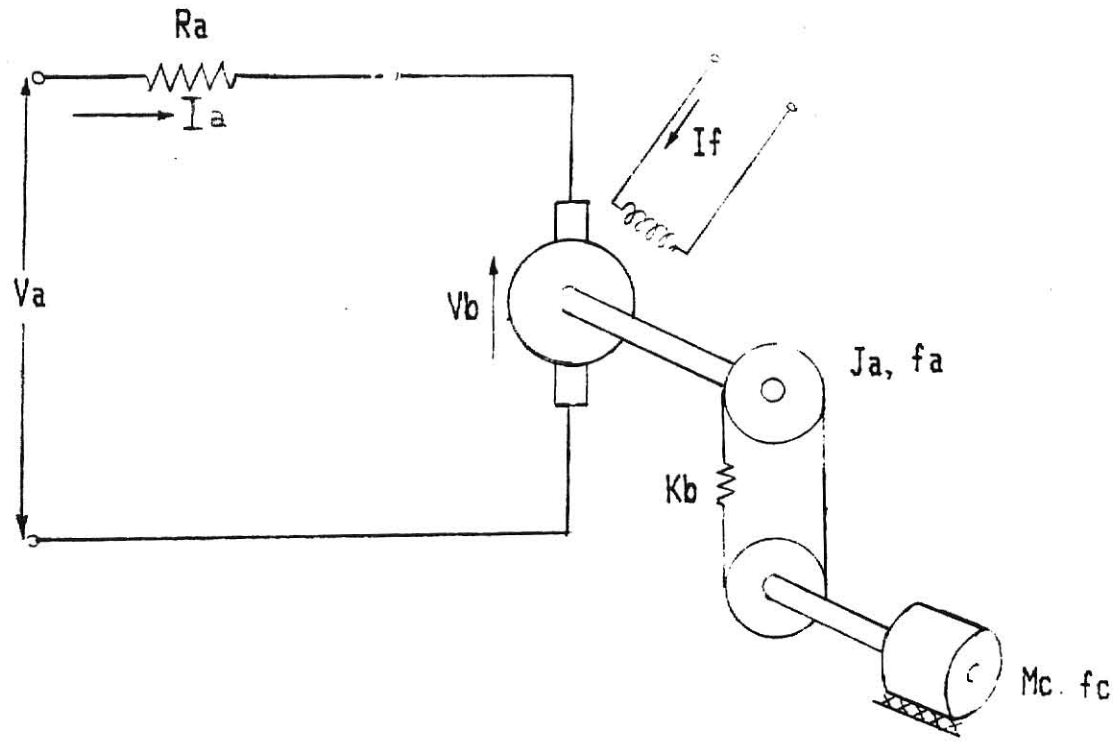

Figure 2.2. Schematic Representation of the D.C. Motor, Drive and Cart

is used to drive the cart. The transfer function of the d.c motor is developed. Second order effects, such as hystersis, inductance of the armature and the voltage drop across the brushes are neglected. The spring constant, Kb, of the belt drive is considered while the damping coefficient of the belt is neglected because it is very small.

The input voltage, Va, is applied across the armature terminals. The back voltage, Vb, is proportional to the speed of the armature.

$$Vb(s) = Ks*Wa(s), \tag{2.1}$$

where Ks is the motor speed constant (volts/radian/sec), and

Wa is the speed of the armature (radians/sec).

The armature current , Ia, is related to the input voltage applied to the armature by

$$Ia(s) = \left[Va(s) - Vb(s)\right]/Ra, \tag{2.2}$$

where Ra is the resistance of the armature.

The torque, Tm, produced by the motor is proportional to the current through the armature. The motor torque is given by the relation,

$$Tm(s) = Km*Ia(s), \tag{2.3}$$

where Km is the motor torque constant (Newton-m/ampere).

The motor torque is equal to the torque delivered to the load. Applying D'Alembert's principle to the armature shaft rotation yields the equation,

$$Ja*\dot{W}a + fa*Wa + Kb*(Da - Kw*Dc) = Tm, \quad (2.4)$$

where Ja = moment of inertia of the armature ($Kgm\text{-}m^2$),

fa = coefficient of friction (Newton-m-sec),

Kb = spring constant for the belt (Newton-m),

Da = displacement of the armature (radian),

Dc = displacement of the cart (m),

Wc = speed of the cart (meters/sec), and

Kw = transmission ratio (radians/m).

The Laplace transform of Equation 2.4 yields,

$$(Ja*s^2 + fa*s + Kb)*Wa = Tm*s + Kb*Kw*Wc. \quad (2.5)$$

Similarly, applying D'Alembert's principle to the displacement of the cart yields,

$$Mc*\dot{W}c + fc*Wc + Kb*Kw^2*(Dc - Da/Kw) = 0. \quad (2.6)$$

where Mc = mass of the cart (Kgm) and

fc = coefficient of friction (Newton/m/sec).

The Laplace transform of Equation 2.6 yields,

$$(Mc*s^2 + fc*s + Kb*Kw^2)*Wc = Kw*Kb*Wa . \quad (2.7)$$

Rearranging the terms in Equation 2.7 yields the transfer function relating the speed of the cart and the armature rotation as,

$$\frac{Wc(s)}{Wa(s)} = \frac{Kw*Kb}{Mc*s^2 + fc*s + Kb*Kw^2}. \quad (2.8)$$

Figure 2.3 is a block diagram representation of the elements of the d.c. motor, drive and the cart. Applying block diagram reduction techniques the reduced block diagram is shown in Figure 2.4. The transfer function relating the motor speed to the armature voltage is,

$$\frac{Wa(s)}{Va(s)} = \frac{Km*(Mc*s^2 + fc*s + Kb*Kw^2)}{Ra*(D3*s^3 + D2*s^2 + D1*s + D0)} , \quad (2.9)$$

where D3 = Ja*Mc ,

D2 = Mc*fa + Ja*fc + Mc*Km*Ks/Ra ,

D1 = $Ja*Kb*Kw^2$ + fa*fc + Mc*Kb + fc*Km*Ks/Ra, and

D0 = Kb*fc + $Kb*fa*Kw^2$ + $Kb*Km*Ks*Kw^2/Ra$

The coefficients of the transfer function are determined by frequency response analysis, since it is a very difficult to measure the individual parameters accurately.

2.3 Mathematical Model for the Inverted Pendulum

The inverted pendulum mounted on the cart is illustrated in Figure 2.5. The differential equation describing the motion of the pendulum is obtained by summing the moments about the pivot point . In order to make the equation linear, the assumption made is that the angle of rotation, P, of the pendulum is small. The sum of the moments about the pivot point is,

$$m*l_1*\dot{W}c + m*l_2^2*\ddot{P} - m*l_1*g*P=0 \quad (2.10)$$

where m = mass of the pendulum,

l_1 = length of the center of mass of the pendulum from the pivot point ,

l_2 = radius of gyration of the pendulum from the pivot

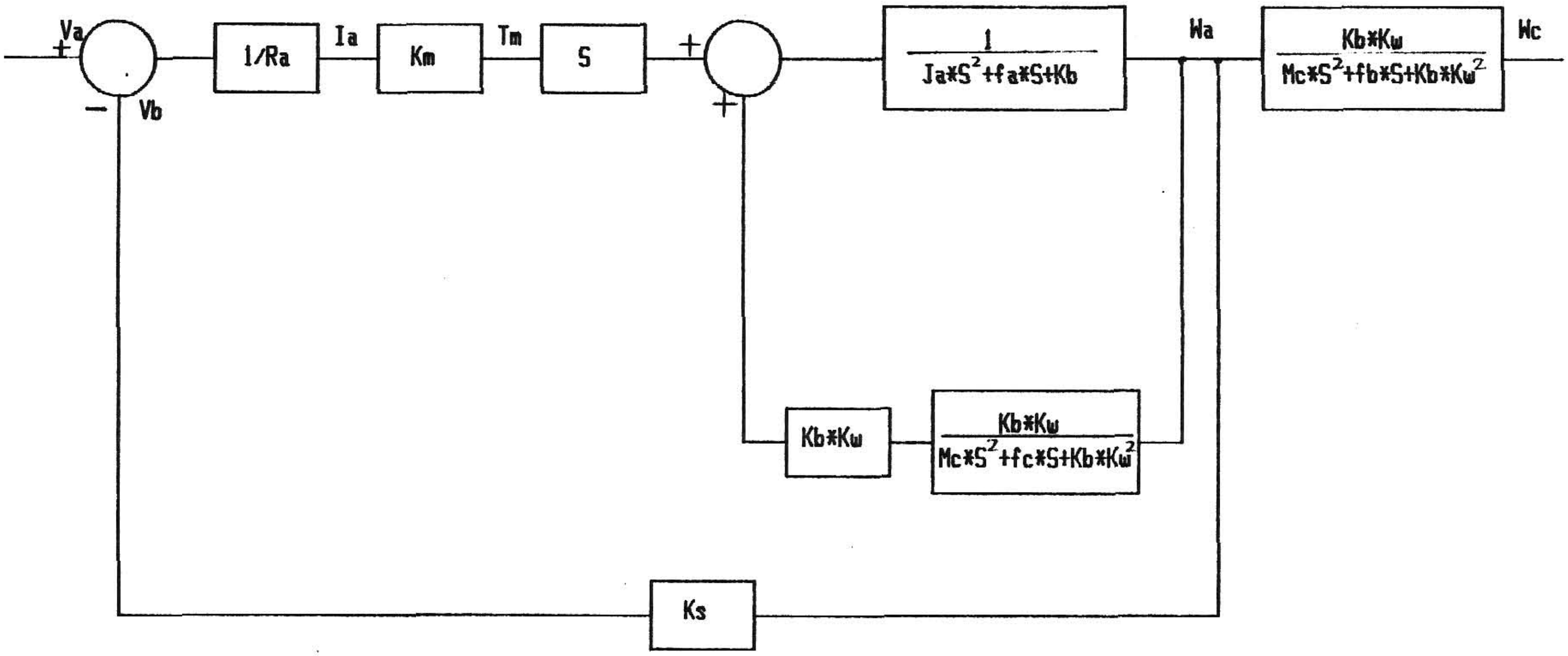

Figure 2.3. Block Diagram Representation of the D.C. Motor, Drive and Cart

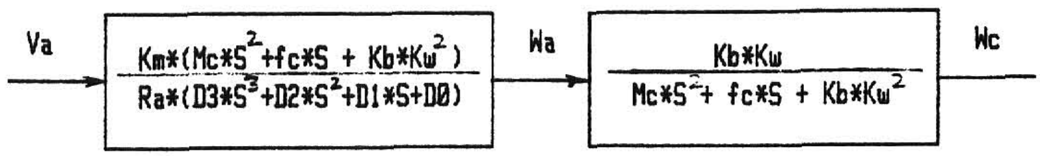

Figure 2.4. Reduced Block Diagram of the D.C. Motor, Drive and Cart

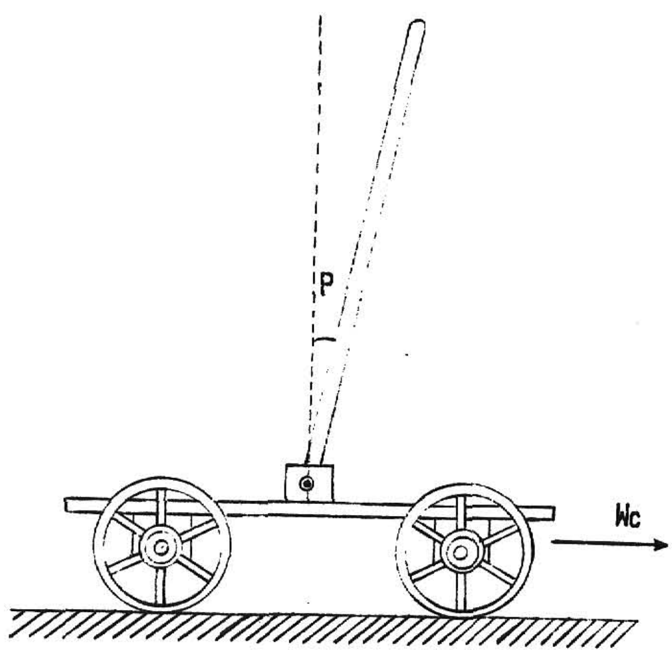

Figure 2.5. A Cart and Inverted Pendulum

point, and

g = acceleration due to gravity = 9.81 m/sec^2 .

A uniform pendulum of length , L= 0.59 m, was chosen for the system. Hence,

l_1 = L/2 and

l_2 = $L/\sqrt{3}$.

Substituting the values of l_1 and l_2 in Equation 2.10 yields,

$$\dot{W}c + 2*L*\ddot{P}/3 - g*P = 0 \tag{2.11}$$

or

$$\dot{W}c + 0.40*\ddot{P} - 9.81*P = 0 \quad . \tag{2.12}$$

The Laplace transform of Equation 2.12 is

$$\frac{P(s)}{Wc(s)} = \frac{-2.5*s}{s^2 - 25} \quad . \tag{2.13}$$

Equation 2.13 is the transfer function relating the dynamics of the pendulum to the speed of the cart.

The d.c. motor used to drive the cart has seperate excitations for the armature and the field. The armature voltage is used for control of the speed of the cart and a constant voltage is applied to the field. Since the dynamic characteristics of the motor are affected by the value of the field voltage it is important to establish the value of the field voltage before carrying out the frequency response analysis to determine the coefficients of the transfer functions. An experiment to explore the effects of the value of the field voltage is described in the next section.

2.4 Dynamic Response of the Speed of the Motor due to Change of Voltage Across the Field Winding

In this experiment the cart is driven by applying a sinusoidal voltage to the armature of the d.c. motor. The field voltage is increased in increments of 1 volt for each succesive observation. The voltage to the armature is a sinusoid with an amplitude of 5 volts peak to peak and a frequency of 1.2 radians/sec. The output from the tachometer is recorded on a strip chart recorder. The values of the tachometer signal amplitude and the frequency response amplitude ratio are given in Table 2.1 and plotted in Figure 2.6.

Observation Number	Field Voltage	Amplitude of Tachometer Signal in Volts	Amplitude Ratio
1)	6	2.61	0.52
2)	7	3.53	0.71
3)	8	4.75	0.95
4)	9	5.32	1.06
5)	10	5.70	1.14
6)	11	6.27	1.25
7)	12	6.67	1.33
8)	13	6.89	1.38
9)	14	7.03	1.41
10)	15	7.03	1.41
11)	16	7.03	1.41

Table 2.1 Experimental Reading of the Amplitude Ratio of the D.C. Motor Versus Field Voltage.

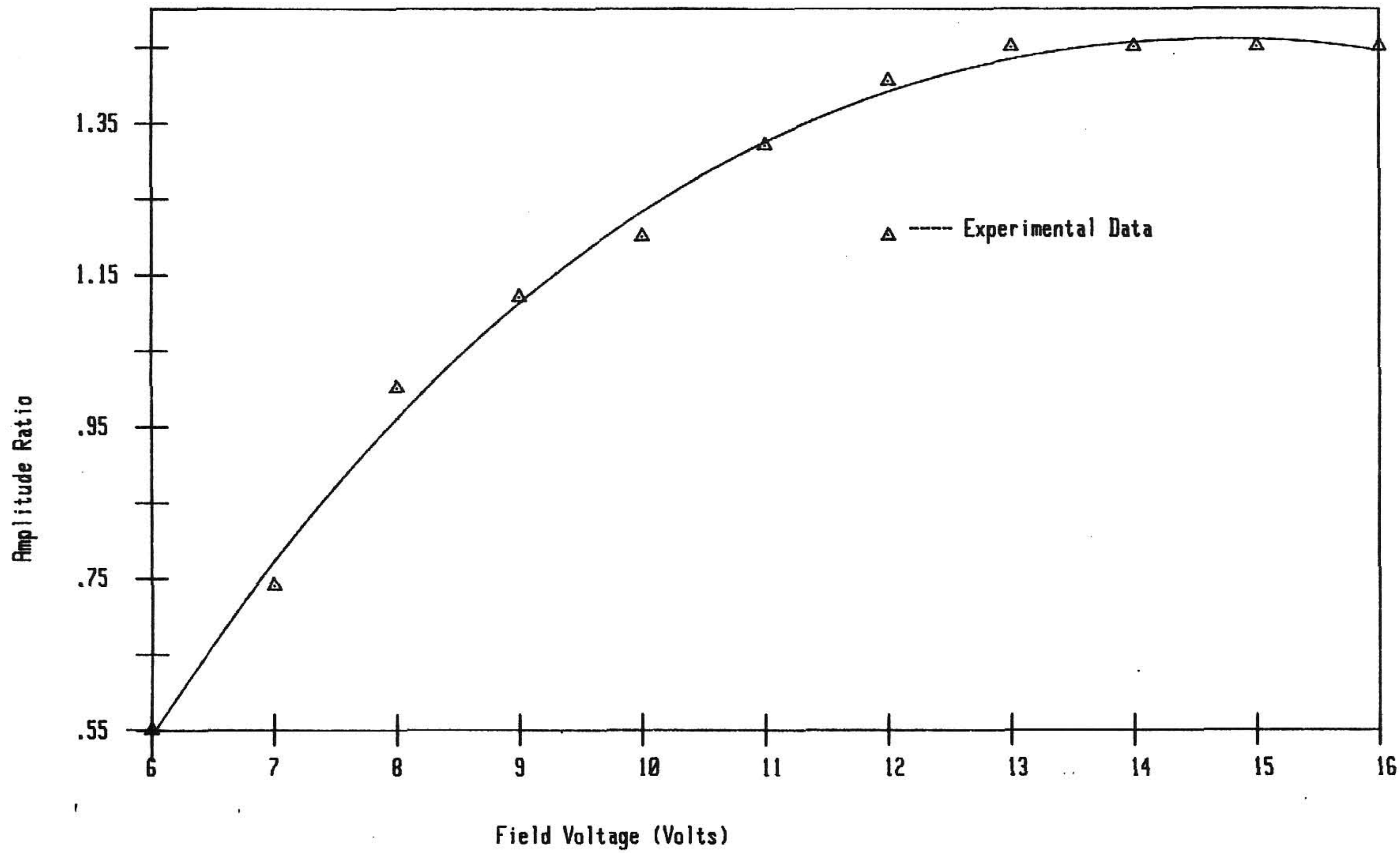

Figure 2.6. Amplitude Ratio versus Field Voltage

From Figure 2.6 it is observed that the amplitude ratio increases as the field voltage is increased. After a certain value, the amplitude ratio does not continue to increase even if the field voltage is increased. This is due to saturation of the field windings.

For all future operation of the system the field voltage is maintained at 14 volts which is the value at which saturation first occurs. Small changes in field voltage setting will have little effect on the performance of the system.

2.5 Tachometer Gain

A d.c. motor is used to drive the wheels of the cart through a belt drive arrangement with a gear ratio of 19.7:1. The speed of the d.c. motor is obtained by a tachometer which is directly coupled to the d.c. motor. Hence, to determine the gain of the tachometer it is important to find a relationship between the speed of the cart and the voltage output from the tachometer.

In this experiment, the cart is supported such that the wheels do not touch the ground. A constant 14 volts is applied across the field of the d.c. motor. To vary the speed the voltage across the armature of the d.c. motor is varied. Various readings for the speed of the d.c. motor are obtained by a stroboscope. The voltage outputs from the tachometer are recorded on the strip chart recorder. The values of the speed of the d.c. motor and the voltage output from the tachometer are given in Table 2.2.

Observation Number	R.P.M. of Armature	Speed of Armature in rad/sec	Voltage Output from the Tachometer
1.	64	6.7	0.475
2.	130	13.6	1.105
3.	235	24.6	1.56
4.	420	44.0	3.20
5.	540	56.6	4.25
6.	990	103.7	7.0
7.	1300	136.1	9.77
8.	1460	152.9	10.87
9.	1530	160.2	11.37
10.	2180	228.3	16.00

Table 2.2 Experimental Data to find the Gain of the Tachometer.

A plot of the speed of the artmature v/s output voltage from the tachometer is shown in Figure 2.7. It is observed that the graph is almost linear, so it can be concluded from the graph that the gain, Kt, of the tachometer is 0.072 volts/rad/sec.

The transmission ratio, Kw, of the cart is calculated by Equation 2.14,

$$Kw = \frac{gr}{rw} = \frac{19.7}{0.1035} = 190.34 \text{ radians/m.} \qquad (2.14)$$

where rw = radius of the wheels of the cart = 0.1035 m, and

gr = gearing ratio between the armature of the motor and the wheels of the cart = 19.7:1.

All physical systems have nonlinearities. It is a well established practice that providing a feedback around a system

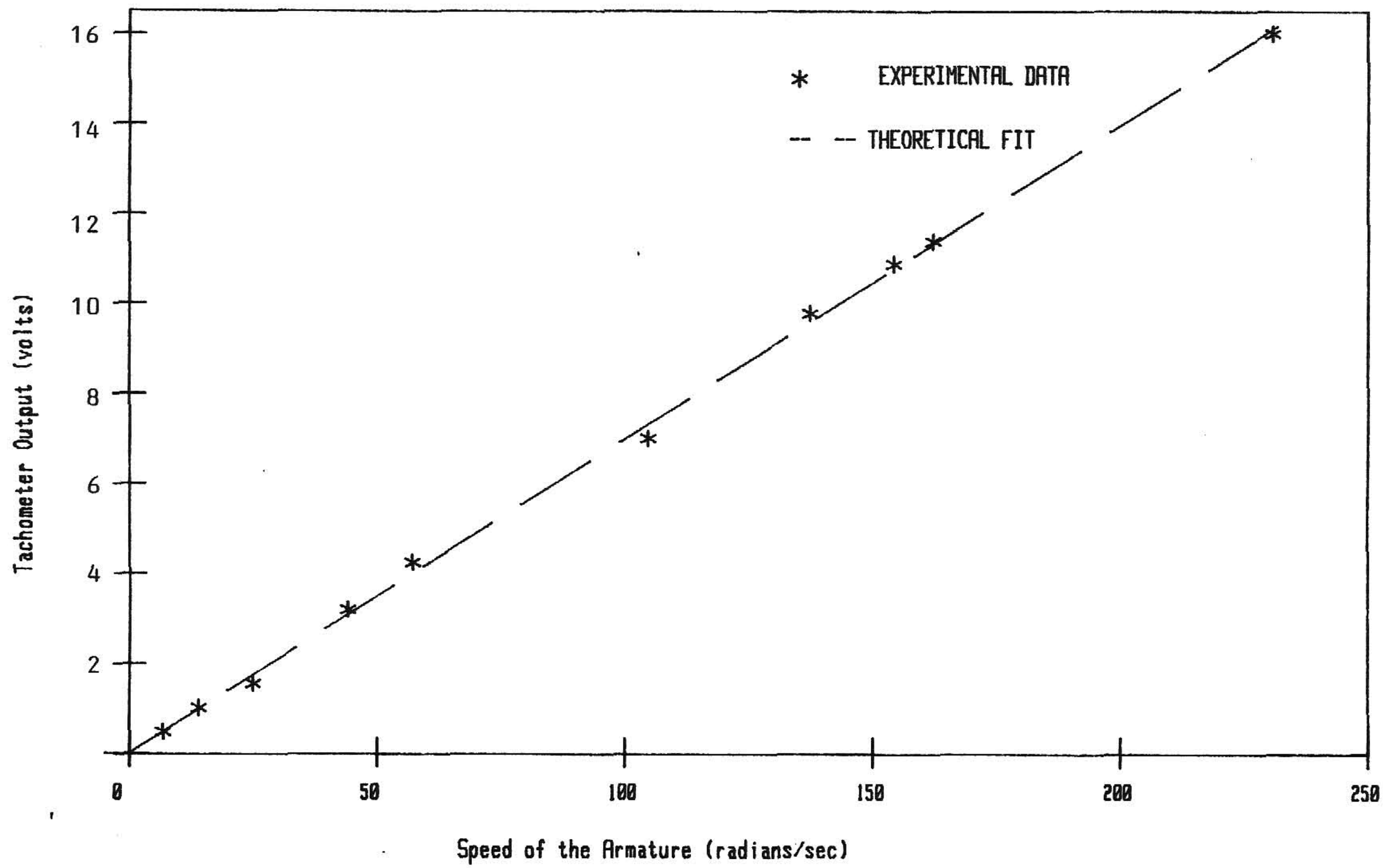

Figure 2.7. Plot of Speed of the Armature versus Output Voltage from the Tachometer

reduces the nonlinear effects. Therefore, the tachometer output was amplified in an operational amplifier with a gain of 2.3 and was fed back, summed with the control signal, and amplified by a gain of 20 to provide the input signal to the power amplifier as shown in Figure 2.8. The signal U is the control input and signal Vb is the output signal from the summing amplifier which drives the the power amplifier to produce a corresponding voltage across the armature of the d.c. motor. After applying feedback around the system, an experiment was conducted to determine the coefficients of the transfer function of the system.

2.6 Experimental Determination of the Transfer Function of the System Consisting of the D.C. Motor, Cart, Drive and Tachometer.

The first step in the analysis and design of a control system is to develope a mathematical model of the system. In Equation 2.9 the transfer function relating the motor speed to the armature voltage is given. Applying feedback around the system as shown in Figure 2.8, yields a transfer function relating the tachometer voltage to the control voltage of the form

$$\frac{Vt(s)}{U(s)} = \frac{Kc*(Mc*s^2 + fc*s + Kb*Kw^2)}{C3*s^3 + C2*s^2 + C1*s + C0} \qquad (2.15)$$

where $Kc = 46*Ka*Kt*Km/Ra$,

Ka = Power Amplifier gain = 1 ,

$C3 = Ja*Mc$,

$C2 = fa*Mc + fc*Ja + Mc*Km*Ks/Ra + Kc*Mc$,

$C1 = Kb*Mc + fa*fc + Ja*Kb*Kw^2 + fc*Km*Ks/Ra + Kc*fc$, and

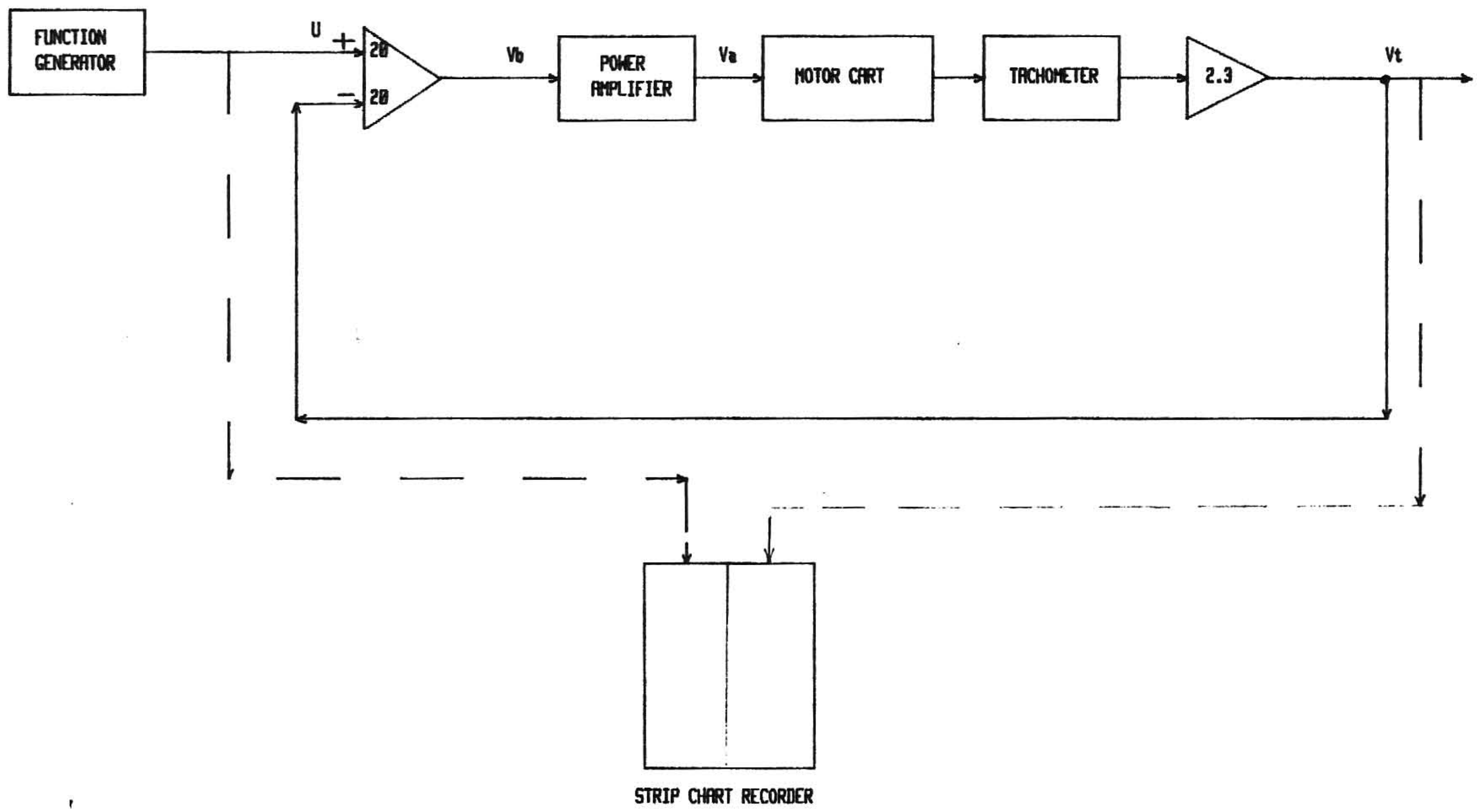

Figure 2.8 Experimental Setup for the Frequency Response Measurement of the System.

$$C0 = fc*Kb + Kb*fa*Kw^2 + Km*KsKw^2/Ra + Kc*Kb*Kw^2.$$

Observe that the form of this transfer function is the same as before, eg. the feed back simply changes the values of the coefficients and reduces the effects of nonlinearities. In order to proceed with the analysis of the control system it is necessary to obtain numerical values for the coefficients of the transfer function. One way to do this is to make tests and measurements to determine the values of the individual parameters and compute the values of the coefficients. Several of the parameters are rather difficult to measure. Another approach is to make a set of measurements of the system and determine coefficient values which cause the transfer function model to fit the measured response. The frequency response method follows this approach.

The procedure involved to obtain a frequency response of the system is to excite the system with a sinusoidal input with measured amplitude. The output amplitude is measured and the amplitude ratio of the output to the input is computed. The measurement is repeated at different frequencies. A plot of the amplitude ratio versus frequency is called a Bode plot. The coefficients of the transfer function are determined by fitting the model to the Bode plot. An asymptotic log-magnitude curve consisting of several segments is built. With some judgement and experience it is usually possible to find a close fit to the experimental data.

Figure 2.8 shows the experimental arrangement for the frequency response measurement . The input voltage from the function generator to the summing operational amplifier is a sinusoid with an amplitude of 3.10 volts peak to peak, which is maintained at all frequencies. A constant 14 volts is applied to the field winding of the d.c. motor. The input signal , U, from the function generator and the output signal, Vt, from the tachometer are recorded on the strip chart recorder. Table 2.3 lists the frequencies and the corresponding output amplitudes and amplitude ratios for which data were collected.

A Bode diagram of the experimental data is plotted on Figure 2.9. The frequency response of the transfer function chosen to approximate the experimental data is shown on Figure 2.9. The transfer function of the system can be approximated by,

$$\frac{Vt(s)}{U(s)} = \frac{0.92*500*(s^2 + 11.5*s + 2285)}{(s + 500)*(s^2 + 21.4*s + 2362)} . \qquad (2.16)$$

The pole at -500 is almost 50 times as far from the imaginary axis as the two complex poles and will have a negligible effect on the performance of the system. It will be ignored in the subsequent analysis .

2.8 Block Diagram Representation of the System

In the previous sections of this chapter mathematical models have been developed for each element of the system. Experiments have been described by which the coefficients of these models have been determined. In this section the form of the control will be developed in preparation for the establishment of specific compensators in subsequent chapters.

No.	Frequency radians/sec	Output voltage peak to peak	Amplitude Ratio=A.R	20*Log(A.R)
1.	6.28	2.75	0.887	-1.04
2.	12.18	2.75	0.887	-1.04
3.	19.10	2.75	0.887	-1.04
4.	22.56	2.75	0.887	-1.04
5.	25.94	2.70	0.871	-1.20
6.	32.04	2.50	0.806	-1.87
7.	34.44	2.40	0.774	-2.23
8.	38.58	2.17	0.700	-3.10
9.	41.37	1.90	0.613	-4.25
10.	43.60	1.75	0.565	-4.96
11.	78.82	1.50	0.484	-6.30
12.	49.52	1.85	0.596	-4.50
13.	52.34	2.00	0.645	-3.81
14.	54.66	2.02	0.652	-3.72
15.	61.20	2.125	0.685	-3.29
16.	67.10	2.26	0.729	-2.75
17.	72.75	2.4	0.774	-2.23
18.	84.88	2.6	0.838	-1.54
19.	100.98	2.71	0.874	-1.17
20.	141.38	2.83	0.913	-0.79
21.	251.32	2.80	0.903	-0.89

Table 2.3 Experimental Frequency Response of the System.

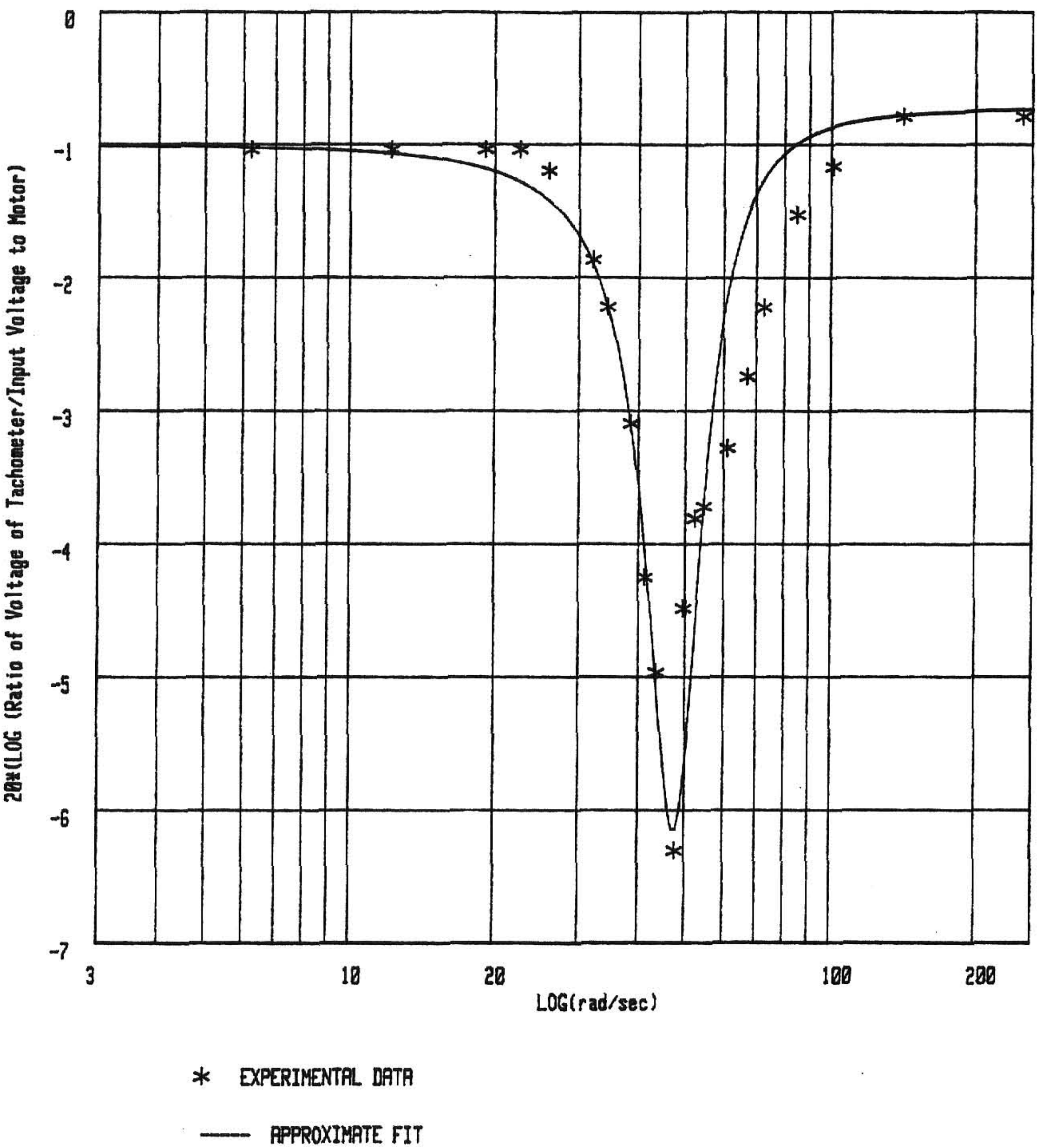

Figure 2.9. Bode Plot of the Closed Loop System with Experimental Data and the Approximate Fit

Figure 2.10 shows a block diagram of the system. The blocks within the dashed line represent compensators implemented in the microcomputer. The rest of the blocks represent the physical elements of the inverted pendulum system. The value which results from the digital computation is placed in a latch, which drives a digital-to-analog converter. Each time the computations are performed a new value is placed in the latch. U represents the analog signal which is output by the digital-to-analog converter. The zero-order-hold (ZOH) represents the action of the latch and the digital-to-analog converter. The signals represented by Vt and Vp are the signals out of the tachometer and potentiometer respectively. These two digital signals are sampled by the computer and provide the signals upon which the digital computations are performed. The two digital values, Vt^* and Vp^* are produced by the analog-to-digital converters and correspondent to the signals Vt and Vp.

The transfer function, Gl(s) relating the voltage output from the tachometer, Vt, to the input voltage, U, to the summing operational amplifier was given in Equation 2.16,

$$Gl(s) = \frac{Vt(s)}{U(s)} = \frac{0.92*(s^2 + 11.5*s + 2285)}{s^2 + 21.4*s + 2362} \qquad (2.17)$$

The block following Gl(s), with a transfer function 1/Kt relates the speed of the armature of the cart with the output voltage from the tachometer. Kt is the tachometer gain equal to 0.072 volts/rad/sec.

Next, the values of the parameters in Equation 2.8 relating

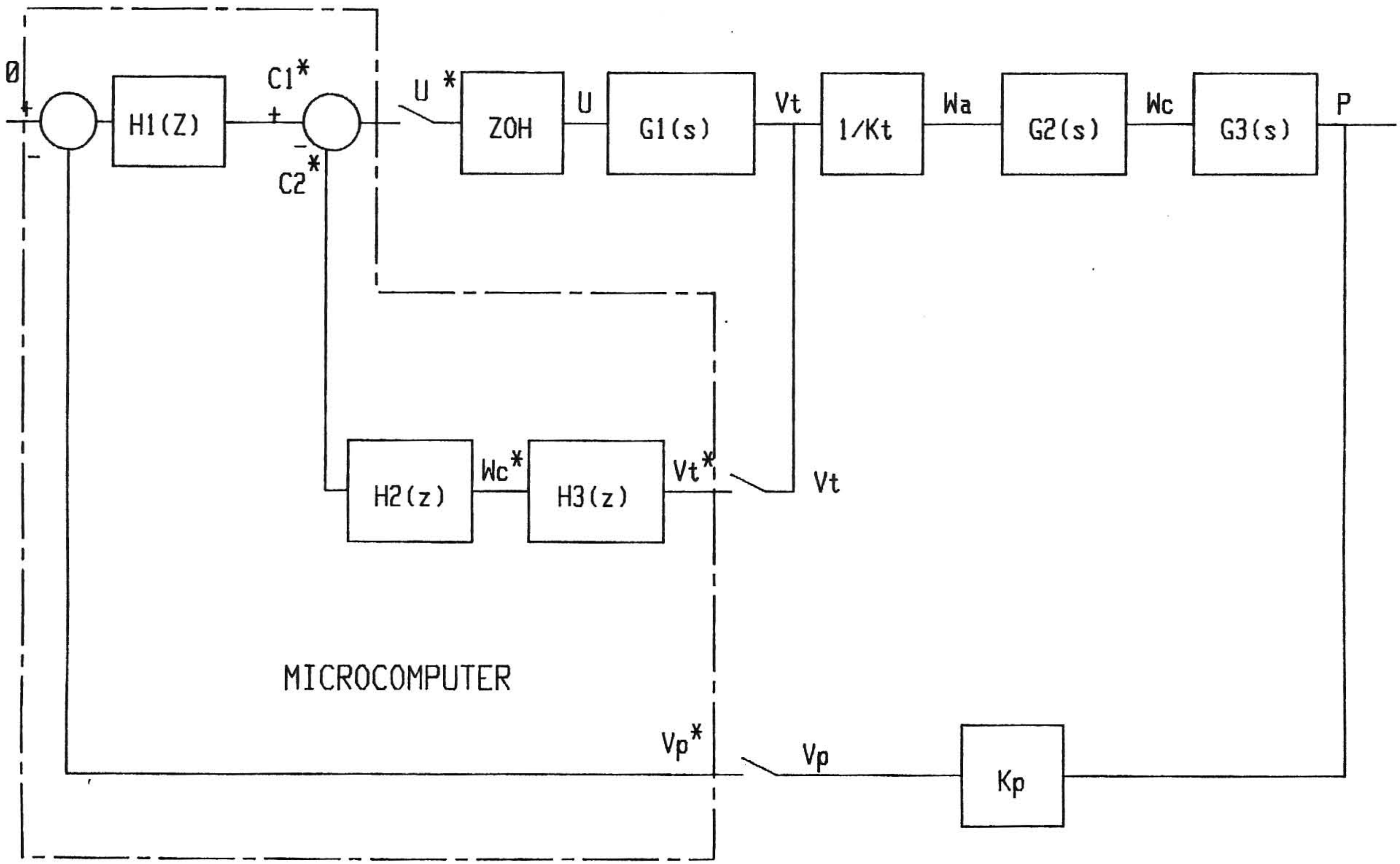

Figure 2.1Ø. Block Diagram Representation of the System.

Wc to Wa can be established using Equations 2.17 and 2.14. Therefore the next transfer function in the block diagram is,

$$G2(s) = \frac{Wc(s)}{Wa(s)} = \frac{12.0}{s^2 + 11.5*s + 2285} . \qquad (2.18)$$

The transfer function, G3(s), relating the position of the pendulum, P, to the speed of the cart, Wc, is given by Equation 2.13,

$$G3(s) = \frac{P(s)}{Wc(s)} = \frac{-2.5*s}{s^2 - 25} . \qquad (2.19)$$

The position of the pendulum is measured by a potentiometer which is mounted on the pivot of the pendulum. The gain of the potentiometer is represented by Kp in the block diagram. The value of Kp is adjustable.

The control strategy requires that the velocity of the cart be known. Since there is no direct measurement of the velocity of the cart it is necessary to compute it from the measured value of the speed of the motor. The digital model H3(z), developed in Chapter III, provides a computed estimate of the velocity of the cart. H1(z) and H2(z) are compensation transfer functions which are developed in Chapter IV to stabilize the pendulum-cart system.

CHAPTER III

Z-TRANSFORM ANALYSIS

3.1 Introduction

The role of the z-transform to sampled data systems is similar to that of the Laplace transform to continuous data systems. The z-transform technique provides a mean of finding the output of a system at the k-th sample time, t=kT, in terms of the input at the k-th sample time, t=kT, and the input, and the output at previous sample times,t=(k-i)T . This technique is applied to the problem of finding the voltage input to the d.c. motor in terms of the position of the pendulum and velocity of the cart. The z-transform technique is demonstrated below.

A time function x(t), shifted in time by an amount nT is x(t-nT). The Laplace transform is,

$$L(x(t - nT)) = \int_{0}^{\infty} x(t - nT) * e^{-st} dt. \qquad (3.1)$$

Subsituting , h = t - nT, yields

$$L(x(t - nT)) = \int_{-nT}^{\infty} x(h) * e^{-s(h+nT)} dh. \qquad (3.2)$$

Since x(h) = 0 for $h \leqslant 0$, Equation 3.1 may be written as

$$L(x(t - nT)) = e^{-snT} * \int_{0}^{\infty} x(h) * e^{-sh} dh.$$
$$= X(s) * e^{-snT} \qquad (3.3)$$

The z-transform is obtained from the Laplace transform by substituting $z= e^{sT}$. Thus Equation 3.3 gives the z-transform of x(t-nT) as

$$Z(x(t-nT)) = X(z)*z^{-n} . \qquad (3.4)$$

Taking the inverse z-transform yields $x(t-nT)=Z^{-1}(X(z)*z^{-n})$ from Equation 3.4. This is known as the shifting theorem of z-transforms. The z-transfer function of a system, which represents the z-transform of the output over the z-transform of the input is a ratio of the polynomials in z^{-1} thus,

$$\frac{X(z)}{Y(z)} = \frac{a_0 + a_1*z^{-1} + a_2*z^{-2} + ... + a_m*z^{-m}}{1 + b_1*z^{-1} + b_2*z^{-2} + ... + b_n*z^{-n}} \qquad (3.5)$$

where X(z) and Y(z) are the transforms of the output and input respectively. Cross multiplying yields

$$\begin{aligned}&(1 + b_1*z^{-1} + b_2*z^{-2} +....... + b_n*z^{-n})*X(z)\\ &= (a_0 + a_1*z^{-1} + a_2*z^{-2} + ... + a_m*z^{-m})*Y(z) \qquad (3.6)\end{aligned}$$

Taking the inverse z-transform and using the shifting theorem yields,

$$\begin{aligned}&x(t) + b_1*x(t-T) + b_2*x(t-2T) + ... + b_n*x(t-nT)\\ &= a_0*y(t) + a_1*y(t-T) + a_2*y(t-2T) +....+ a_m*y(t-mT) \qquad (3.7)\end{aligned}$$

The output function, x(t), at time t can be written in terms of the output and input at previous times t-iT as,

$$\begin{aligned}x(t) = &a_0*y(t) + a_1*y(t-T) + ... + a_m*y(t-mT)\\ &-(b_1*x(t-T) + b_2*x(t-2T)+ ......+ b_n*x(t-nT)) \qquad (3.8)\end{aligned}$$

For t=kT Equation 3.8 may be written as,

$$x(k) = a_0*y(k) + a_1*y(k-1) + \ldots + a_m*y(k-m)$$
$$- (b_1*x(k-1) + b_2*x(k-2) + \ldots + b_n*x(k-n)) \qquad (3.9)$$

where the sampling period T is understood.

3.2 Z-Transforms of the System

The z-transform of the system is found from a table of Laplace to z-transforms. The technique, used for transformation, is to first find the partial fraction expansion of a Laplace transfer function and then to obtain the z-transforms from the table.

The s-transfer functions whose z-transforms are required, are $Vt(s)/U^*(s)$, $Vp(s)/U^*(s)$ and $Wc(s)/Vt^*(s)$, as shown in Figure 2.10. The Laplace transfer functions of the system have been derived in the previous chapter where,

Vt = output signal from the tachometer,
Vt^* = sampled value of Vt,
Wc = velocity of the cart,
U = output voltage to the summing power amplifier,
U^* = sampled signal of U, the control signal,and
Vp = output signal from the potentiometer.

3.2.1 Z-Transfer Function of $Vt(s)/U^*(s)$

The transfer function of $Vt(s)/U^*(s)$ is equal to ZOH*Gl(s), as shown in Figure 2.10. Therefore,

$$\frac{Vt(s)}{U^*(s)} = \frac{(1 - e^{-sT})}{s} * \frac{0.92*(s^2 + 11.5*s + 2285)}{(s^2 + 21.4*s + 2362)} \qquad (3.10)$$

Taking the partial fraction expansion of Equation 3.10 gives,

$$\frac{Vt(s)}{U^*(s)} = 0.92*(1 - e^{-sT})*\left[\frac{A}{s} + \frac{B*s + C}{s^2 + 21.4*s + 2362}\right] \qquad (3.11)$$

where A, B and C are the coefficients of the expansion. These constant are, A = 0.9674, B = 0.0326 and C = -9.2024.

Hence,

$$\frac{Vt(s)}{U^*(s)} = 0.92*(1 - e^{-sT})*\left[\frac{0.9674}{s} + \frac{0.0326*s - 9.2024}{s^2 + 21.4*s + 2362}\right] \qquad (3.12)$$

The z-transform of Equation 3.12 is ,

$$\frac{Vt(z)}{U(z)} = 0.92*(1 - z^{-1})*\left[\frac{0.9674*z}{z - 1} + \frac{0.0326*(z^2 - z*\cos(bT)*e^{-aT}))}{z^2 - 2*z*\cos(bT)*e^{-aT} + e^{-2aT}} - \frac{0.2015*z*\sin(bT)*e^{-aT}}{(z^2 - 2*z*\cos(bT)*e^{-aT} + e^{-2aT})}\right] \qquad (3.13)$$

where, a = 10.7 and b = 47.41.

Simplifying Equation 3.13 yields,

$$\frac{Vt(z)}{U(z)} = \frac{0.92*(z^2 - N11*z + N10)}{(z^2 - D11*z + D10)} \qquad (3.14)$$

where,

$N11 = 1.9674*\cos(bT)*e^{-aT} + 0.2017*\sin(bT)*e^{-aT} + 0.0326$,

$N10 = 0.9674*e^{-2aT} + (0.0326*\cos(bT) + 0.2017*\sin(bT))*e^{-aT}$,

$D11 = 2*\cos(bT)*e^{-aT}$, and

$D10 = e^{-2aT}$.

3.2.2 Z-Transfer Function of $Wc(s)/U^*(s)$

The transfer function of $Wc(s)/U^*(s)$ is equal to ZOH*G1(s)*(1/Kt)*G2(s) as shown in Figure 3.1.

Therefore,

$$\frac{Wc(s)}{U^*(s)} = \frac{153.3*(1 - e^{-sT})}{s*(s^2 + 21.4*s + 2362)} \tag{3.15}$$

The partial fraction expansion of Equation 3.15 is,

$$\frac{Wc(s)}{U^*(s)} = \frac{153.3}{2362}*(1 - e^{-sT})*\left[\frac{A}{s} + \frac{B*s + C}{s^2 + 21.4*s + 2362}\right] \tag{3.16}$$

where, A = 1, B = -1 and C = -21.4.

Therefore,

$$\frac{Wc(s)}{U^*(s)} = 0.0649*(1 - e^{-sT})*\left[\frac{1}{s} - \frac{s + 21.4}{s^2 + 21.4*s + 2362}\right] \tag{3.17}$$

The z-transform of the partial fraction expansion is obtained from the table as,

$$\frac{Wc(z)}{U(z)} = 0.0649*(1 - z^{-1})*\left[\frac{z}{z-1} - \frac{z*(z-(\cos(bT)-0.2256*\sin(bT))*e^{-aT})}{z^2 - 2*z*\cos(bT)*e^{-at} + e^{-2aT}}\right] \tag{3.18}$$

where, a = 10.7 and b = 47.41.

Simplifying Equation 3.18 as,

$$\frac{Wc(z)}{U(z)} = \frac{0.0649*(N21*z - N20\ )}{(z^2 - D11*z + D10\ )} \qquad (3.19)$$

where

$N21 = 1 - (\cos(bT) + 0.2256*\sin(bT))*e^{-aT}$, and

$N20 = e^{-2aT} - (\cos(bT) - 0.2256*\sin(bT))*e^{-aT}$.

3.2.3 Z-Transfer Function of $Vp(s)/U^*(s)$

The transfer function of $Vp(s)/U^*(s)$ is equal to $ZOH*G1(s)*(1/Kt)*G2(s)*G3(s)*Kp$. Therefore,

$$\frac{Vp(s)}{U^*(s)} = \frac{(1 - e^{-sT})}{s} * \frac{-0.92*(1/0.072)*12.0*2.5*Kp*s}{(s^2 + 21.4*s + 2362)*(s^2 - 25)} \qquad (3.20)$$

The partial fraction expansion of Equation 3.20 is,

$$\frac{Vp(s)}{U^*(s)} = -383.5*Kp*(1-e^{-sT})*\left[\frac{A}{s - 5} + \frac{B}{s + 5} + \frac{C*s + D}{s^2 + 21.4*s + 2362}\right] \qquad (3.21)$$

where the coefficients of expansion are, $A = 4.0098*10^{-5}$, $B = -4.3859*10^{-5}$, $C = 3.7607*10^{-6}$ and $D = -3.3937*10^{-4}$.

The z-transform of Equation 3.21 is,

$$\frac{Vp(z)}{U(z)} = -383.5*Kp*(1-z^{-1})\left[\frac{A1*z}{z-e^{5T}} + \frac{A2*z}{z-e^{-5T}} + \frac{A3*(z^2 - z*\cos(bT)*e^{-aT}}{z^2-2*z*\cos(bT)*e^{-aT}+e^{-2a}} - \frac{A4*z*\sin(bT)*e^{-aT}}{z^2 - 2*z*\cos(bT)*e^{-aT} + e^{-2aT}}\right]*10^{-5} \qquad (3.22)$$

where, A1 = 4.0098, A2 = -4.3859, A3 = 0.3761,

A4 = 0.8006, a = 10.7 and b = 47.41.

Simplifying Equations 3.22 as,

$$\frac{Vp(z)}{U(z)} = \frac{+383.56*Kp*10^{-5}*(z-1)*(N32*z^2 - N31*z + N30)}{(z-e^{-5T})*(z-e^{5T})*(z^2 - D11*z + D10)} \qquad (3.23)$$

where,

$$N32=(A1+A3)*e^{-5T} + (A2+A3)*e^{5T} + 2*(A1+A2)*\cos(bT)*e^{-aT} + (A3*\cos(bT)+A4*\sin(bT))*e^{-aT},$$

$$N31= (A1 + A2)*e^{-2aT} + A3 + (e^{-5T}+e^{5T})*(A3*\cos(bT)+A4*\sin(bT))*e^{-aT} + 2*(A1*e^{-5T} + A2*e^{-5T})*\cos(bT)*e^{-aT},$$ and

$$N30=(A1*e^{-5T}+A2*e^{5T})*e^{-2aT}+(A3*\cos(bT)+A4*\sin(bT))*e^{-aT},$$

3.3 Transfer Function for the Block H3(z)

The symbol Wc represents the velocity of the cart, and Wc^* represents an estimate of the velocity of the cart that is generated by the microcomputer from the measurement of the tachometer voltage. In order to obtain the estimated velocity, the digital model H3(z) is developed. This digital model describes the relationship between the digital signal from the output of the tachometer, Vt^* , and the speed of the cart.

This relationship is obtained by dividing Equation 3.19 by Equation 3.14, thus,

$$H3(z) = \frac{Wc(z)}{Vt(z)} = \frac{0.0655*(N21*z - N20)}{0.92*(z^2 - N11*z + N10)} \qquad (3.24)$$

The next chapter contains the stability analysis of the system in the z-domain. The compensating transfer functions H1(z) and H2(z) are developed to stabilize the system.

CHAPTER IV

STABILITY ANALYSIS IN Z-DOMAIN

4.1 Introduction

A linear continuous feedback control system is stable if all the poles of the closed loop transfer function lie in the left half of the s-plane. The z-plane is related to the s-plane by the transformation

$$z = e^{sT}$$

or

$$z = e^{(r+jw)T} . \qquad (4.1)$$

This relationship may also be written as

$$\left| z \right| = e^{rT} . \qquad (4.2)$$

For the left - half s-plane, $r < 0$, the relative magnitude of z varies between 0 and 1. Therefore, the imaginary axis of the s-plane corresponds to the unit circle in the z-plane and the inside of the unit circle corresponds to the left half of the s-plane. Thus a sample data system is stable if all the poles of the closed loop transfer function lie within the unit circle (6).

A sampling period of T=0.01 secs is selected for the system. The frequency of sampling is 12 times the natural frequency of the drive system, and hence the reconstruction of the sampled signal is quite accurate.

The naturally unstable pendulum-cart system is stabilized by a feedback compensation technique. The stability analysis is given in the rest of this chapter.

4.2 Stability Analysis

The stability of a closed loop control system is directly related to the location of the roots of the characteristic equation in the z plane. Therefore, it is necessary to determine the roots of the characteristic equation for a check on stability. The root locus method was introduced by Evans in 1948 and has been developed and utilized extensively in control engineering. The root locus technique is a graphical method for drawing the locus of the roots in the s and z planes as a parameter is varied. If the root locations are not satisfactory the necessary parameter adjustment to stabilize the system can be readily deduced from the root locus.

The block diagram of the system is shown in Figure 4.1. The transfer functions Vt(z)/U(z), Vp(z)/U(z) and Wc(z)/Vt(z) are derived in Chapter III. Substituting T=0.01 secs. into the above transfer functions yields,

$$D1(z) = \frac{Vt(z)}{U(z)} = \frac{0.92*(z^2 - 1.688*z + 0.89)}{(z^2 - 1.599*z + 0.807)}, \tag{4.3}$$

$$D2(z) = \frac{Vp(z)}{U(z)} = \frac{-3.835*10^{-5}*Kp*(z-1)*(1.562*z^2+5.862*z+1.4033)}{(z-0.9521)*(z-1.0513)*(z^2-1.599*z+0.807)} \tag{4.4}$$

and

$$H3(z) = \frac{Wc(z)}{Vt(z)} = \frac{0.0077*(z + 0.931)}{(z^2 - 1.688*z + 0.89)}. \tag{4.5}$$

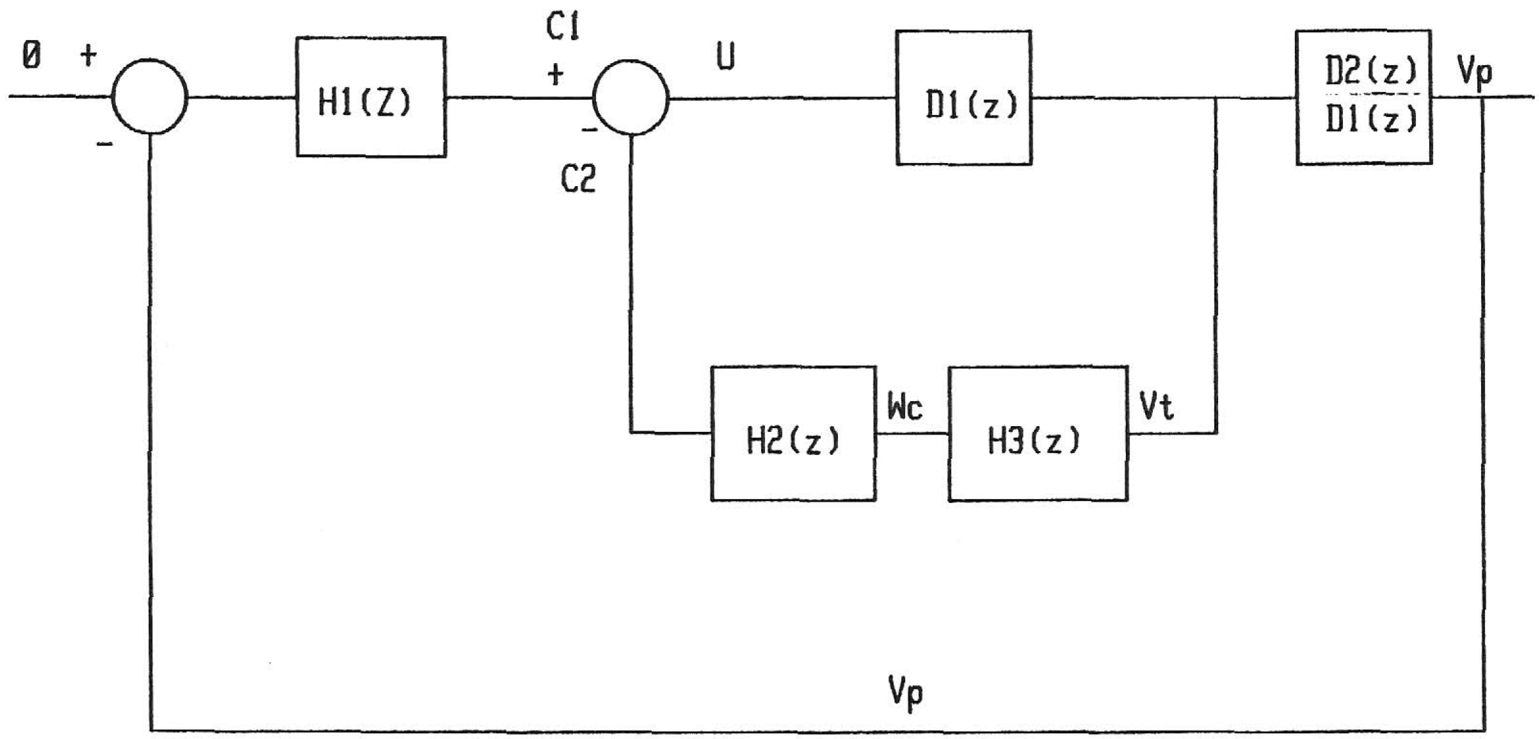

Figure 4.1 Block Diagram of the System.

4.2.1 Stability Analysis of the Inner Loop.

The transfer function of the inner loop of the system shown in Figure 4.1 is,

$$\frac{Vt(z)}{C1(z)} = \frac{D1(z)}{1 + D1(z)*H2(z)*H3(z)} \tag{4.6}$$

Therefore, the charactersitic equation of the inner loop is,

$$1 + D1(z)*H2(z)*H3(z) = 0 \tag{4.7}$$

The compensating transfer function, H2, is unknown, and hence needs to be chosen to have a stabilizing effect on the inner loop of the system.

Substituting D1(z) from Equation 4.3 and H3(z) from Equation 4.5 into Equation 4.7, yields,

$$1+\frac{H(2)*0.0077*(z + 0.931)*0.92(z^2 -1.688*z +0.89)}{(z^2 - 1.688*z + 0.89)*(z^2 - 1.599*z + 0.807)} = 0 \tag{4.8}$$

or

$$1 + \frac{H2(z)*0.0071*(z + 0.931)}{(z^2 - 1.599*z + 0.807)} = 0 . \tag{4.9}$$

If H2(z) is a gain compensation, then Equation 4.9 has a zero at Z1 = -0.931 and a pair of complex conjugate poles at P1 = 0.7995+0.409j and P2 = 0.7995-0.409j.

Therefore, the characteristic equation is,

$$z^2 - 1.599*z +0.807 + H2*0.0071*(z + 0.931) = 0 . \tag{4.10}$$

In order to determine a suitable gain, H2, the roots of the characteristic Equation 4.10 are plotted for $0 < H2 < \infty$. Figure 4.2 shows the location of the poles and the zero, as well as the

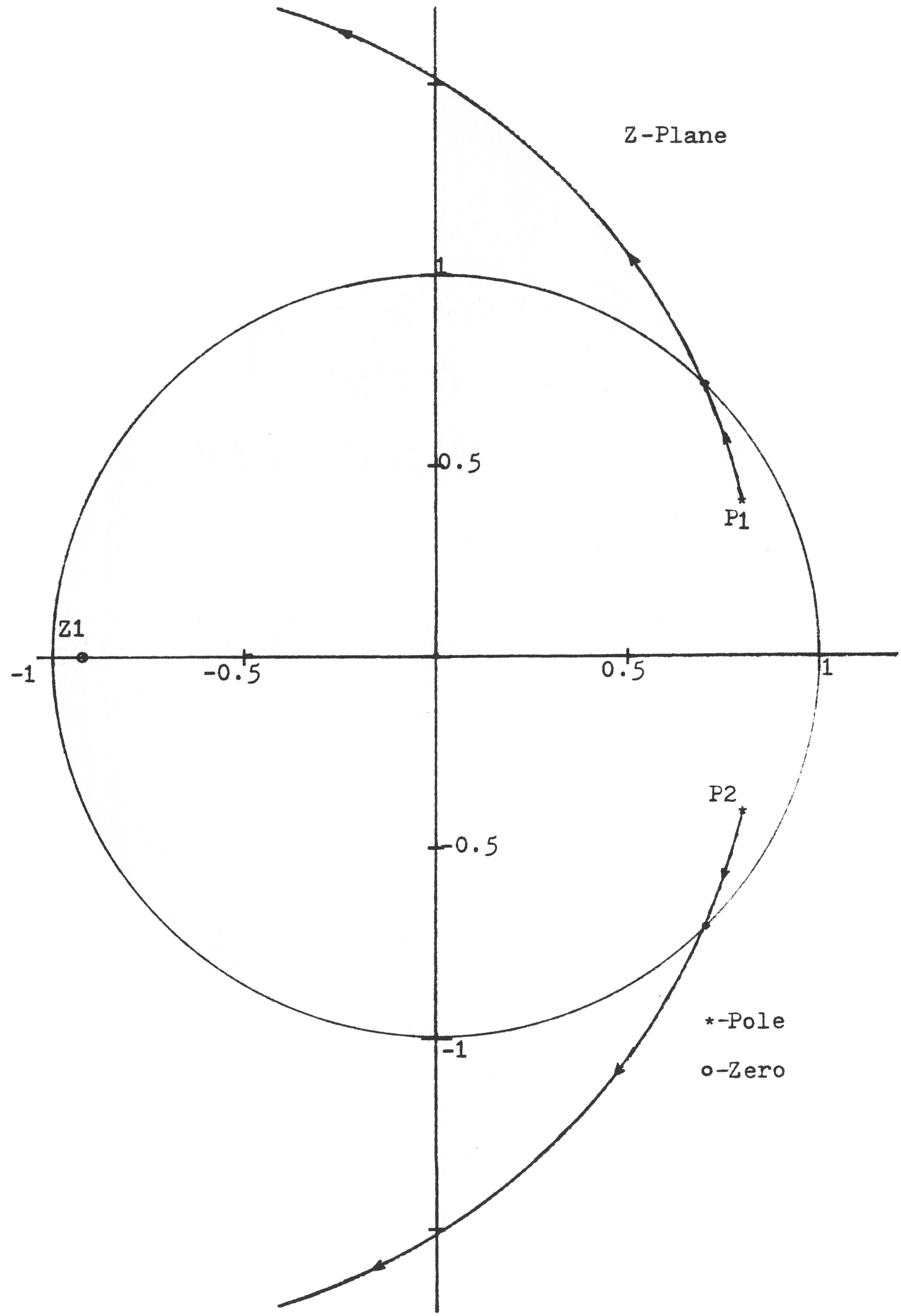

Figure 4.2 Root Locus of the Inner Loop of the System with a Gain Compensation

locus of the roots of the inner loop of the system for increasing gain, H2. The roots have a tendency to move out of the unit circle as the gain H2 is increased. This causes a destabilizing effect on the inner loop of the system. Therefore the approach at this point is to choose a transfer function, H2, such that the roots do not have a tendency to move out of the unit circle. Hence H2 is chosen as,

$$H2(z) = \frac{C2(z)}{Wc(z)} = \frac{K1*(z - 0.85)^2}{(z + 0.931)*(z - 1)}, \qquad (4.11)$$

where K1 is a variable gain which is determined by root locus analysis.

Chosing H2(z) above is also due to the following reasons:

a) A pole (z + 0.931) of the transfer function H2(z) cancels the zero of H3(z). This reduces the computation time in the real time controller and simplifies the analysis.

b) The pole (z - 1) in H2(z) provides digital integration of the velocity, Wc^*, of the cart, eg., a signal proportional to the displacement of the cart.

c) Second order zeros need to be placed such that the roots of the characteristic equation of the inner loop remain inside the unit circle. Hence the zeros $(z - 0.85)^2$ are chosen in the transfer function H2(z).

Substituting H2(z) in Equation 4.10 yields the characteristic equation of the inner loop as

$$(z^2 - 1.599*z + 0.807)*(z - 1) + Ki*(z - 0.85)^2 = 0 \qquad (4.12)$$

where $Ki = K1*0.0071$. (4.13)

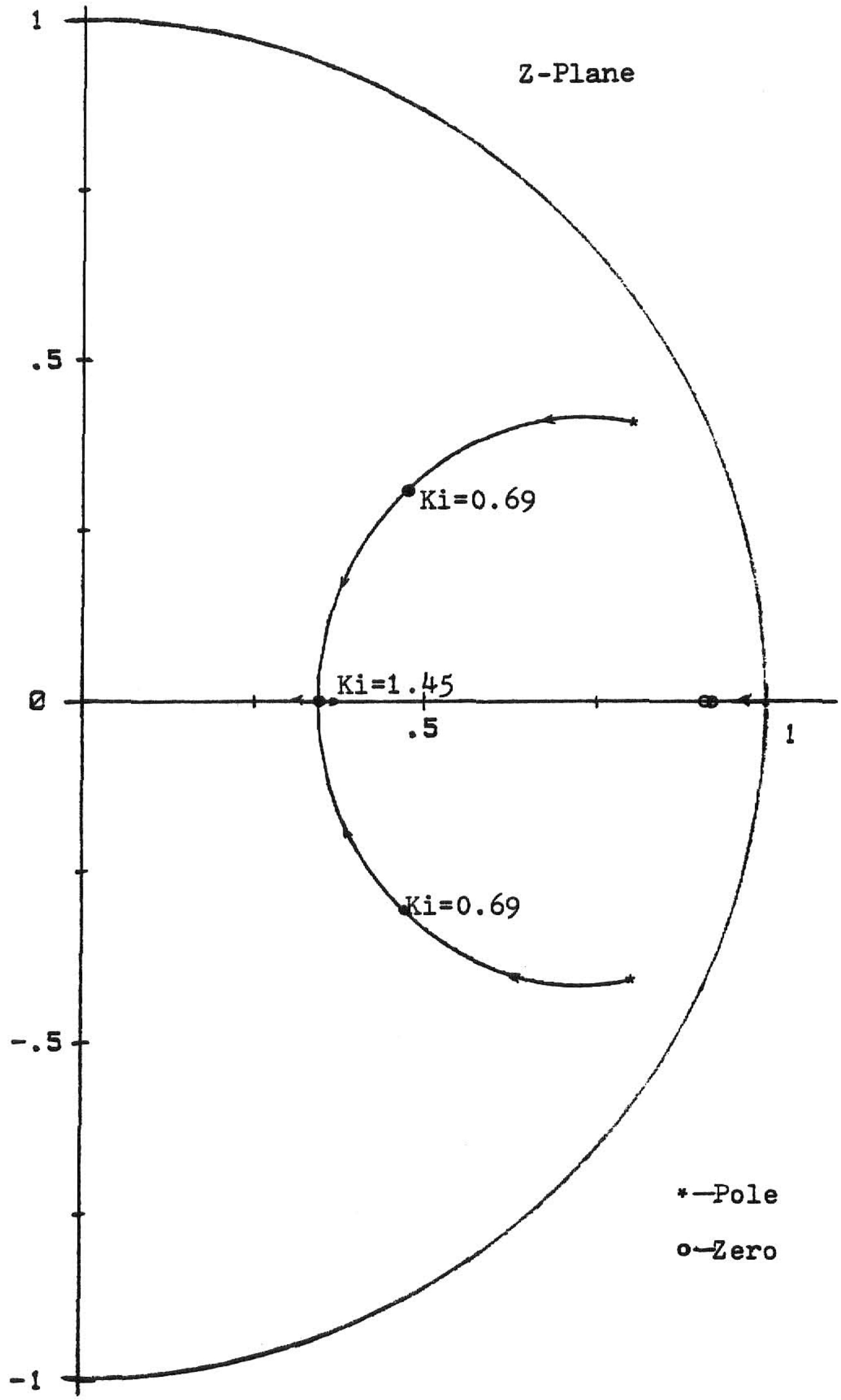

Figure 4.3 Root Locus of the Inner Loop of the System in the Z-Plane with a Compensation Transfer Function

Figure 4.3 shows the locus of the roots of the characteristic equation of the inner loop of the system as the gain, Ki, is increased. The values of the roots of the characteristic equations for increasing gain, Ki, are given in Appendix A. From Figure 4.3 it can be deduced that the inner loop of the system has a tendency to stabilize, as the gain Ki is increased. A gain of Ki = 0.69 is selected for further analysis of the system. Therefore the value of K1 as deduced from Equation 4.13 is 97.18.

4.2.2 Stability Analysis Of The Entire System

The characteristic equation of the entire system shown in Figure 4.1 can be deduced as,

$$1 + D1(z)*H2(z)*H3(z) + D2(z)*H1(z)=0 \; . \qquad (4.14)$$

The compensating transfer function, H1(z), is selected as

$$H1(z) = \frac{C1(z)}{Vp(z)} = \frac{K2*(z - 0.85)}{(z - 1.1)} \, , \qquad (4.15)$$

where K2 is the variable gain which is determined by root locus analysis.

Substituting D1(z), H3(z), H2(z), D2(z) and H1(z) from Equations 4.3, 4.5, 4.11, 4.4 and 4.15 respectively into Equation 4.14 yields,

$$1 + \frac{0.69*(z^2 -1.7*z +.7225)}{(z-1)*(z^2 -1.599*z + 0.807)}$$

$$+ \frac{Km*(1.562*z^3 + 4.3*z^2 - 4.4587*z - 1.4033)*(z-0.85)}{(z-0.9512)*(z-1.0512)*(z^2-1.599*z+0.807)*(z-1.1)} = 0 \qquad (4.16)$$

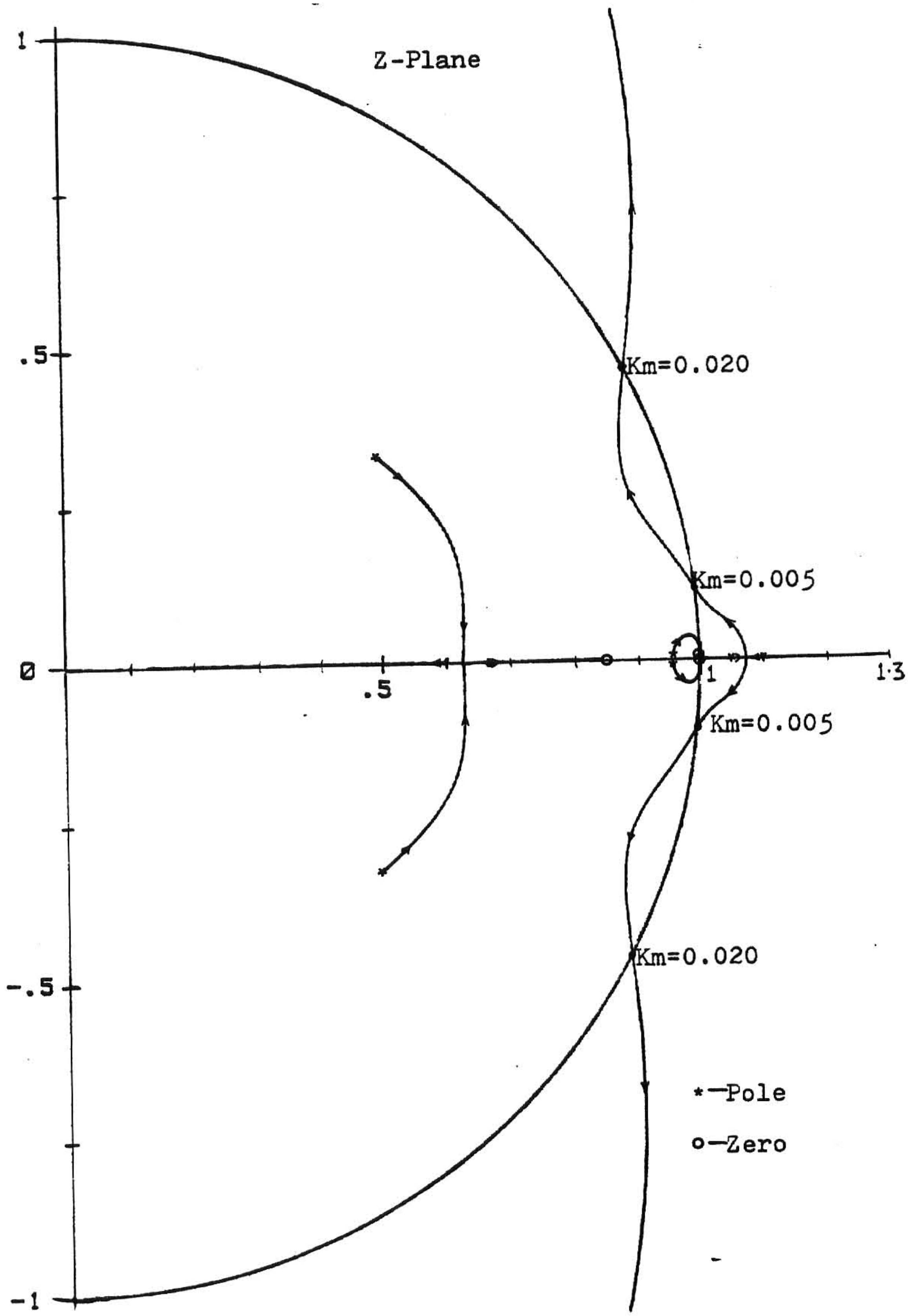

Figure 4.4 Root Locus of the Entire System in the Z-Plane

where $Km = -3.835*10^{-5}*Kp*K2$ (4.17)

Simplifying Equation 4.16, the characteristic equation of the entire system is,

$$z^6 - 5.0506*z^5 + 10.5483*z^4 - 11.7148*z^3 + 7.3588*z^2 - 2.5133*z + 0.3716 + Km*(1.562*z^5 + 1.4097*z^4 - 11.8069*z^3 + 10.5019*z^2 - 1.194*z - 1.192) . \quad (4.18)$$

A plot of the root locus of the characteristic equation of t entire system is shown in Figure 4.4. as the gain Km is varie The values of the roots are given in Appendix B. From Figure 4.4 is observed, that the poles of the characteristic equation of t system lie within the unit circle in the z plane for 0.005 < Km 0.020. Hence the pendulum cart system is stable for values of within this range.

Therefore from Equation 4.17, K2*Kp, can be deduced to l within -521.5 < K2*Kp < -130 , for the system to be stable. With assumed value of K2 equal to -256, the range of Kp is between 0. and 2.04 in order for the system to be stable. The next chapt discusses the simulation analysis, where the effect of varying t gain of the potentiometer, Kp, is studied.

CHAPTER V

SIMULATION OF THE SYSTEM

5.1 Introduction

If a model is available for a component or a system, a computer can be utilized to investigate the behaviour of the system. A computer model of the system in a mathematical form suitable for demonstrating the system's behavior may be utilized to investigate designs of a planned system without actually building the system itself. A computer simulation uses a model and the actual condition of the system being modeled and actual input commands to which the system will be subjected (6).

The simulation program was written on a Hewlett-Packard 9845B series desk top computer in Basic Language and is listed in Appendix C. This program was later modified and implemented in Assembly Language on a Z-80 microcomputer for the real time control.

The system equations are represented by recurrence equations by taking the inverse transformation as described in Section 3.1.

5.2 Recurrence Equations

The transfer functions of the system were presented in Chapter III and the compensating transfer functions to stabilize the system have been developed in Chapter IV. Figure 4.1 shows the block diagram of the system. The transfer functions are,

$$D1(z) = \frac{Vt(z)}{U(z)} = \frac{0.92*(z^2 - 1.688*z + 0.89)}{(z^2 - 1.599*z + 0.807)} \tag{5.1}$$

where D1(z) describes the relationship between the output signal of the tachometer and the control signal, U.

$$D2(z) = \frac{Vp(z)}{U(z)} = \frac{-3.835*10^{-5}*Kp*(1.562*z^3 + 4.30*z^2 - 4.4587*z - 1.4033)}{(z^4 - 3.601*z^3 + 5.009*z^2 - 3.216*z + 0.807)} \quad (5.2)$$

where D2(z) describes the relationship between the output signal from the potentiometer and the control signal, U.

$$H1(z) = \frac{C1(z)}{Vp(z)} = \frac{-256*(z - 0.85)}{(z - 1.1)} \quad . \quad (5.3)$$

where H1(z) is the digital compensator which generates the values C1, by sampling the signal, Vp, of the potentiometer.

$$H2(z)*H(3) = \frac{C2(z)}{Vt(z)} = \frac{0.75*(z^2 - 1.7*z + 0.7225)}{(z^3 - 2.688*z^2 + 2.578*z - 0.89)} \quad , \quad (5.4)$$

where H2(z)*H(3) is the digital compensator which generates the values C2, by sampling the signal, Vt, from the tachometer.

In a recurrence form the above equations can be written as,

$$Vt(k) = 1.599*Vt(k-1) - 0.807*Vt(k-2) + 0.92*(U(k) - 1.688*U(k-1) + 0.89*U(k-2)) \quad (5.5)$$

$$Vp(k) = 3.601*Vp(k-1) - 5.009*Vp(k-2) + 3.216*Vp(k-3) - 0.8075*Vp(k-4) - 3.835*Kp*10^{-5}*(1.562*U(k-1) + 4.30*U(k-2) - 4.487*U(k-3) - 1.403*U(k-4)) . \quad (5.6)$$

$$C1(k) = 1.1*C1(k-1) - 256*Vp(k) + 217.6*Vp(k-1) . \quad (5.7)$$

$$C2(k) = 2.688*C2(k-1) - 2.578*C2(k-2) + 0.89*C2(k-3)$$
$$+ 0.75*(Vt(k-1) - 1.7*Vt(k-2) + 0.7225*Vt(k-3)) . \quad (5.8)$$

The recurrence Equations 5.5 through 5.8 are used in the computer simulation.

5.3 Simulation

It is assumed that there is no delay between the time the data is sensed and the time the control signal is computed. The control signal, U, is computed based on the signals available from the potentiometer, Vp, and the tachometer, Vt. Figure 4.1 shows the digital models of the components of the system. The control signal, U, at instant k is the input to the block D1(z). D1(z) produces a signal representing the output from the tachometer, Vt, and the next block, with a transfer function D2(z)/D1(z) produces, Vp, both at time k. Vt and Vp are processed by the digital compensators H1(z) and H2(z)*H3(z) respectively. The corresponding computed digital values C1 and C2 are sent to the summer which produces a new control signal, U, at time k. This process is repeated to describe the behavior of the system.

The simulation program took into consideration the saturation of the control signal, U, equal to 10 volts. The discretization of the D/A and the A/D converters were also accounted for. The pendulum was given an initial angular displacement of 10^{o} from the upright position. Results of the simulation were obtained for various gains of the potentiometer, Kp.

From the simulation of the system, it was observed that the system was stable for a gain of the potentiometer, Kp, between 0.55 and 1.875 and unstable for all other gains. However, at the end of Chapter IV, it was concluded that the values of Kp, should lie within 0.51 and 2.04 for stability. This reduced range of Kp is due to the discretization of the sampled data and the saturation effect of the control signal which have been taken into account in the computer simulation of the system.

The simulation results of the position of the pendulum, the output control voltage, speed of the d.c. motor and displacement of the cart vs. time were plotted as shown in Figures 5.1, 5.2, 5.3 and 5.4 for a variable gain, Kp, equal to 0.55, 1.0 and 1.875.

From Figure 5.1 it is observed that the overshoot of the position of the pendulum for a gain Kp equal to 0.55 is large compared to a gain of Kp equal to 1. For a gain of Kp equal to 1.875 ,the position of the pendulum is observed to fluctuate more compared to gains of Kp equal to 0.55 and 1.0. For Kp equal to 1.0, and with an initial angular displacement of the pendulum equal to 10^{o} ,the pendulum is brought to within 1^{o} of the upright position within 0.75 sec, and is limited to this range for all greater time.

Figure 5.2 shows the simulation results for the output control voltage vs. time, for a variable gain, Kp. From the graphs it can be deduced that for gains of potentiometer, Kp, equal to 0.55 and 1.0 less control voltage is required to stabilize the system. However for gains of Kp greater than 1.875 the saturation of the control signal makes the system unstable.

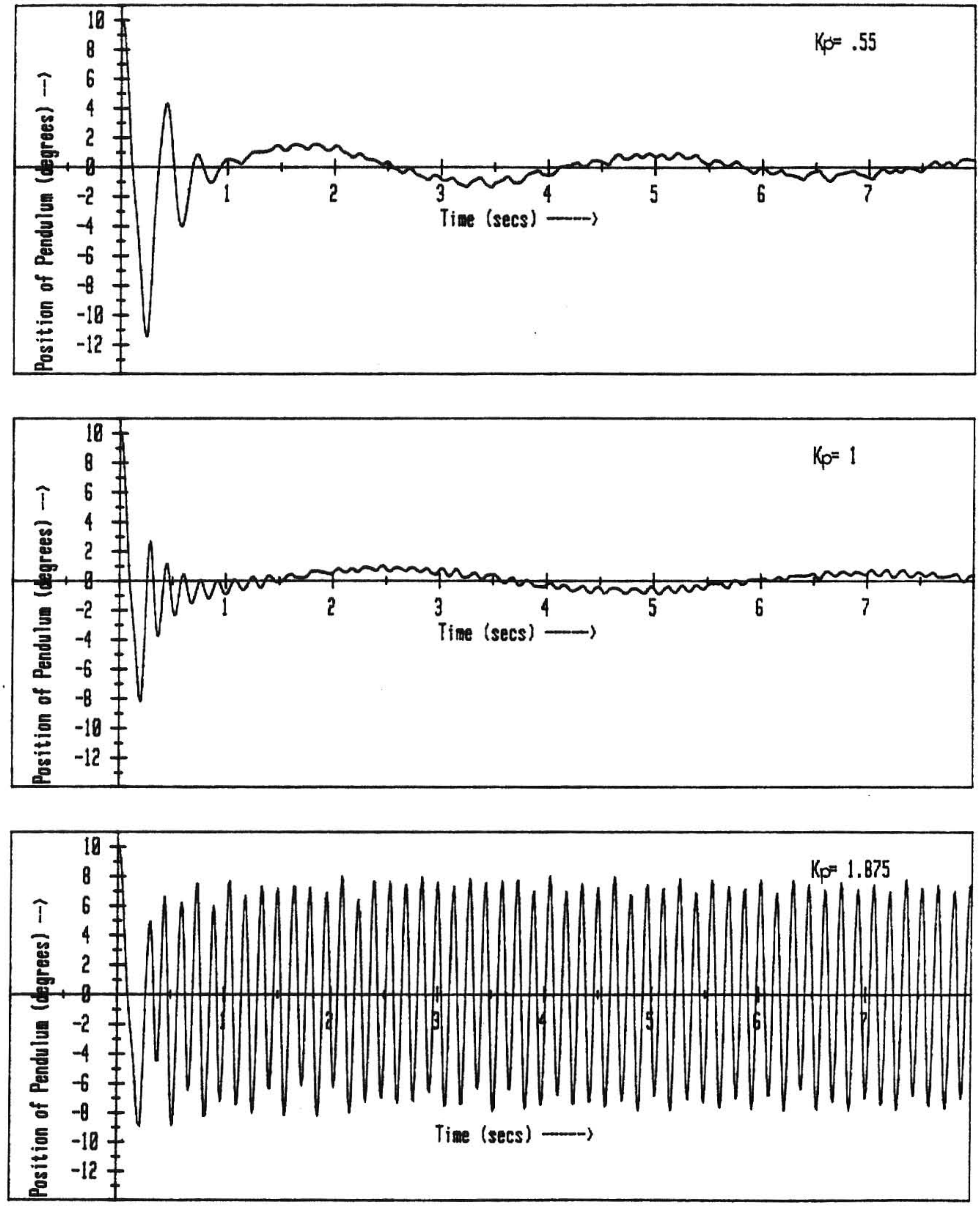

Figure 5.1. Plot of the Angular Position of the Pendulum vs Time for potentiometer Gain Equal to 0.55, 1 and 1.875

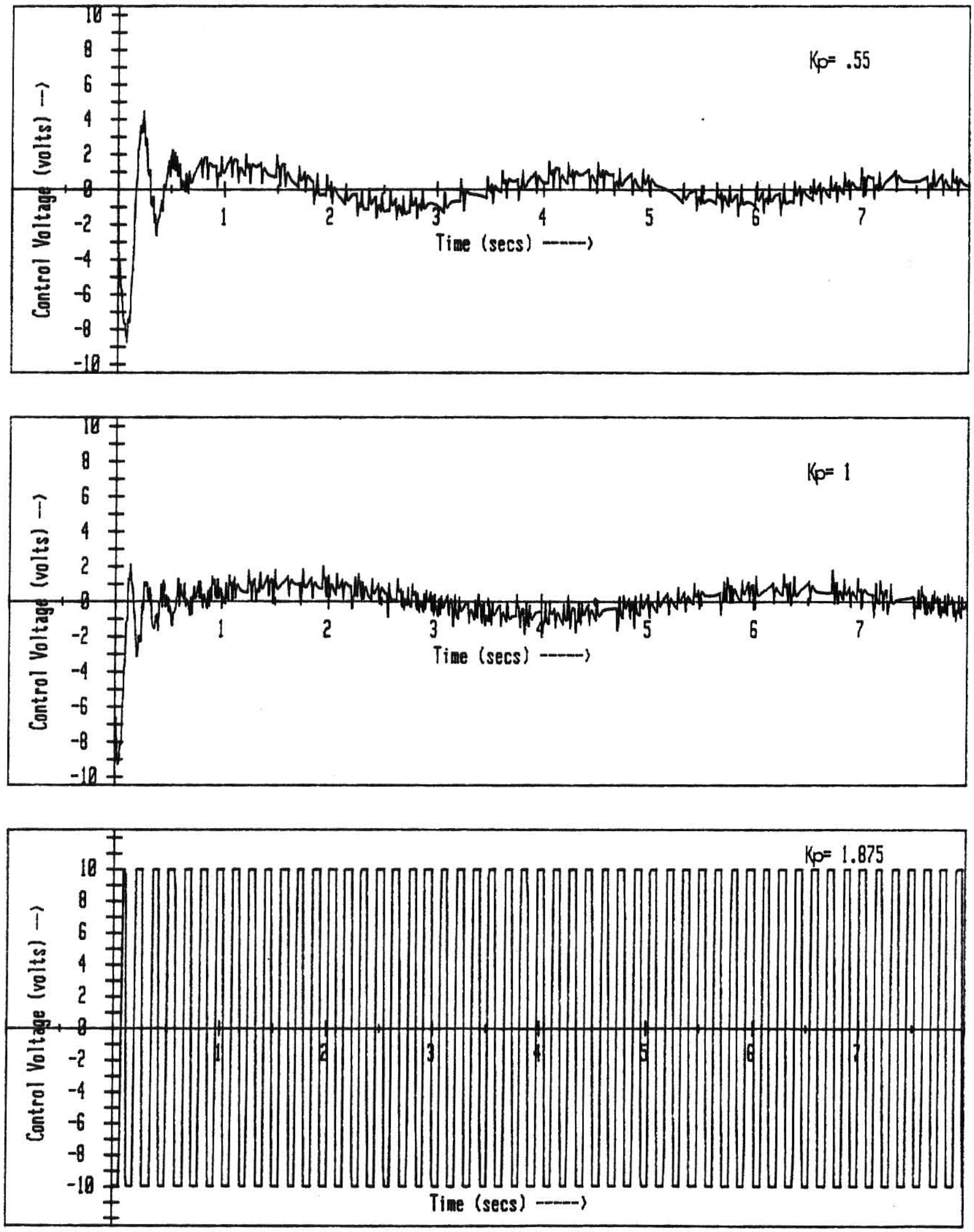

Figure 5.2. The Plot of Control Voltage vs Time for a Potentiometer Gain of 0.55, 1 and 1.875

Refering to Figure 5.3, the plots of the speed of the d.c. motor shows a very small change for increasing Kp from 0.55 to 1.0, but for a value of Kp equal to 1.875 the response is oscillatory, which is consistent with the inference from Figures 5.1 and 5.2.

Figure 5.4 shows the results of the displacement of the cart vs. time. It is observed that changes in gain, Kp, do not have significant effects on this parameter.

From the above analysis of the plots, it was decided to set the values of the gain Kp equal to 1.0 for the experiment.

The next chapter deals with the development of the assembly language real time control program.

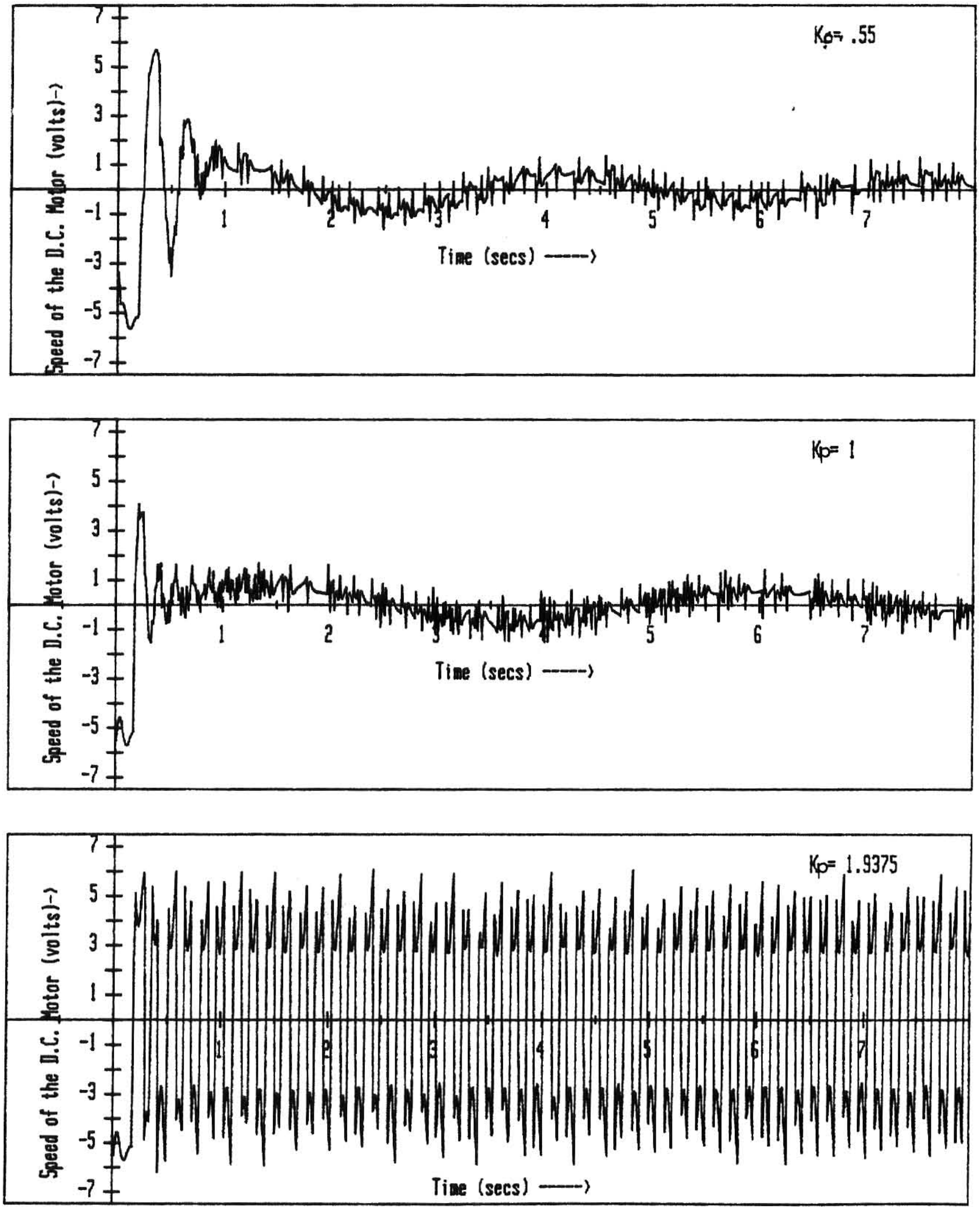

Figure 5.3. The Plot OF Speed of D.C. Motor (volts) vs Time
for a Potentiometer Gain of 0.55, 1 and 1.875

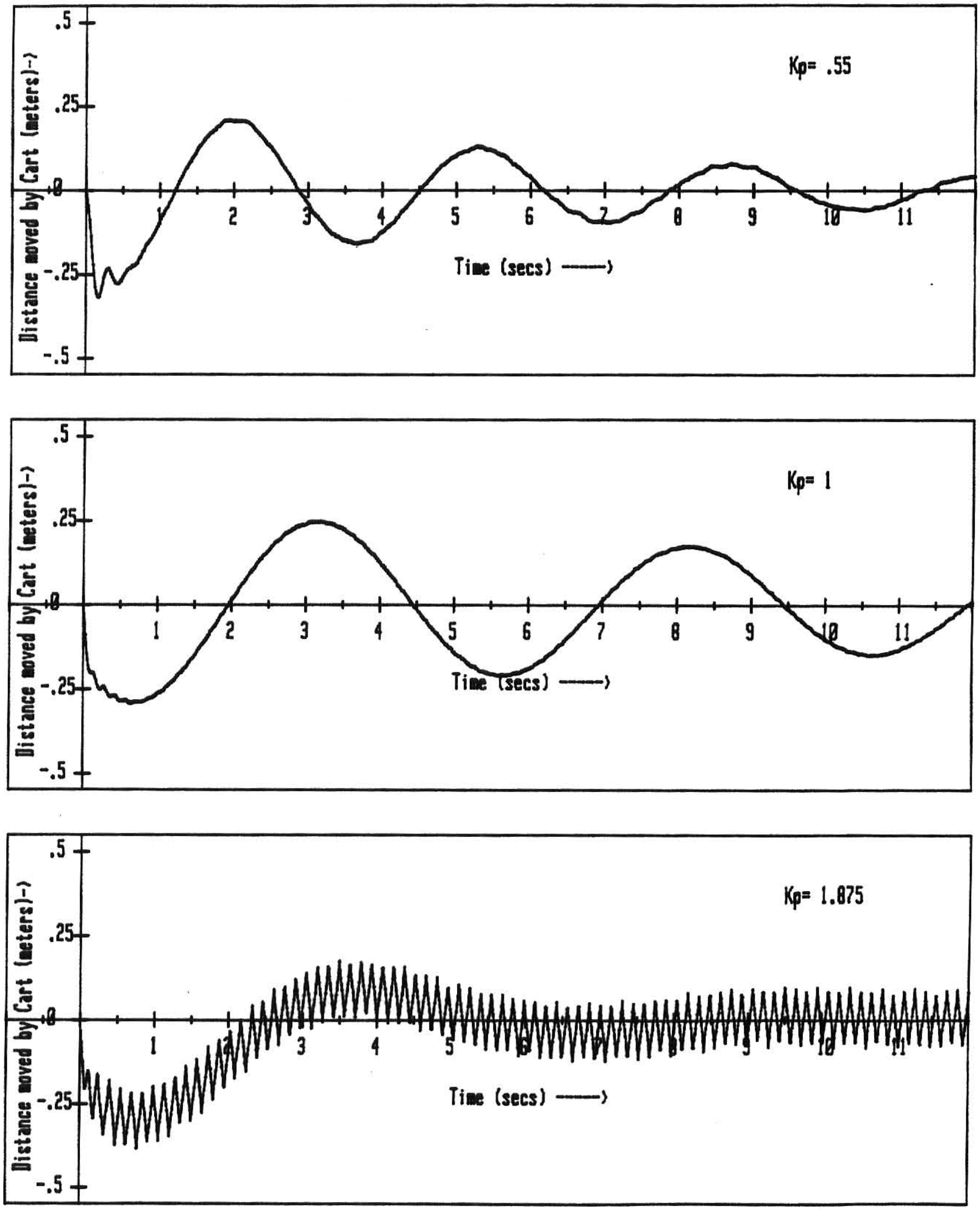

Figure 5.4. The Plot of Displacement of the Cart (meters) vs Time for a Potentiometer Gain of 0.55, 1 and 1.075

CHAPTER VI

SOFTWARE

6.1 Introduction

The computer used in this research was a Z-80 from Digital Group. The real time control of the system was done by this computer. The control program was written in assembly language. The Assembler occupies addresses from 0 to 2FFF in hexadecimal. The programmer source and object code could be placed at any address above the assembler. Additional facilities were available to store and retrive the programs from a floppy disk or an audio cassette. Listing of the program was possible on a printer.

The software developed for this research was one of the major tasks. The programming was done in assembly language. Since memory was limited, the software was divided into many parts and stored on the disk. The object code of each part was assembled and stored separately. These object code parts were later linked in the correct order. The memory used by the object code was 2.2 K. The object code was placed from address 3000 Hex to 38D3 Hex.

A floating point binary representation with a 15 bit mantissa and an 8 bit exponent was used for numerical calculations. This provided a resolution of 1 part in 32,764. By using this type of representation the accuracy of computation was maintained.

The control program consists of a main program and subroutines. Memory addresses 3E00H to 3E7FH were used for storing the coefficients, variables, and parameters of the program. A memory map that shows the memory used in the first page of 3E is given in Appendix D. The starting addresses of the main program and the subroutines are also shown in Appendix D.

6.2 Main Program

A flow chart of the main program is given in Figure 6.1 and the assembly langauge program is listed in Appendix E. The memory used for the object code of the main program is 0.94 K. bytes.

The function of the main program is to read the digital signals of the speed of the d.c. motor and the position of the pendulum from the A/D converter, perform specific computations, and send a digital control signal to the D/A converter. The main program is divided into three parts:

1) The Initialization,
2) The Computations, and
3) The Updating of the Parameters.

6.2.1 Initialization

The flow chart of this program is shown in Figure 6-2. The first step of this program is to initialize the output voltage from the D/A converter to zero, so that the cart remains stationary. The next operation resets memory block 3E00H - 3E7FH to zero. Then the subroutine INITI and PENDIS are executed. By executing the INITI subroutine, the zero reference of the pendulum position and the bias are determined. The zero

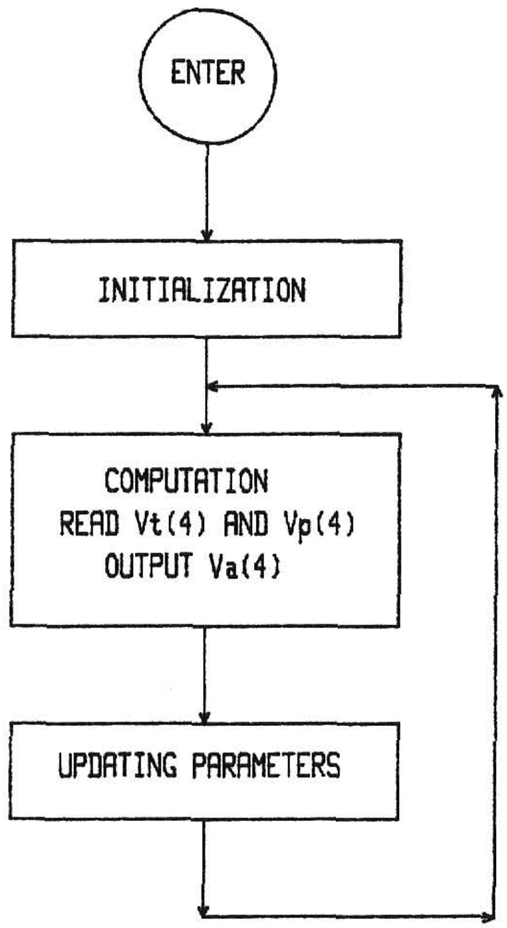

Figure 6-1 THE MAIN PROGRAM

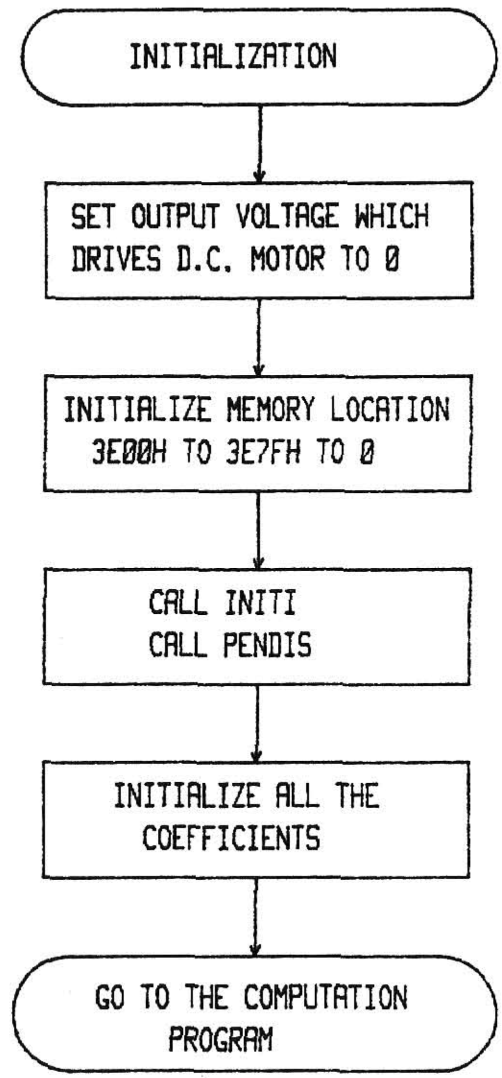

Figure 6-2 THE INITIALIZATION PROGRAM

reference is the output voltage from the potentiometer corresponding to the vertical position of the pendulum. The subroutine PENDIS helps in fixing the limits of the angular displacement of the pendulum. Details of these two subroutine are given in Section 6-3.

The next operation stores all the coefficients and the variables required for the software program into appropriate memory locations.

The initialization program is executed once at the beginning of each experiment. The execution time of this program is not critical because it is not a part of the sampling period.

6.2.2 Computation

The computation program is the most important part of the main program. This program samples the velocity of the cart and the position of the pendulum from the A/D converter and performs computations of the equations shown below for generating a control signal U(4).

$$C1(4) = 1.1*C1(3) - 256*Vp(4) + 217.6*Vp(3) \quad (6.1)$$

$$C2(4) = 2.69*C2(3) - 2.58*C2(2) +0.89*C2(1) + 0.75*Vt(4) -1.275*Vt(3) + 0.542*Vt(2) \quad (6.2)$$

$$U(4) = C1(4) - C2(4). \quad (6.3)$$

The coefficients in the above equations are obtained based on the computer simulation, as derived in the previous chapter. The values of the coefficients, which are in decimal, are

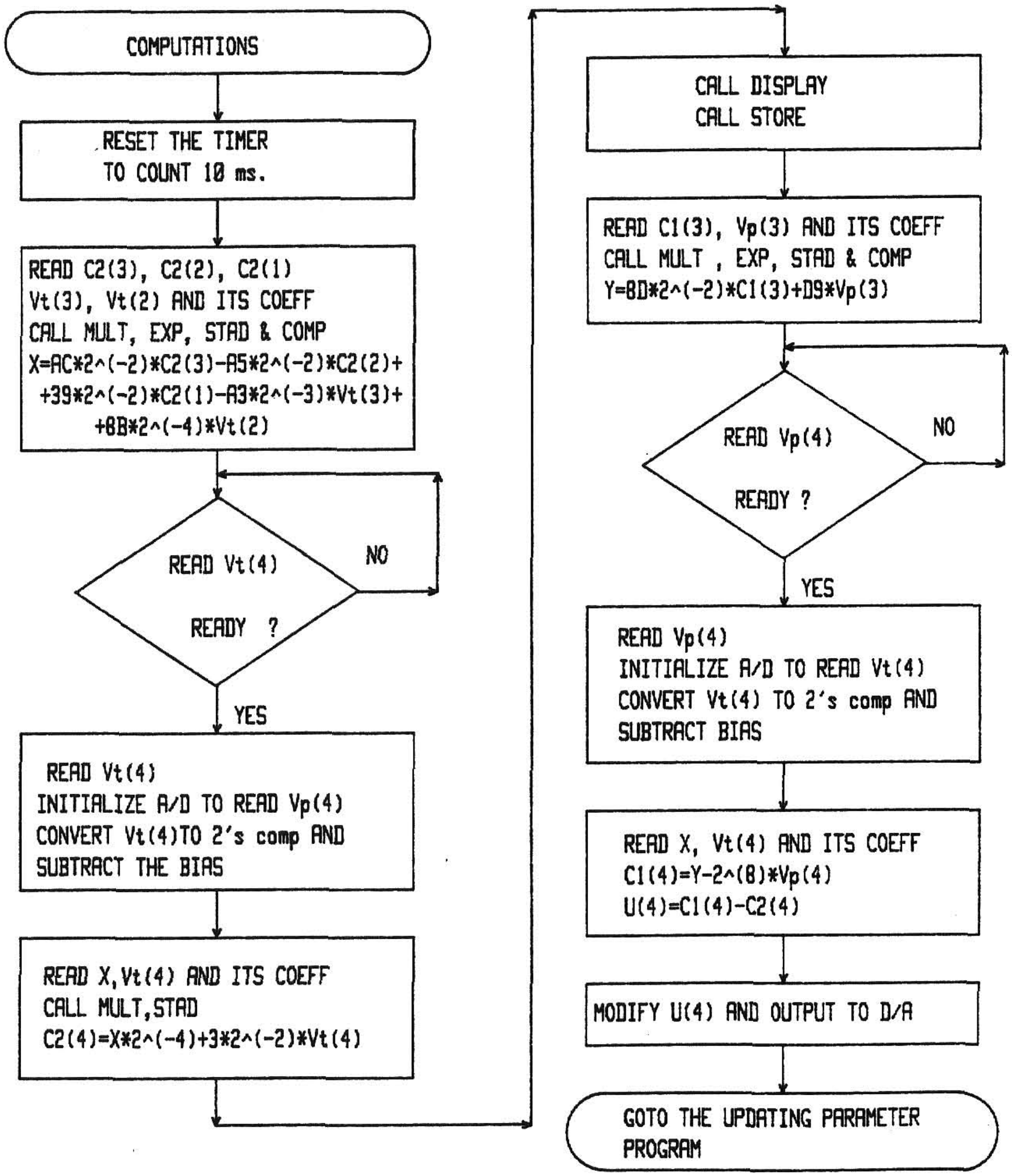

Figure 6-3 THE COMPUTATION PROGRAM

converted to hexadecimal and stored in the memory location shown in Appendix D.

The equations are represented in hexadecimal as,

$$C1(4) = 8D*2^{-7}*C1(3) + D9*Vp(3) - 2^{8}*Vp(4) \quad (6.4)$$

$$C2(4) = AC*2^{-2}*C2(3) - A5*2^{-2}*C2(2) + 39*2^{-2}*C2(1) - \\ -A3*2^{-3}*Vt(3) + 8B*2^{-4}*Vt(2) \ *2^{-4} + 3*2^{-2}*Vt(4). \quad (6.5)$$

$$U(4) = C1(4) - C2(4) . \quad (6.6)$$

The flow chart of the computation program is shown in Figure 6-3. The first step of this program resets the timer to count 10 ms. The next step performs the computation of,

$$X = AC*2^{-2}*C2(3) - A5*2^{-2}*C2(2) + 39*2^{-2}*C2(1) - A3*2^{-3}*Vt(3) + \\ + 8B*2^{-4}*Vt(2). \quad (6.7)$$

In the next operation, the A/D is initialized to read Vt(4) and the 'BUSY' signal of the A/D converter is checked to verify if the signal Vt(4) has settled down. If signal Vt(4) has settled, then it is read by the A/D converter. The A/D converter is initialized to read the signal of the position of the pendulum, Vp(4). The value of Vt(4) is converted to 2's complement form and adjusted by subtracting or adding the bias. Then, Vt(4) is stored in memory and computations are performed to yield,

$$C2(4) = 2^{-4}*X + 3*2^{-2}*Vt(4). \quad (6.8)$$

In the next operation Subroutine DISPLAY is executed to display the the position of the pendulum, speed of the d.c. motor, and the output control signal on the CRT. Then Subroutine STORE is executed to store the data Vt(4), Vp(4), U(4), C1(4), and C2(4).

In the next operation the computation performed is,

$$Y = 8D*2^{-7}*C1(3) + D9*Vp(3). \tag{6.9}$$

The A/D is initialized to read Vp(4). Then the 'BUSY' signal is checked until the value of Vp(4) is ready to be read. The next operation reads Vp(4) from the A/D converter. The A/D is now initialized to read the speed of d.c. motor, Vt(4). The value of Vp(4) is converted to 2's complement form and adjusted by subtracting or adding the bias. Then Vp(4) is stored in memory and computations are performed to yield,

$$C1(4) = Y - 256*Vp(4). \tag{6.10}$$

The control signal U(4) is obtained by solving the equation,

$$U(4) = C1(4) - C2(4) . \tag{6.11}$$

The control signal calculated in Equation 6.11 is in 2's complement form with a 24 bit mantissa. This value is modified and sent to the D/A converter.

6.2.3 Updating Parameters

The parameters have to be updated before executing the computation program in the next sampled period. By executing

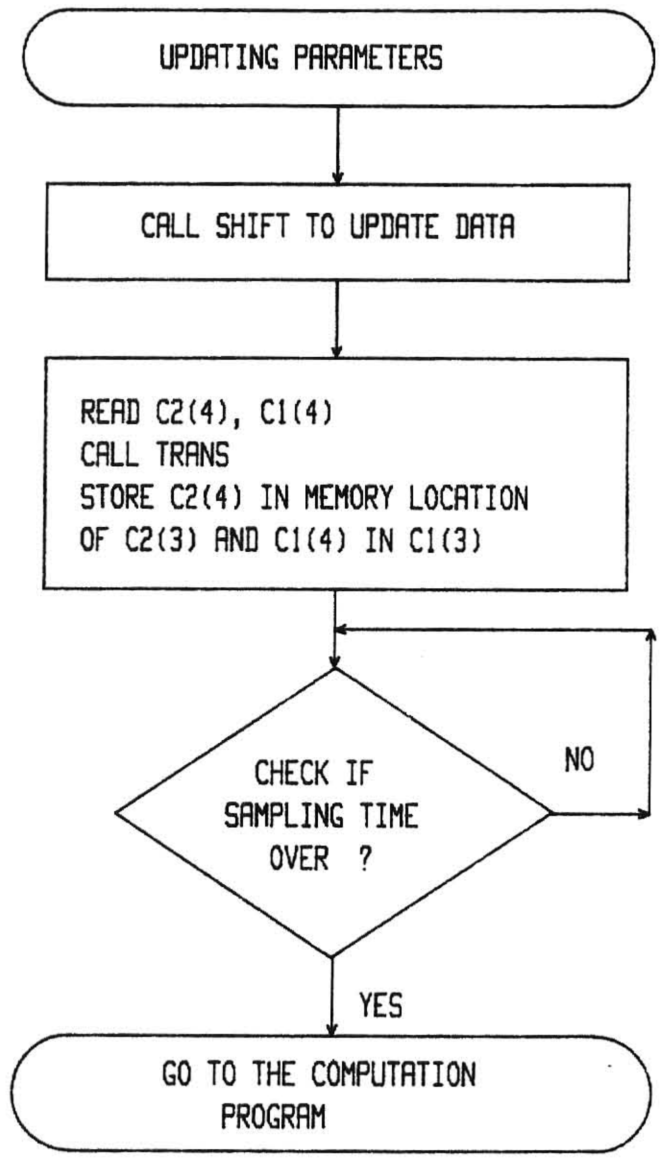

Figure 6-4 THE UPDATING PARAMETER PROGRAM

Subroutine SHIFT explained in Section 6.3.6, the values shift one step down; that is the data of Vt(3) shifts to Vt(2) and the computed values C2(3) and C2(2) to C2(2) and C2(1) respectively. The value of C1(4) and C2(4) are in 2's complemnent form with 24 bit mantissa. These values are converted into the 15 bit mantissa, a sign byte and an exponent byte by executing Subroutine TRANS which is explained in Section 6.3.5. The converted value of C1(4) is stored in memory locations C1(3) and C2(4) is stored in memory locations C2(3).

The next step in the program is to check if the 10 ms sampling time is over. If the sampling time is over, then the operation will return back to the computation program.

6.3 Subroutines

The main program is supported by eleven subroutines. They are, a fifteen by eight bit multiplication routine; a 24 bit signed addition routine; a routine for performing the 2's complement operation; a routine to shift data; a routine for converting a value from a fixed point representation to floating point; a routine to update values; a routine to halt the execution of the program; a routine to display data on the CRT; a routine to store the data in specific memory blocks; a routine to find the zero reference of the pendulum when it is in the upright position and a routine to adjust the range of the angular movement of the pendulum about the vertical axis.

6.3.1 Subroutine MULT

This subroutine performs the multiplication of a 15 bit by an

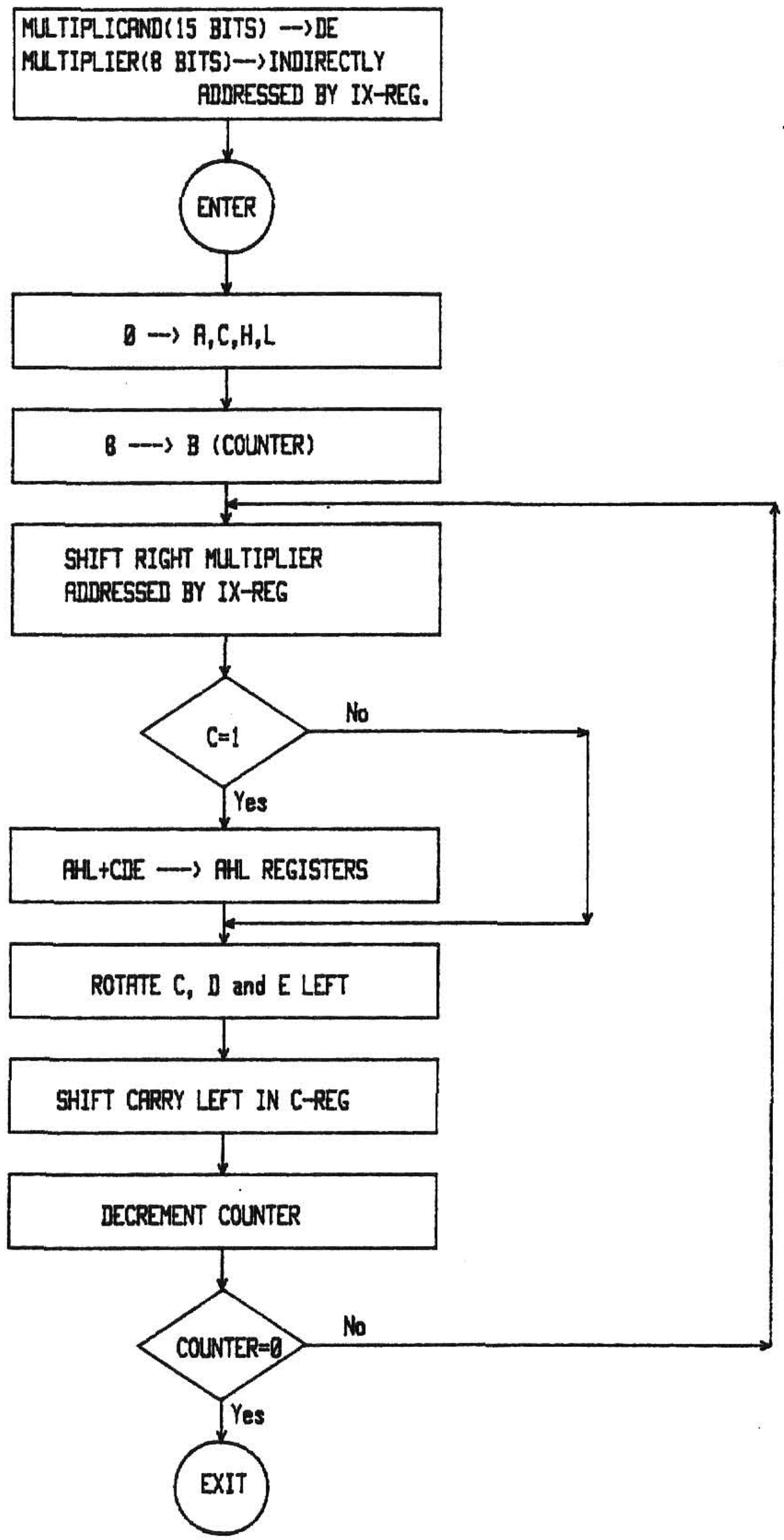

Figure 6.5 SUBROUTINE MULT

8 bit unsigned binary number. Figure 6.5 shows the flow chart of this subroutine. Basically, the multiplication routine is a series of tests and shifts of the multiplier and multiplicand.

To use this subroutine, the 15 bit multiplicand is loaded into the DE register pair and the 8 bit multiplier is indirectly addressed by the IX register A, B, C, D, E, H and L registers are utilized by this subroutine. The maximum time for performing a multiplication is 0.298 ms. The final result is placed in the CHL registers.

6.3.2 Subroutine STAD

This subroutine converts an unsigned 23 bit binary value stored in the CHL registers into a signed value. The sign bit of the 23 bit unsigned data is loaded in the least significant bit of the A register. Then this subroutine adds the two 24 four bit signed values which are loaded in the CHL and C'H'L' registers. The result of the adding operation is placed in the C'H'L' registers. A, C, D, E, H, L, C', D', E', H' and L' registers are utilized by this subroutine. A flow chart of this subroutine is given in Figure 6.6.

6.3.3 Subroutine COMPLEMENT

Figure 6.7 shows the flow chart of this subroutine. This subroutine converts a twenty-three bit binary number loaded in the CHL registers to 2's complement form. The result of this operation is placed in the CHL registers. A, C, H and L registers are utilized by this routine.

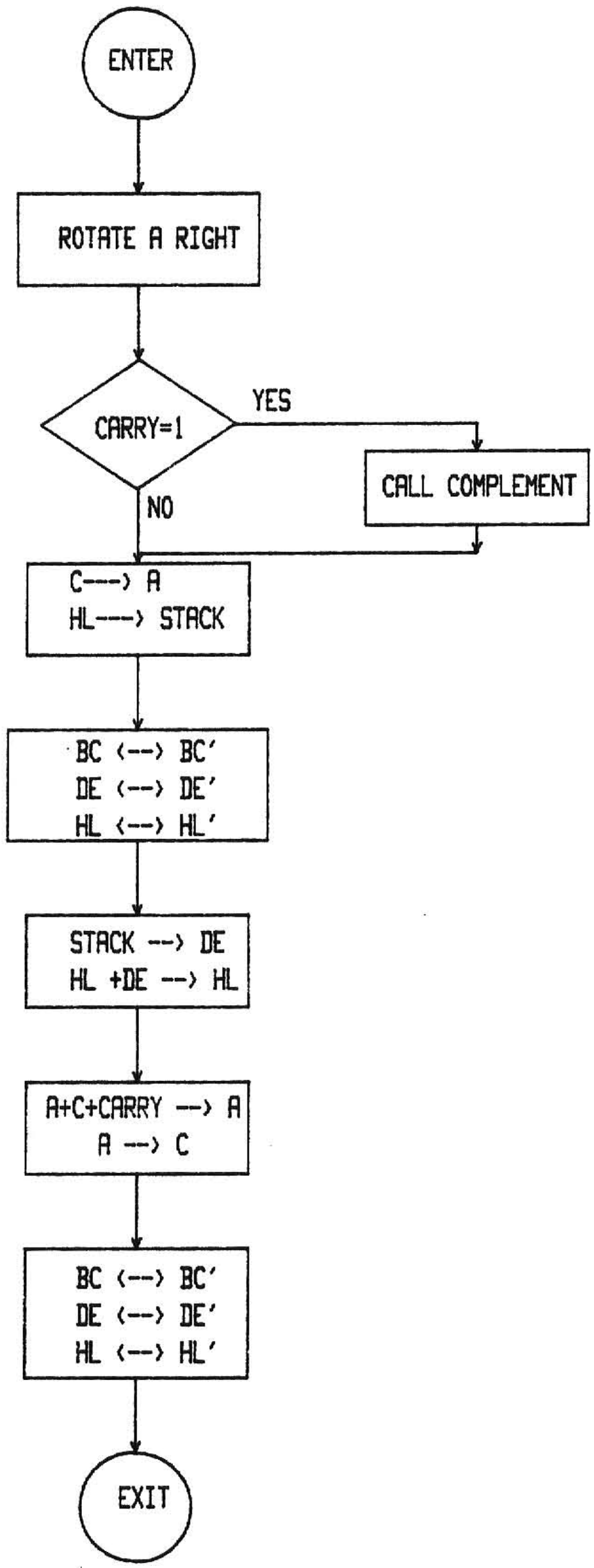

Figure 6.6 SUBROUTINE STAD

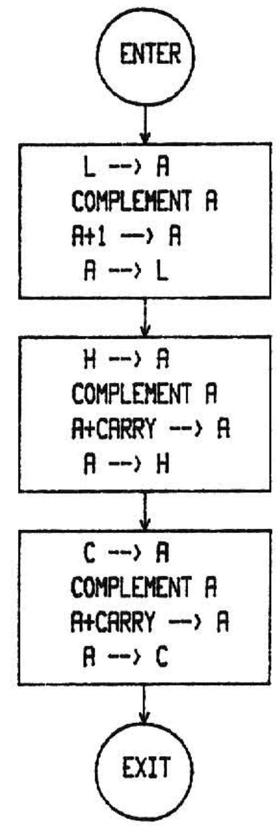

Figure 6.7 SUBROUTINE COMPLEMENT

6.3.4 Subroutine EXPONENT

The subroutine Exponent divides a twenty-three bit binary number stored in CHL registers by four. It then shifts the remaining value in the CHL registers to the left a number of bits depending on the value placed in the B register before this subroutine is executed. The final result is placed in the CHL registers. Figure 6.8 shows the flow chart of this subroutine.

6.3.5 Subroutine TRANS

This subroutine converts a signed three bytes value loaded in the CHL registers into a fifteen bit unsigned mantissa stored in the HL registers, a sign byte, and an exponent byte. The sign byte is placed in the memory location pointed to by the IY register and the exponent byte is placed in the memory location pointed to by IX register. Before calling this subroutine, the value is loaded in the CHL registers and the IX and IY are loaded. A, C, D, H and L registers are utilized by this subroutine. A flow chart of this subroutine is shown in Figure 6.9.

6.3.6 Subroutine SHIFT

Figure 6.10 shows the flow chart for this subroutine. The updating of the values in the program is done by this subroutine.

This subroutine shifts the magnitude and sign of Vt(3) to memory location for Vt(2); the magnitude, sign and exponent byte of C2(2) to C2(1) and C2(3) to C2(2). A, H and L registers are utilized by this subroutine.

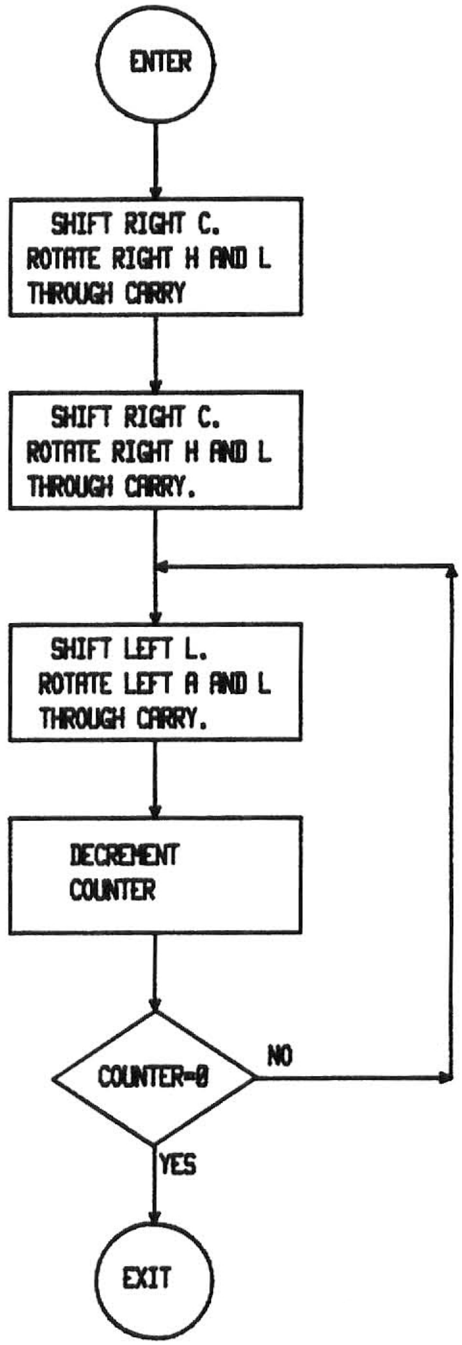

Figure 6.8 SUBROUTINE EXPONENT

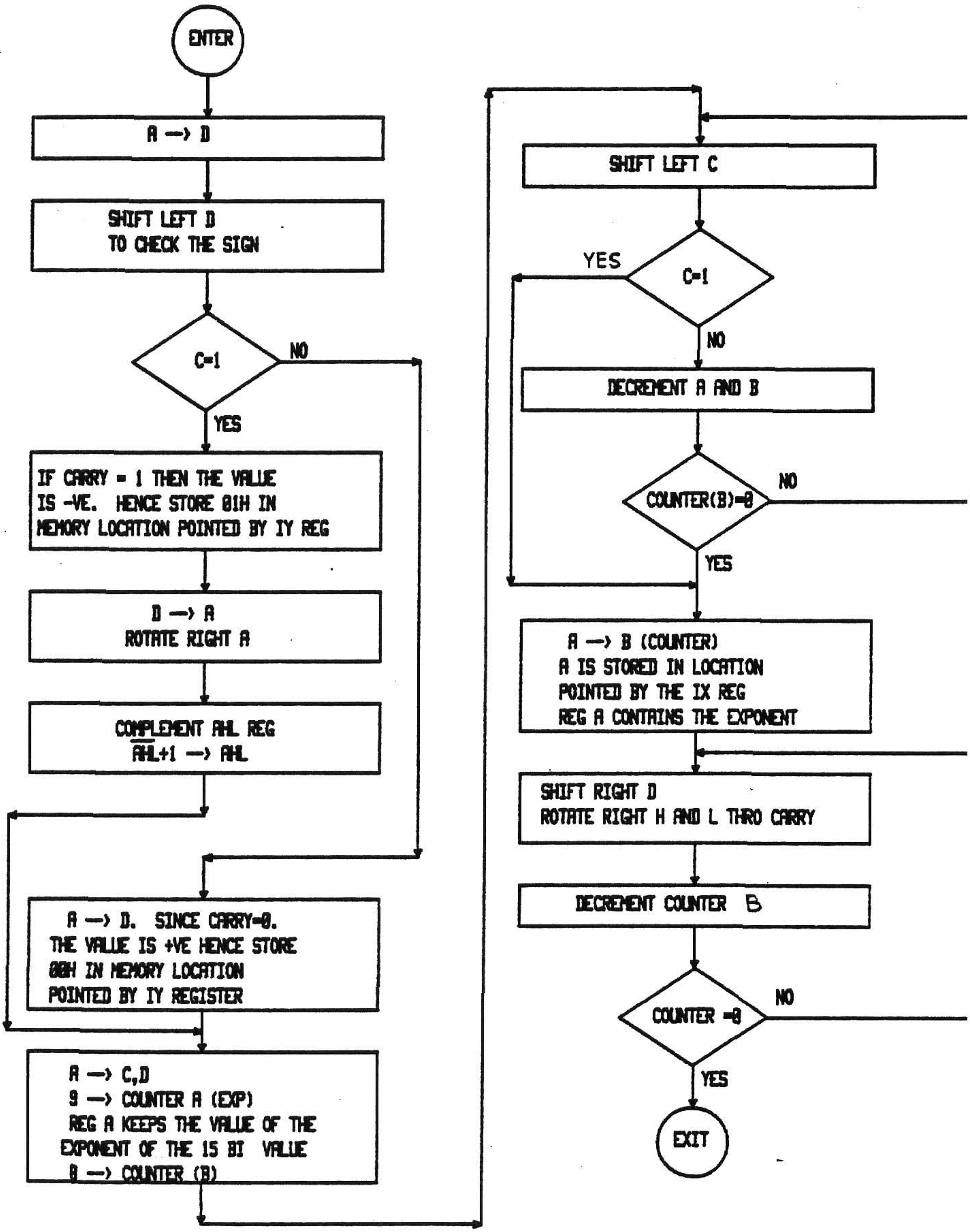

Figure 6.9 SUBROUTINE TRANS

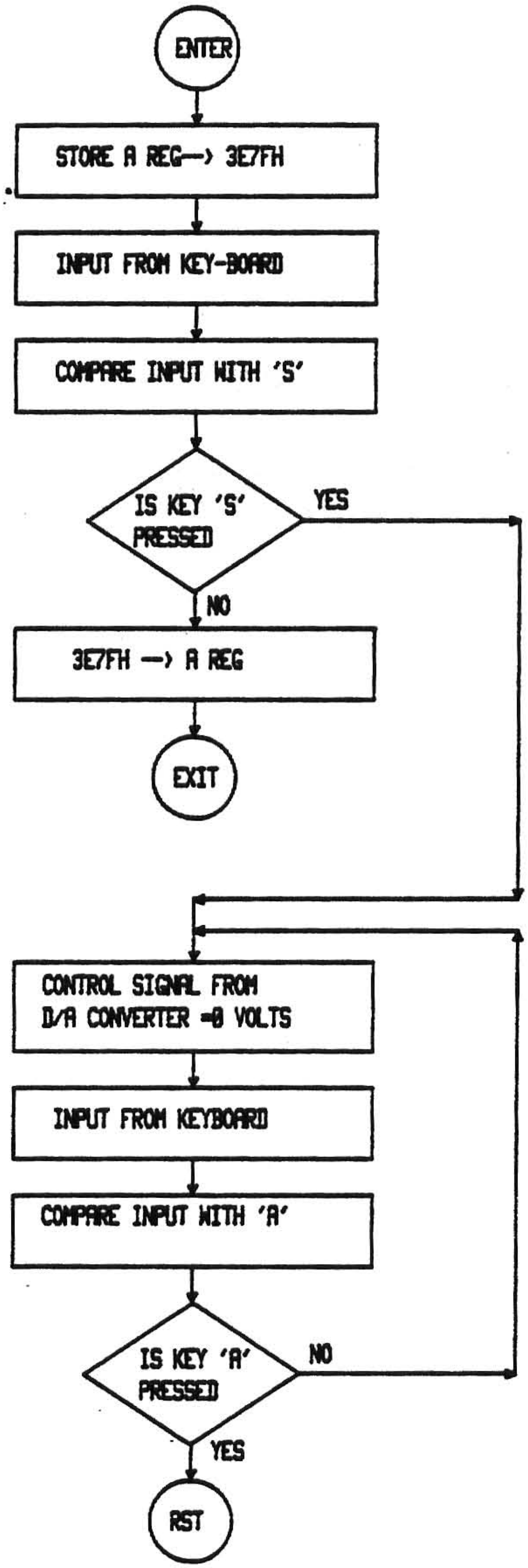

Figure 6.11 SUBROUTINE STOP

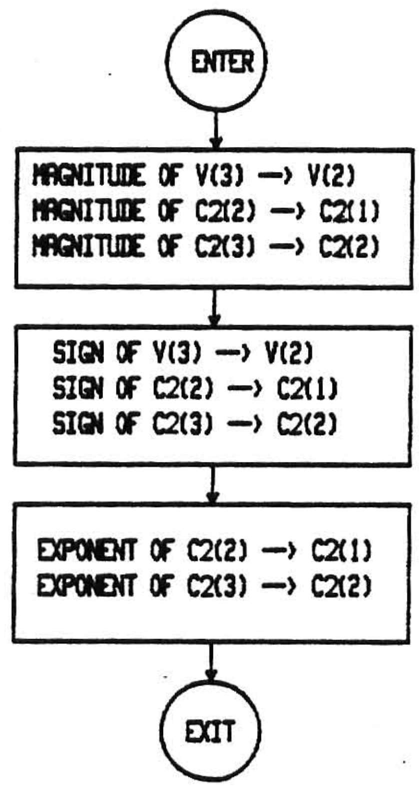

Figure 6.10 SUBROUTINE SHIFT

6.3.7 Subroutine STOP

This subroutine is used to halt the execution of the main program. The flow chart of this subroutine is shown by Figure 6.11.

If key 'S' is depressed from the keyboard when the main program is being executed , the entire execution of the program is terminated. At the same instant the control voltage output from the D/A converter, which is used to drive the d.c. motor, is initialized to 0. The past values of the position of the pendulum, velocity of cart and the control voltage remain displayed on the screen. If the key 'A' is also depressed, the system is reset and then the contents of the register are displayed on the CRT.

6.3.8 Subroutine DISPLAY

This subroutine displays the pendulum position , speed of the motor and the control voltage output from the D/A converter on the CRT. This was useful in debugging the software and to verify the correct functioning of the hardware. Signed binary values that represent the pendulum position, speed of the motor and the control voltage are converted to a hexadecimal ASCII codes and then displayed on the CRT with positive or negative signs. Figure 6.12 shows the flow chart of this subroutine.

6.3.9 Subroutine STORE

The store routine is written to store the data of Vp, Vt,

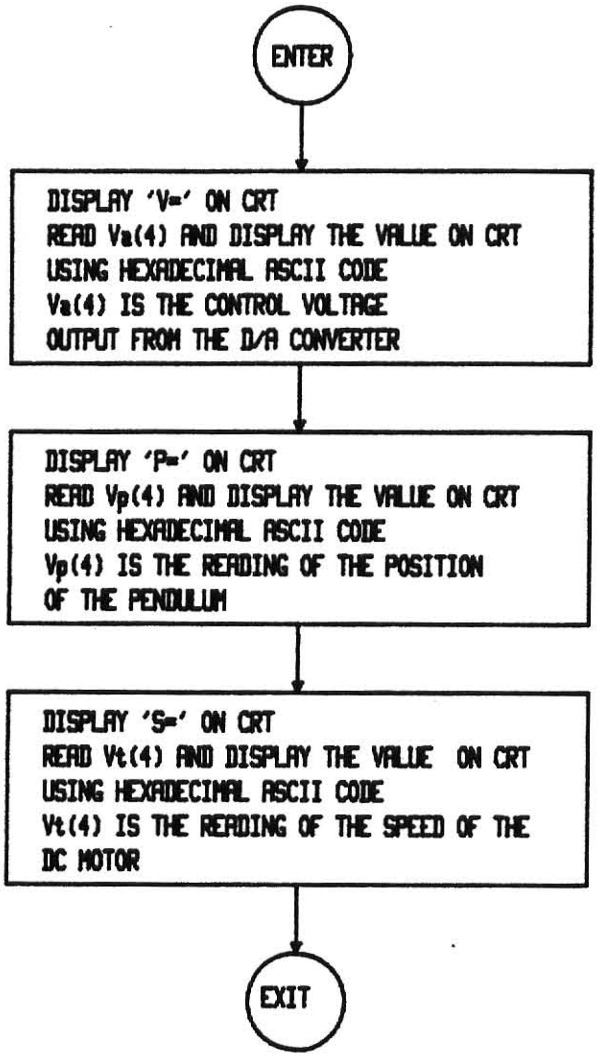

Figure 6.12 SUBROUTINE DISPLAY

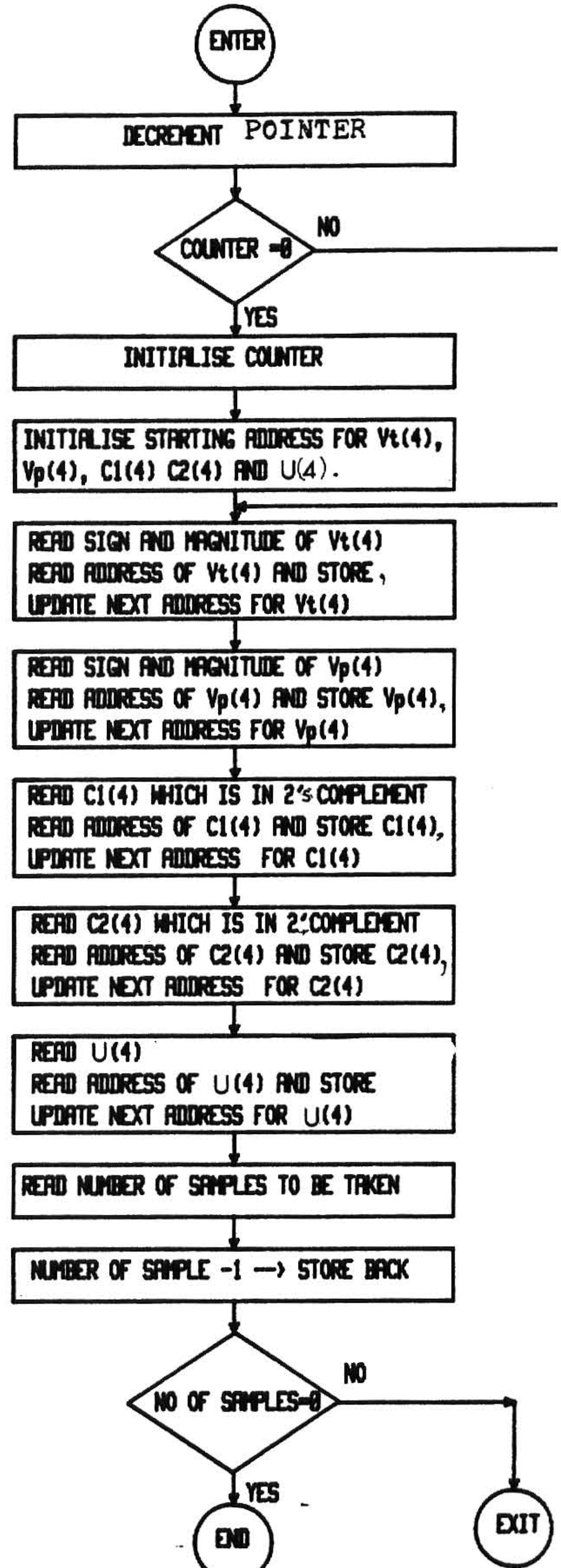

Figure 6.13 SUBROUTINE STORE

C1, C2 and U in certain memory blocks. This routine is used for error diagnostics and data acquisition.

The signal from the tachometer, Vt, is stored from memory locations 3E80H to 4000H and the signal from the potentiometer, Vp, is stored from memory locations 2E80H to 3000H. Each of these data is stored in three bytes. The first two bytes contain the magnitude and the third byte contains the sign. For the sign byte, 01H indicates negative values while 00H indicates positive values.

Computed values C1(4) are stored from 3980H to 3B00H, C2(4) from 3B00H to 3C80H and U(4) from 3C80H to 3E80H. Each of these values is stored in three bytes and are signed numbers.

The subroutine also stops the execution of the program after a certain number of sampling events have passed. The desired number of sampling events are stored in memory locations 3E44H and 3E43H. A, D, H, and L registers are utilized by this subroutine. The flow chart of this subroutine is shown in Figure 6.13. When the subroutine is entered a pointer is decremented. The pointer gives the address off set for storing these values.

6.3.10 Subroutine INITI

This subroutine is executed before the main program is executed. This subroutine has further access to three more subroutines CHECH, INC, and DEC. Figures 6.14 and 6.15 show the flow chart for the subroutines.

This subroutine initializes the reading of the position of the pendulum to 0 when the pendulum is almost vertical by a bias

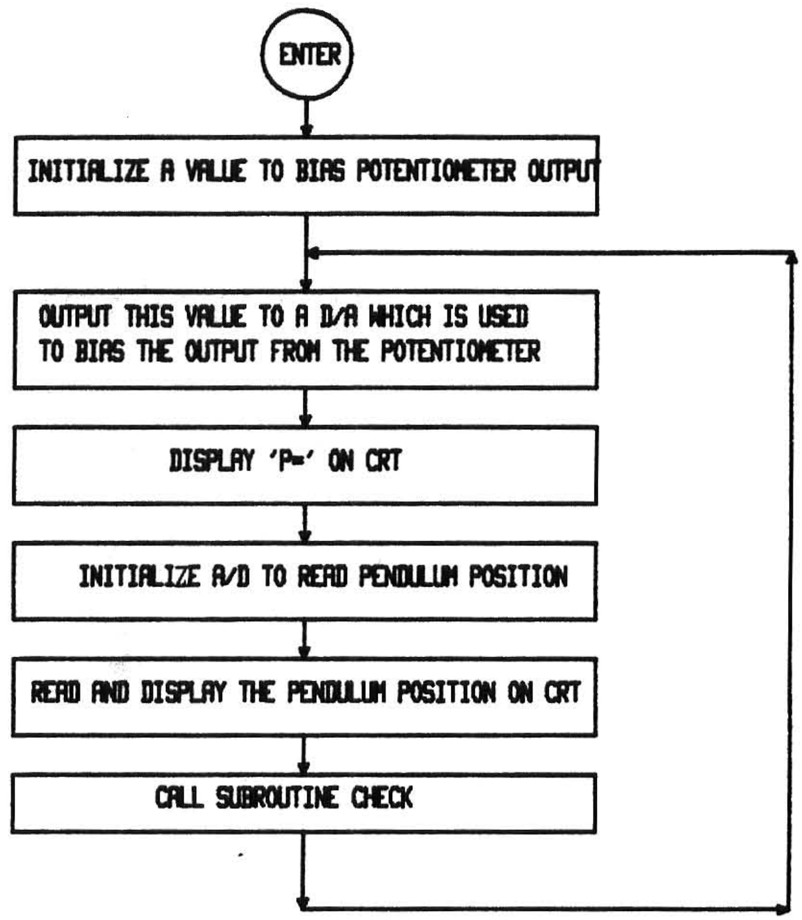

Figure 4.14 SUBROUTINE INITI

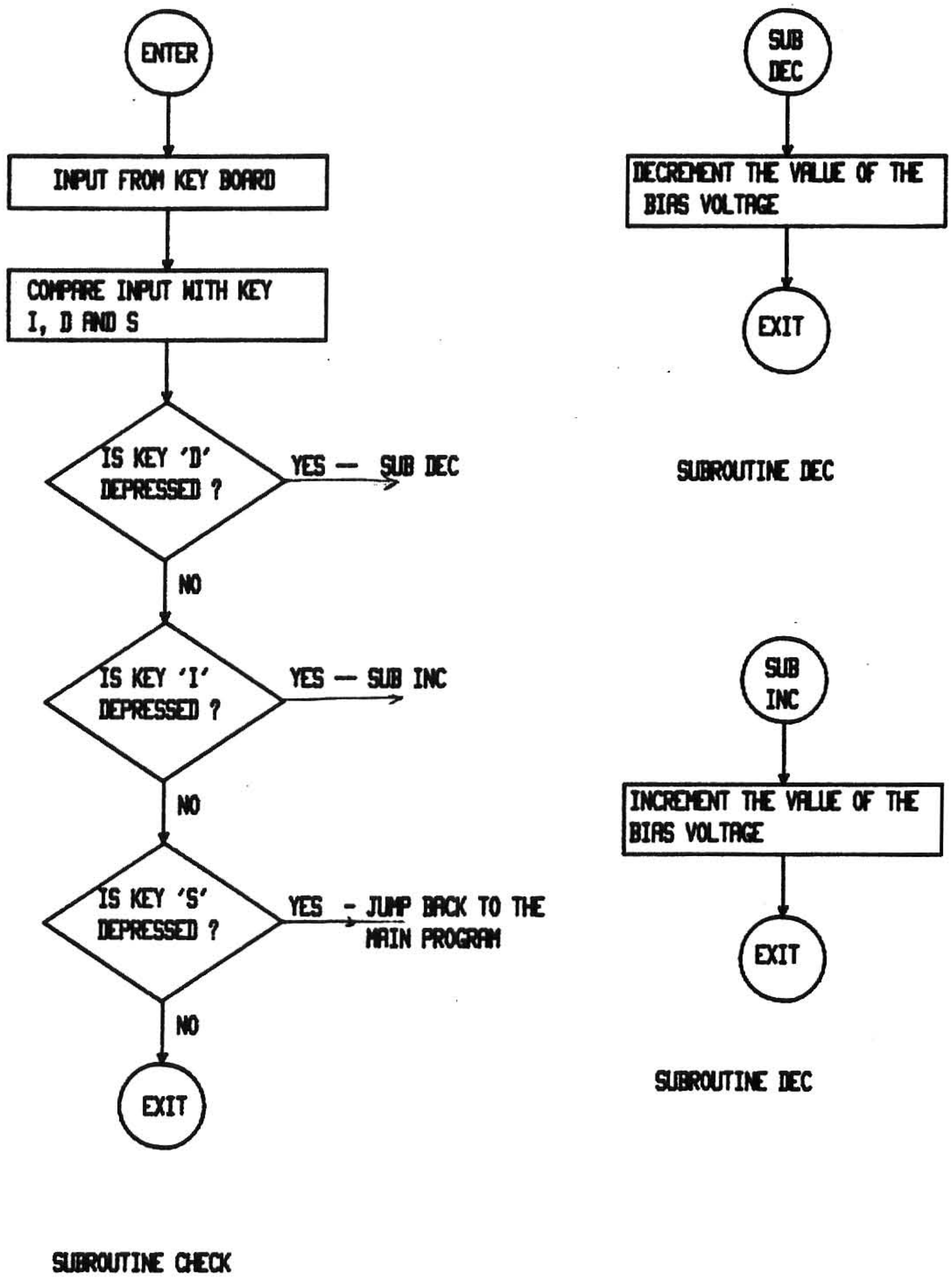
ENTER
INPUT FROM KEY BOARD
COMPARE INPUT WITH KEY I, D AND S
IS KEY 'D' DEPRESSED ?
YES — SUB DEC
NO
IS KEY 'I' DEPRESSED ?
YES — SUB INC
NO
IS KEY 'S' DEPRESSED ?
YES - JUMP BACK TO THE MAIN PROGRAM
NO
EXIT
SUBROUTINE CHECK
SUB DEC
DECREMENT THE VALUE OF THE BIAS VOLTAGE
EXIT
SUBROUTINE DEC
SUB INC
INCREMENT THE VALUE OF THE BIAS VOLTAGE
EXIT
SUBROUTINE DEC

Figure 6.15

from the D/A converter which is the input to the A/D converter. The A/D converter is set to be in differential mode with one input from the potentiometer and the other from the D/A converter.

The values of the current readings of the position of the pendulum and the reading of the velocity of the cart are displayed on the CRT. The program periodically executes subroutine CHECK which checks if key 'I','D' or 'S' is depressed. If the key 'I' is pressed then the output voltage from the D/A is increased for biasing the reading of the pendulum position by subroutine INC. If the key 'D' is pressed then the output voltage from the D/A is decreased for biasing the reading of the pendlum position by subroutine DEC. Since the value of the position of the pendulum is displayed, the biasing can be adjusted continuously. Since it is extremely difficult to get an exact 0 reading from the A/D converters when the cart is stationary, the zero error readings of the pendulum position and velocity of the motor are stored. These error readings are used as correction factors for correcting the readings of the pendulum position and the velocity of the armature when the main program is executed.

Key 'S' is depressed to return back to the main program.

6.3.11 Subroutine PENDIS

This subroutine is executed immediately after execution of subroutine INITI.

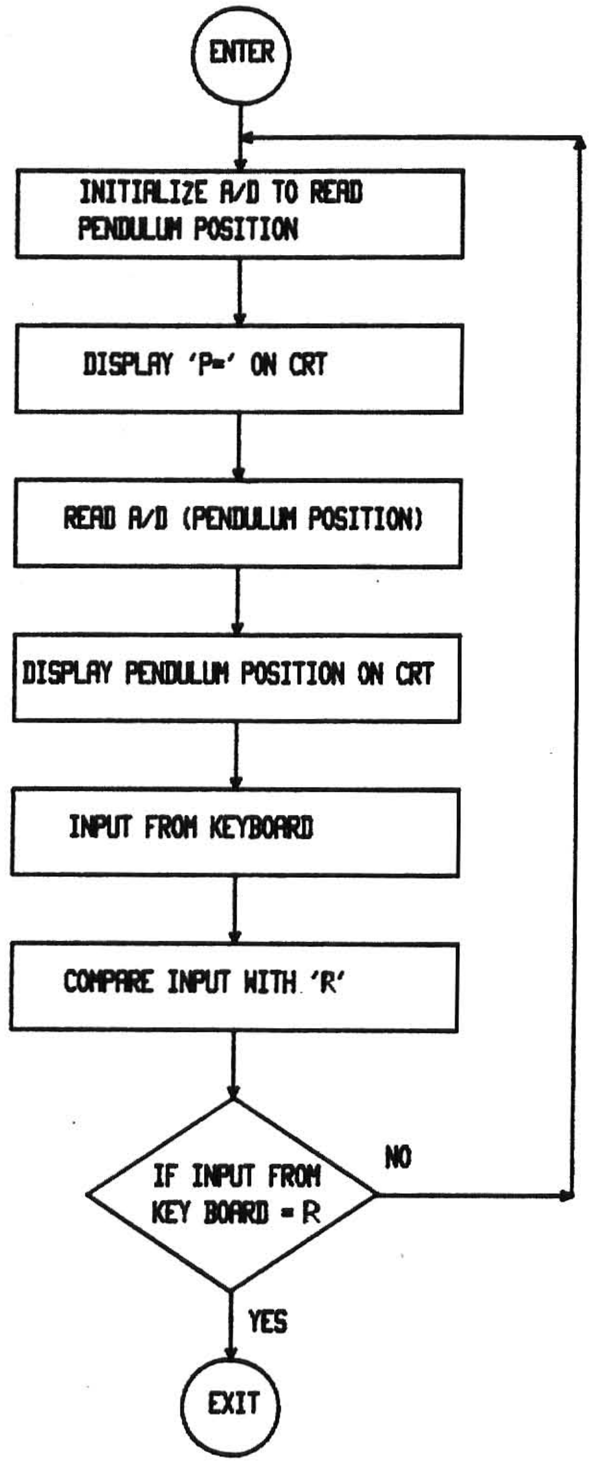

Figure 6.16 SUBROUTINE PENDIS

This subroutine displays the position of the pendulum on the CRT and hence helps in fixing the limits of the angular displacement of the pendulum about its mean position. The limits of the pendulum are fixed by adjusting the two screws at the bottom of the pendulum. After the desired adjustment is achieved, key 'R' is pressed to return back to the main program. Figure 6.16 shows the flow chart for this subroutine.

Chpaterv VII

EXPERIMENTAL PROCEDURE

7.1 Introduction

In this chapter the equipment used for experimentation will be explained first. The following section will contain a description of the testing procedure.

7.2 Equipment Arrangement

The inverted pendulum is mounted in two ball bearings on top of a light weight four wheeled cart. To measure the angular position of the pendulum, a micro torque potentiometer is attached to the pendulum shaft which provides a voltage, Vp, proportional to the angular position of the pendulum. A 10 volt power supply is used to energize the potentiometer.

The cart is driven by a d.c. servo motor, mounted on the bottom of the cart, through a speed reducing belt drive. The field of the motor is energized by a 14 volt d.c. power supply. The armature of the motor is controlled by the output of a d.c. power amplifier which is supplied by a $\pm$35 volt d.c. power supply. A permanant magnet d.c. tachometer is attached to the motor shaft which provides a voltage , Vt, proportional to the speed of the motor. Electrical signals are passed to and from the cart by a flexible cord.

The Real time control of the system is accomplished by a Digital Group, Z-80 microcomputer. The analog signals Vp and Vt are sampled and digitized by a 12 bit A/D converter. The digital control signal is delivered to the system by a 10 bit D/A converter. An 8253 programmable timer was used to time the 10 ms sampling period. Two operational amplifiers were used to provide feedback around the power amplifier and the d.c. motor. Figure 2.7 shows the arrangement of these amplifiers. Strip chart recorders were used for data collection.

7.3 Experimental Procedure

Two Z-80 microcomputers were used for this research. One was used as the real time controller described in the previous section. The other was used for program development. The program development computer operates under the DISKMON Operating System with two floppy disks for file storage. Both microcomputers are equipped with audio cassette recorder interfaces. By connecting the audio cassette output from one computer to the audio cassette input of the other, files can be transferred directly between the two microcomputers.

The assembler program is loaded into both the microcomputers from the disk and the cassette drive. The object code of the real-time control program is then loaded from the disk and is placed in memory locations starting from 3200H in the development computer. By executing the instruction 'SAVES' on the development microcomputer and the instructions '1OADS' on the real-time control microcomputer , the object code program is transfered through the audio cassette interface to the real-time control microcomputer. At this point, the program is in the real time controller and is ready to be executed. Before executing the program the inverted pendulum is made vertical to find a zero reference voltage. This is accomplished by attaching a plumb line to the top of the inverted pendulum and adjusting the two screws at the bottom of the pendulum until the pendulum becomes vertical as checked by the plumb line. After removing the plumb line the real time control program is executed. By pressing keys 'I' and 'D' on the keyboard the digital value of the position of the pendulum , displayed on the CRT, is adjusted as close as possible to 800 H, which corresponds to a zero reading of the pendulum position. After biasing the potentiometer output the key 'S' is pressed. The position of the pendulum is displayed on the CRT. The angular displacement of the pendulum in either direction from the upright position is fixed by adjusting the two screws which are at the bottom of the inverted pendulum. The computation of the control signal to stabilize the system is started by depressing key 'R'.

It is not possible to adjust the pendulum to a perfectly upright position. Hence there is a bias present in the reading of the pendulum. This bias is defined as the reading of the position of the pendulum when it is in the perfectly upright position or the actual position of the pendulum when the reading is zero.It is evident from the model of the pendulum and cart, Equation 2.12, that the pendulum can be maintained at a nonvertical position by maintaining a constant acceleration of the cart. The larger the angle, the larger the required acceleration. Since it is not possible to completely eliminate the bias in the pendulum position measurement there is a tendency for the cart to move in the direction of the bias. If a slight, upward curvature is built into each end of the surface on which the cart runs the effect of the bias is eliminated and the cart will find an equilibrium position about which it will operate.

Chapter VIII

PRESENTATION OF EXPERIMENTAL RESULTS

8.1 Introduction

Data obtained from the testing described in the previous chapter is discussed in this chapter. Data of the position of the pendulum, speed of the armature of the d.c. motor which drives the cart,distance of travel of the cart and the control signal from the D/A converter which control the speed of the d.c. motor are obtained on strip chart recorders at different chart speeds.

8.2.1 Results of Position of Pendulum versus Time

The pendulum was given an initial angular displacement of about 10^{o} from the upright position. The computation for the control signal to stabilize the pendulum was started by depressing key 'R' on the keyboard. The signal from the potentiometer which gives the angular displacement of the pendulum was recorded on the strip chart recorder for different speeds of 1 mm/sec, 5 mm/sec and 25 mm/sec. The sensitivity of the strip chart recorder was held constant at 50 mv/division. Figures 8.1.1, 8.1.2 and 8.1.3 show the results of the pendulum position versus time for the different chart speeds. The results indicate that at all times the position of pendulum is within 2.5^{o} from the upright position of the pendulum.

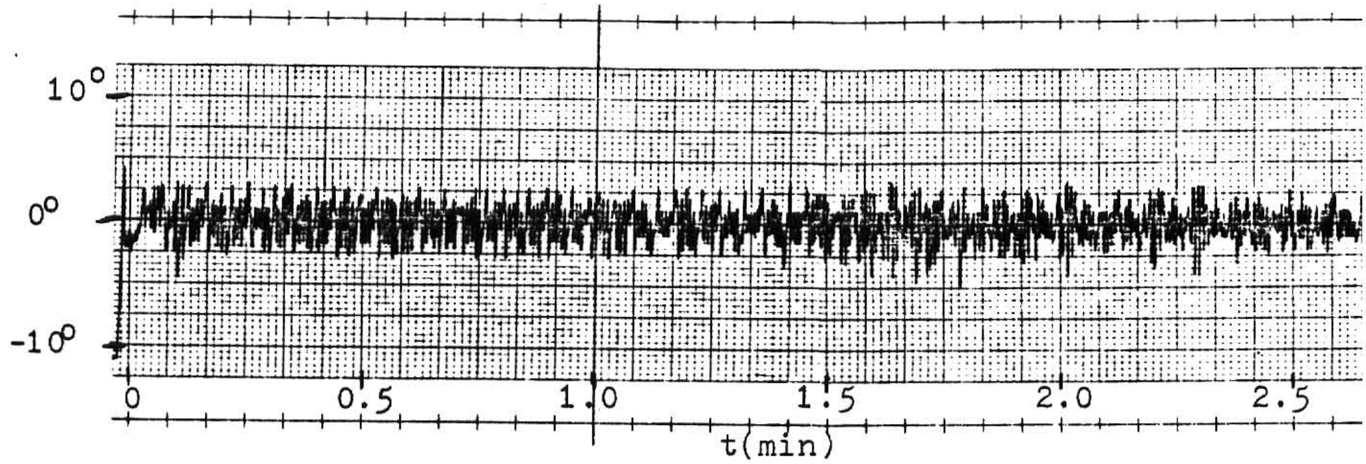

Figure 8.1.1 Position of Pendulum (degree) vs Time (minute), 1mm/sec chart speed.

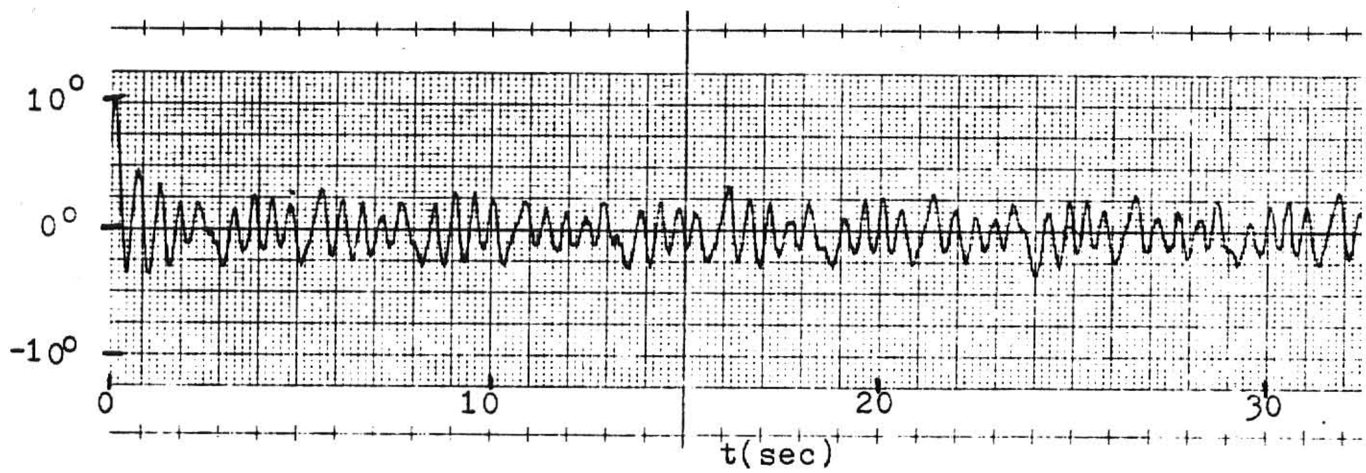

Figure 8.1.2 Position of Pendulum (degree) vs Time (second), 5 mm/sec chart speed.

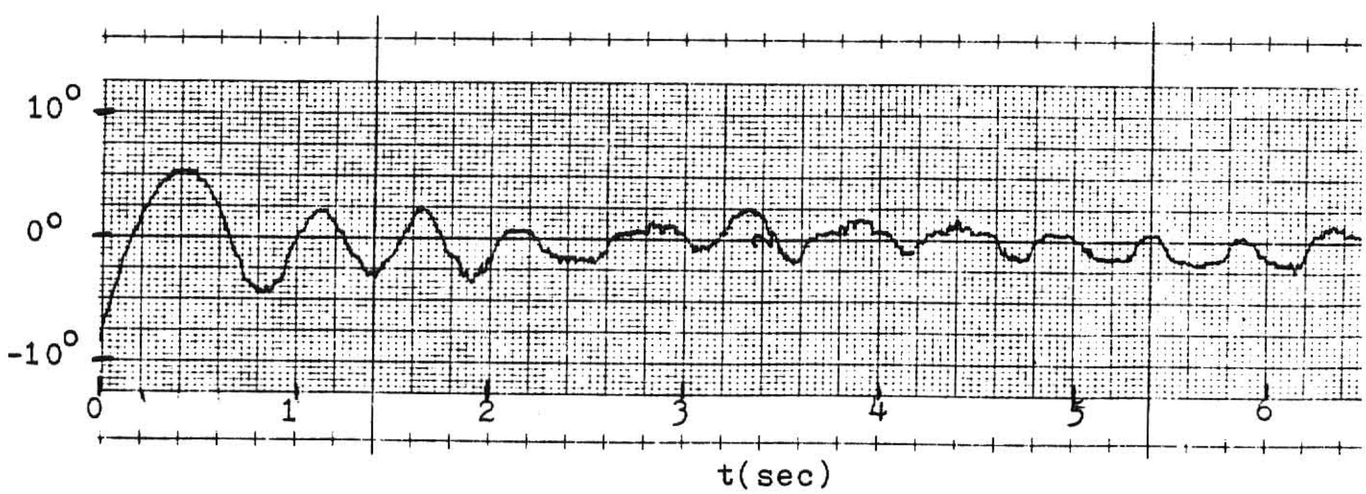

Figure 8.1.3 Position of Pendulum (degree) vs Time (second), 25 mm/sec chart speed.

8.2.2 Results of Speed of the D.C. Motor versus Time

The signal from the tachometer which is coupled to the shaft of the d.c. motor was recorded on the strip chart recorder for different chart speeds of 1 mm/sec, 5 mm/sec and 25 mm/sec. The sensitivity of the strip chart recorder was maintained at 200 mv/division. Figures 8.2.1, 8.2.2 and 8.2.3 show the speed of the d.c. motor versus time for different chart speeds. The maximum speed of the cart is 1.0 m/sec. It was observed that the output signal from the tachometer is periodic in nature with an almost constant amplitude.

8.2.3 Results of Distance Moved by Cart versus Time

The signal from the tachometer was integerated by an operational amplifier and was recorded on a strip chart recorder. The results thus obtained from the strip chart gave the distance moved by the cart from the starting position. The readings were recorded for chart speeds of 5 mm/sec and 25 mm/sec. The sensitivity of the strip chart was maintained at 500 mv/division. Figures 8.3.1 and 8.3.2 show the results of the distance moved by the cart versus time. For all times, the cart was observed to move back and forth through a distance of about 1 meter on either side of the mean position.

8.2.4 Result of Control Signal versus Time

The control signal from the D/A converter was recorded on the strip chart recorder for chart speeds of 5 mm/sec and 25 mm/sec. The sensitivity of the strip chart was maintained at 500

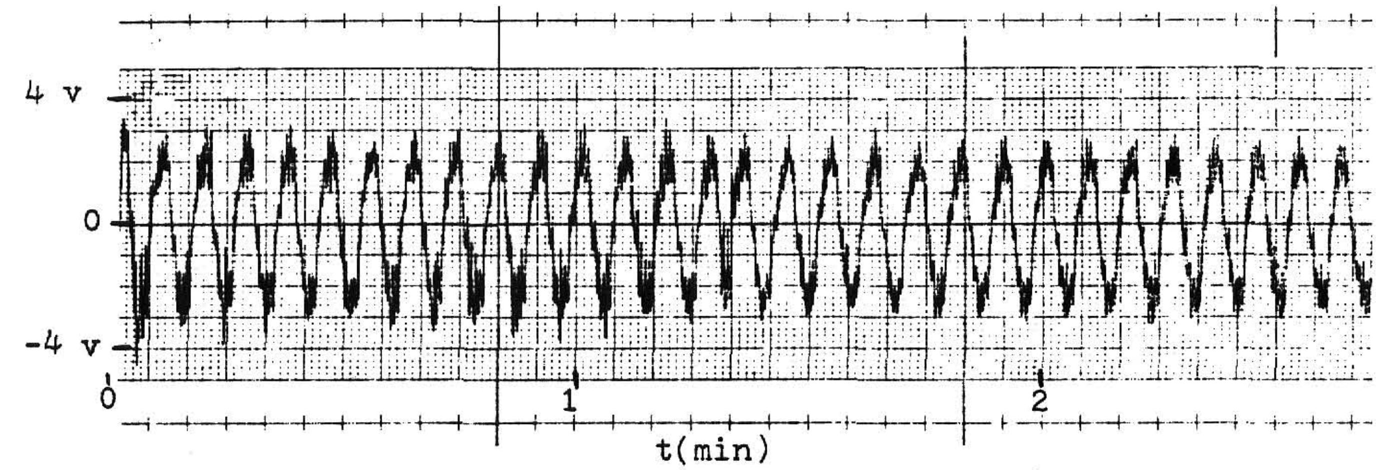

Figure 8.2.1 Speed of the D.C. Motor (200 mv/div) vs Time (mi
1 mm/sec chart speed.

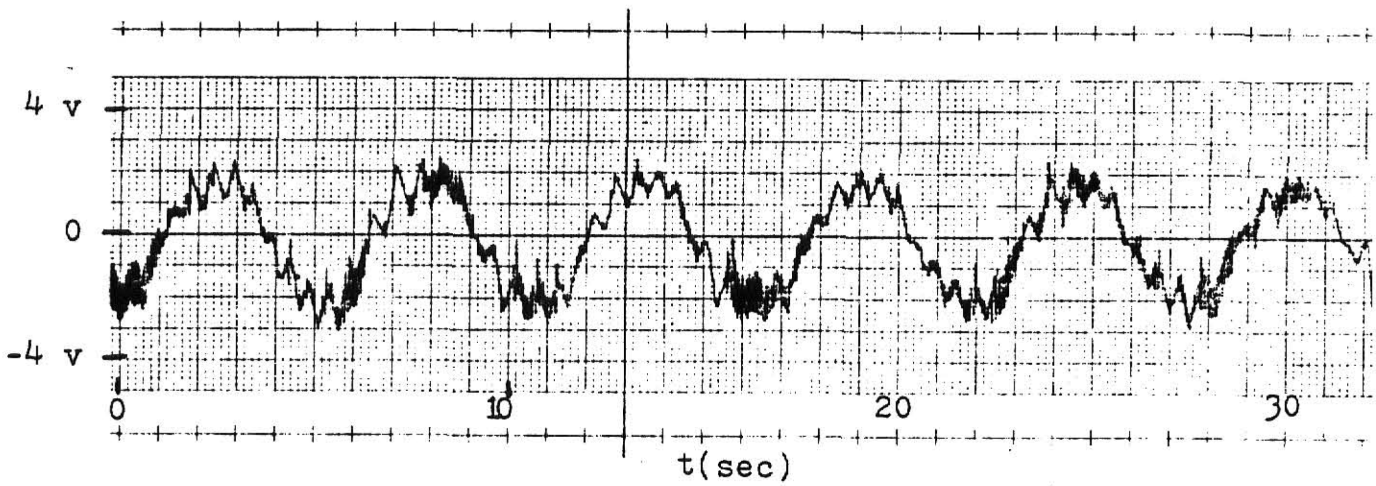

Figure 8.2.2 Speed of the D.C. Motor (200 mv/div)vs Time (sec
5 mm/sec chart speed.

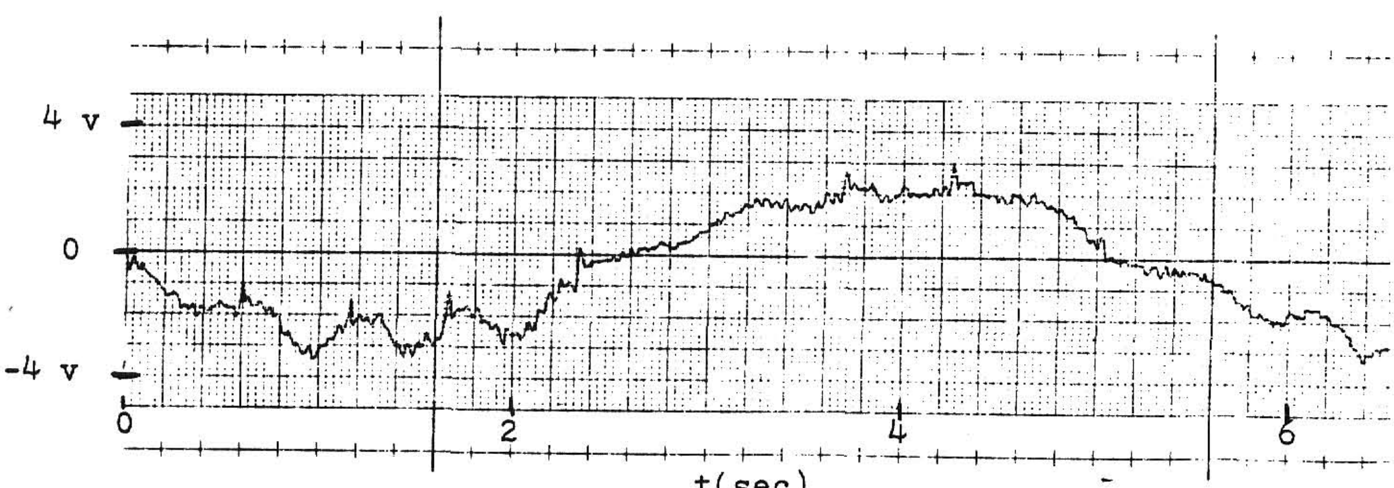

Figure 8.2.3 Speed of the D.C. Motor (200 mv/div) vs Time (se
25 mm/sec chart speed.

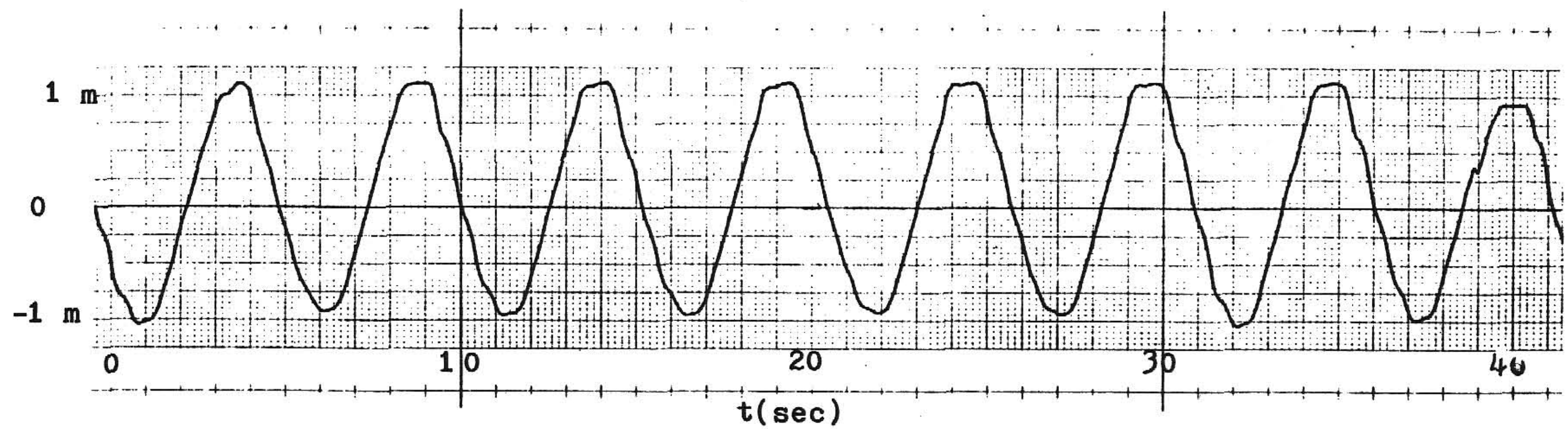

Figure 8.3.1 Distance moved by the cart (meters) vs Time (sec), 5 mm/sec chart speed.

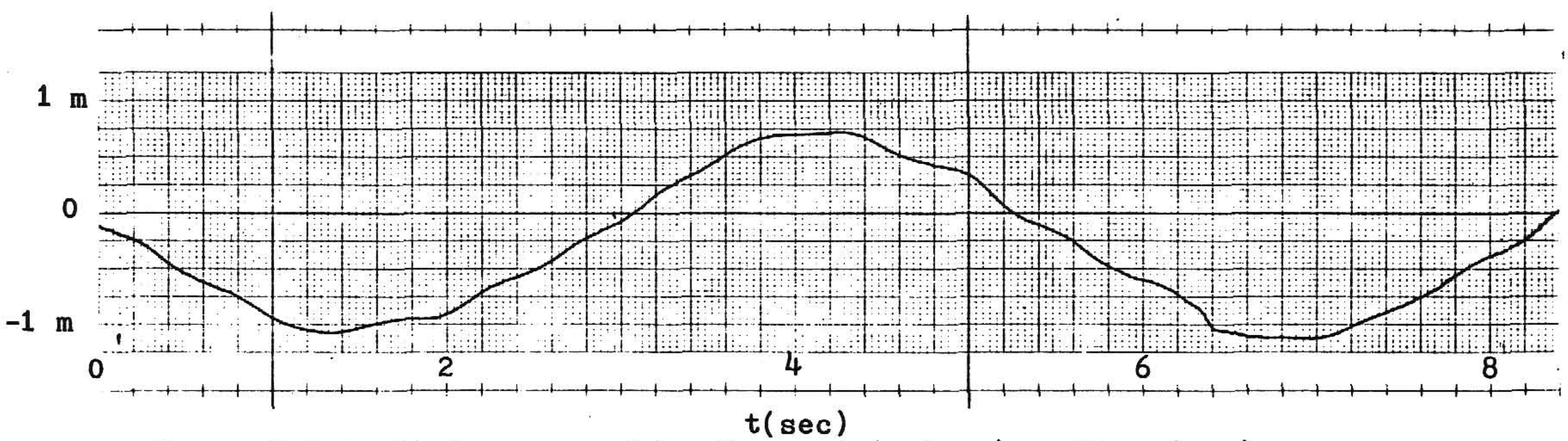

Figure 8.3.2 Distance moved by the cart (meters) vs Time (sec), 25 mm/sec chart speed.

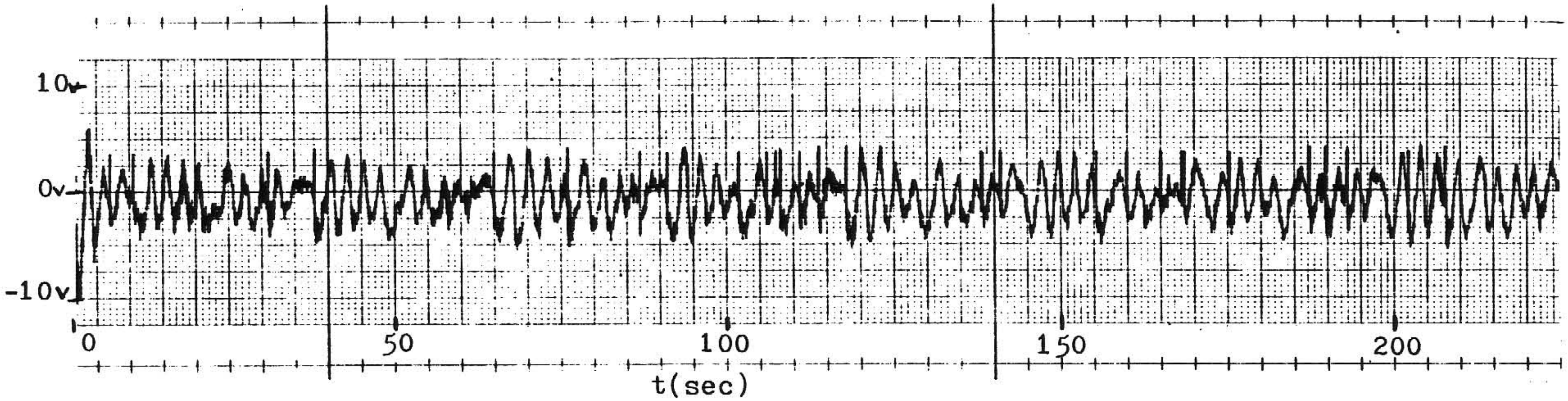

Figure 8.4.1 Control Signal from the A/D Converter vs Time, 5 mm/sec chart speed

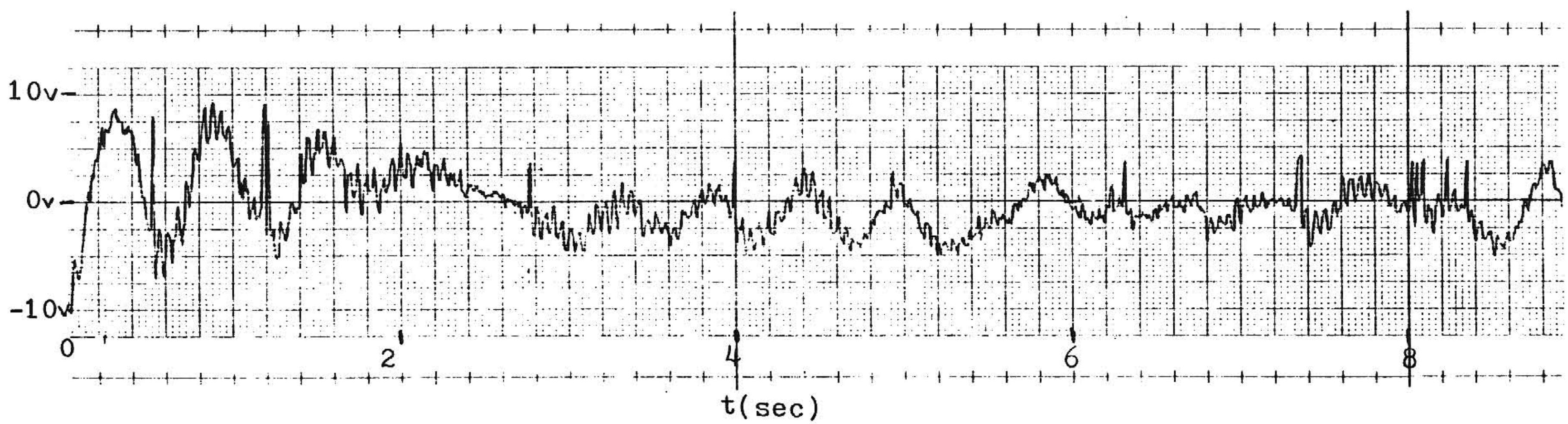

Figure 8.4.2 Control Signal from the A/D Converter vs Time,25 mm/sec chart speed

mv/division. The output control voltage from the D/A converter was limited to 10 volts. Figures 8.4.1 and 8.4.2 show the results of the control signal versus time. It may be observed that the output voltage from the D/A converter is large when the pendulum is initially at about 10^{o} angular displacement from the mean position Subsequently, when the cart moves and when the pendulum displacement becomes smaller, the output voltage also becomes corresponding smaller.

CHAPTER IX

CONCLUSIONS AND RECOMMENDATIONS

9.1 Introduction

This chapter provides a conclusion of this research and recommendations for further study.

9.2 Conclusions

From the experiment it is observed that the digital compensation is succesful in stabilizing the naturally unstable cart pendulum system. A comparison of Figures 5.1 and 5.2 (for a gain Kp= 1.0) with Figures 8.1 and 8.4 show that the response of the pendulum and the control voltage matches with the result of the computer simulation during an initial period of about 1.5 seconds. Subsequently, the effects of the bias, the non-linearities and the discretization of the sampled data makes the actual response deviate from the predicted response. It is believed that in the absence of these three effects, the actual response could match with the computer simulation for a much longer period.

9.3 Recommendations

There are several recommendations which can be made to improve upon the research conducted for this thesis, and to expand upon the system for further research.

A major time of this research was spent in writing and debugging the assembly language software control program. Hence it would be advantageous to work on a more advanced microprocessor (16 bits) or a microcomputer having a Fortran or Basic compiler.

If a higher stiffness belt is used, then the transfer function relating the speed of the cart to the speed of the d.c. motor can be modeled by a gain alone. This would simplify the stability analysis considerably.

Resolution of the accurate output control signal will provide smoother control. Hence, it is recommended that a 12 bit D/A converter be used instead of the 10 bit D/A converter used in the present research.

Two problems which were frequently encountered with the physical system were the drive belts and the potentiometer. Due to misalignment or fatigue, the life of the belts were short and hence needed replacement periodically. It would be worthwhile to modify the driving mechanism. The microtorque potentiometer, which was used to measure the position of the pendulum, failed. Therefore,a different method of measuring the angular displacement of the pendulum should be considered.

Since the mathematical model of the system is developed and adequate software is available, the technique of adaptive optimal control may be applied to the system

REFERENCES

1) Ogata, K., Modern Control Engineering, Prentice-Hall, Englewood Cliffs, N.J., 1970.

2) Rodany Kaks, Programming the Z-80, Second Edition, Sybex Inc., California, 1980

3) Tou, Julius T., Digital and Sampled Data Control Systems, McGraw Hill, New York, 1959.

4) Kuo, B.C., Discrete-Data Control Systems, Prentice-Hall, Englewood Cliffs, N.J., 1970.

5) Bryant, David, DISKMON Operating System Version 1.0, 1st Edition , The Digital Group, Denver, 1978.

6) Dorf, R. C., Modern Control Systems, Third Edition, Addison-Wesley, Reading, Massachussetts, 1980.

7) Givone, D.D., and Roesser, R. P., Microprocessor/Microcomputer, An Introduction, McGraw-Hill, New York, 1980.

APPENDICES

APPENDIX A

VALUES OF THE ROOTS OF THE INNER LOOP OF THE SYSTEM

VALUES OF THE ROOTS OF THE INNER LOOP OF THE SYSTEM

RANGE OF K: Kmin= 0

7.995000E-01	-4.102435E-01
1.000000E+00	1.091060E-11
7.995000E-01	4.102435E-01

ROOTS: K = .05

REAL	IMAGINARY
7.770426E-01	-4.141527E-01
9.949149E-01	1.010151E-11
7.770426E-01	4.141527E-01

ROOTS: K = .1

REAL	IMAGINARY
7.543124E-01	-4.164231E-01
9.903752E-01	7.109594E-12
7.543124E-01	4.164231E-01

ROOTS: K = .15

REAL	IMAGINARY
7.313561E-01	-4.170888E-01
9.862879E-01	-4.453062E-11
7.313561E-01	4.170888E-01

ROOTS: K = .2

REAL	IMAGINARY
7.082096E-01	-4.161644E-01
9.825807E-01	-1.688629E-11
7.082096E-01	4.161644E-01

ROOTS: K = .25

REAL	IMAGINARY
6.849016E-01	-4.136458E-01
9.791967E-01	8.161054E-12
6.849016E-01	4.136458E-01

ROOTS: K = .3

REAL	IMAGINARY
6.614549E-01	-4.095096E-01
9.760902E-01	1.457845E-11
6.614549E-01	4.095096E-01

ROOTS: K = .35

REAL	IMAGINARY
6.378879E-01	-4.037117E-01
9.732241E-01	2.445292E-13
6.378879E-01	4.037117E-01

ROOTS: K = .4

REAL	IMAGINARY
6.142159E-01	-3.961843E-01
9.705682E-01	-3.656455E-11
6.142159E-01	3.961843E-01

ROOTS: K = .45

REAL	IMAGINARY
5.904515E-01	-3.868312E-01
9.680970E-01	-1.739887E-11
5.904515E-01	3.868312E-01

ROOTS: K = .5

REAL	IMAGINARY
5.666052E-01	-3.755206E-01
9.657895E-01	-8.005918E-13
5.666052E-01	3.755206E-01

ROOTS: K = .55

REAL	IMAGINARY
5.426861E-01	-3.620732E-01
9.636279E-01	6.720318E-12
5.426861E-01	3.620732E-01

ROOTS: K = .6

REAL	IMAGINARY
5.187016E-01	-3.462442E-01
9.615968E-01	1.495548E-11
5.187016E-01	3.462442E-01

ROOTS: K = .65

REAL	IMAGINARY
4.946584E-01	-3.276925E-01
9.596833E-01	-4.559176E-12
4.946584E-01	3.276925E-01

ROOTS: K = .7

REAL	IMAGINARY
4.705620E-01	-3.059269E-01
9.578760E-01	-8.691951E-13
4.705620E-01	3.059269E-01

ROOTS: K = .75

REAL	IMAGINARY
4.464174E-01	-2.802033E-01
9.561651E-01	9.913813E-12
4.464174E-01	2.802033E-01

ROOTS: K = .8

REAL	IMAGINARY
9.545422E-01	4.000000E-12
4.222289E-01	-2.493033E-01
4.222289E-01	2.493033E-01

ROOTS: K = .85

REAL	IMAGINARY
9.529996E-01	-6.000000E-12
3.980002E-01	-2.109687E-01
3.980002E-01	2.109687E-01

ROOTS: K = .9

REAL	IMAGINARY
9.515308E-01	0.000000E+00
3.737346E-01	-1.599452E-01
3.737346E-01	1.599452E-01

ROOTS: K = .95

REAL	IMAGINARY
9.501298E-01	-8.000000E-13
3.494351E-01	-7.333270E-02
3.494351E-01	7.333270E-02

ROOTS: K = 1

REAL	IMAGINARY
9.487914E-01	-1.252500E-15
1.981982E-01	3.832000E-16
4.520104E-01	8.693000E-16

ROOTS: K = 1.05

REAL	IMAGINARY
9.475109E-01	-7.594890E-16
1.036031E-01	7.603000E-16
4.978860E-01	-8.110000E-19

ROOTS: K = 1.1

REAL	IMAGINARY
9.462842E-01	1.148200E-15
2.555930E-02	-1.148200E-15
5.271565E-01	3.000000E-23

ROOTS: K = 1.15

REAL	IMAGINARY
9.451074E-01	-3.894200E-15
-4.505467E-02	3.894200E-15
5.489473E-01	1.800000E-22

ROOTS: K = 1.2

REAL	IMAGINARY
9.439771E-01	-5.000000E-26
-1.112992E-01	-2.000000E-26
5.663221E-01	7.000000E-26

ROOTS: K = 1.25

REAL	IMAGINARY
9.428903E-01	-1.521805E-16
-1.746331E-01	-7.295946E-17
5.807427E-01	2.251400E-16

ROOTS: K = 1.3

REAL	IMAGINARY
9.418442E-01	-1.101857E-16
-2.358798E-01	-4.636628E-17
5.930356E-01	1.565520E-16

ROOTS: K = 1.35

REAL	IMAGINARY
9.408361E-01	2.599400E-22
-2.955545E-01	-2.600000E-22
6.037184E-01	6.000000E-26

ROOTS: K = 1.4

REAL	IMAGINARY
9.398638E-01	1.600000E-18
-3.540026E-01	-1.600000E-18
6.131388E-01	5.000000E-25

ROOTS: K = 1.45

REAL	IMAGINARY
9.389251E-01	9.171478E-16
-4.114676E-01	2.817872E-16
6.215424E-01	-1.198935E-15

ROOTS: K = 1.5

REAL	IMAGINARY
9.380182E-01	0.000000E+00
-4.681280E-01	0.000000E+00
6.291098E-01	0.000000E+00

ROOTS: K = 1.55

REAL	IMAGINARY
9.371410E-01	6.698719E-14
-5.241186E-01	-9.191919E-14
6.359775E-01	2.493200E-14

APPENDIX B

VALUES OF THE ROOTS OF THE ENTIRE SYSTEM

VALUES OF THE ROOTS OF THE ENTIRE SYSTEM

RANGE OF K: Kmin= 0

1.079552E+00	1.310619E-02
4.946450E-01	-3.276478E-01
1.079552E+00	-1.310622E-02
9.511167E-01	-3.167512E-02
9.511167E-01	3.167516E-02
4.946450E-01	3.276478E-01

ROOTS: K = .001

REAL	IMAGINARY
5.020681E-01	-3.212212E-01
1.068103E+00	5.054474E-02
1.068103E+00	-5.054476E-02
9.543613E-01	-3.314937E-02
9.543613E-01	3.314939E-02
5.020681E-01	3.212212E-01

ROOTS: K = .002

REAL	IMAGINARY
5.098415E-01	-3.143125E-01
1.055700E+00	7.126864E-02
1.055700E+00	-7.126862E-02
9.582096E-01	-3.410478E-02
9.582096E-01	3.410476E-02
5.098415E-01	3.143125E-01

ROOTS: K = .003

REAL	IMAGINARY
5.180066E-01	-3.068421E-01
1.042401E+00	8.876453E-02
1.042401E+00	-8.876457E-02
9.625626E-01	-3.412500E-02
9.625628E-01	3.412504E-02
5.180066E-01	3.068421E-01

ROOTS: K = .004

REAL	IMAGINARY
5.266118E-01	-2.987086E-01
1.028680E+00	1.055223E-01
1.028680E+00	-1.055223E-01
9.668970E-01	-3.282528E-02
9.668971E-01	3.282530E-02
5.266118E-01	2.987086E-01

ROOTS: K = .005

REAL	IMAGINARY
5.453778E-01	-2.798786E-01
1.002387E+00	1.387069E-01
1.002387E+00	-1.387069E-01
9.728614E-01	-2.758488E-02
9.728615E-01	2.758488E-02
5.453778E-01	2.798786E-01

ROOTS: K = .007

REAL	IMAGINARY
5.556750E-01	-2.687644E-01
9.897143E-01	1.548502E-01
9.897143E-01	-1.548502E-01
9.744555E-01	-2.485252E-02
9.744555E-01	2.485252E-02
5.556750E-01	2.687644E-01

ROOTS: K = .008

REAL	IMAGINARY
5.666763E-01	-2.560968E-01
9.768734E-01	1.708854E-01
9.768734E-01	-1.708854E-01
9.755139E-01	-2.236289E-02
9.755139E-01	2.236288E-02
5.666763E-01	2.560968E-01

ROOTS: K = .009

REAL	IMAGINARY
5.784281E-01	-2.413833E-01
9.636118E-01	1.872381E-01
9.636118E-01	-1.872381E-01
9.762424E-01	-2.011619E-02
9.762424E-01	2.011620E-02
5.784281E-01	2.413833E-01

ROOTS: K = .01

REAL	IMAGINARY
5.908897E-01	-2.238971E-01
9.498486E-01	2.044954E-01
9.498486E-01	-2.044953E-01
9.767627E-01	-1.807215E-02
9.767628E-01	1.807215E-02
5.908897E-01	2.238971E-01

ROOTS: K = .011

REAL	IMAGINARY
9.357967E-01	-2.234157E-01
6.037763E-01	-2.025793E-01
9.357967E-01	2.234157E-01
9.771469E-01	-1.618428E-02
9.771469E-01	1.618427E-02
6.037763E-01	2.025793E-01

ROOTS: K = .012

REAL	IMAGINARY
9.222478E-01	-2.447989E-01
6.162523E-01	-1.760290E-01
9.222478E-01	2.447990E-01
9.774385E-01	-1.440874E-02
9.774385E-01	1.440874E-02
6.162523E-01	1.760290E-01

ROOTS: K = .013

REAL	IMAGINARY
9.107764E-01	-2.688608E-01
6.267158E-01	-1.427613E-01
9.107764E-01	2.688608E-01
9.776652E-01	-1.270519E-02
9.776651E-01	1.270519E-02
6.267158E-01	1.427613E-01

ROOTS: K = .014

REAL	IMAGINARY
9.029487E-01	-2.943885E-01
6.335825E-01	-1.006692E-01
9.029487E-01	2.943885E-01
9.778449E-01	-1.103109E-02
9.778448E-01	1.103110E-02
6.335825E-01	1.006692E-01

ROOTS: K = .015

REAL	IMAGINARY
8.988857E-01	-3.193909E-01
6.367193E-01	-3.243024E-02
8.988857E-01	3.193909E-01
9.779898E-01	-9.335428E-03
9.779898E-01	9.335429E-03
6.367193E-01	3.243024E-02

ROOTS: K = .016

REAL	IMAGINARY
8.975191E-01	-3.426622E-01
8.975191E-01	3.426622E-01
5.554908E-01	4.594307E-12
9.781085E-01	-7.541159E-03
9.781085E-01	7.541159E-03
7.188814E-01	-2.508800E-17

ROOTS: K = .017

REAL	IMAGINARY
8.977534E-01	-3.639645E-01
8.977534E-01	3.639645E-01
5.210561E-01	-6.811305E-12
9.782068E-01	-5.485795E-03
9.782068E-01	5.485795E-03
7.510882E-01	2.564400E-15

ROOTS: K = .018

REAL	IMAGINARY
8.988720E-01	-3.834805E-01
8.988720E-01	3.834805E-01
4.967412E-01	-6.851178E-13
9.782893E-01	-2.501534E-03
9.782893E-01	2.501534E-03
7.714385E-01	-4.971620E-15

ROOTS: K = .019

REAL	IMAGINARY
9.004628E-01	-4.014802E-01
9.004628E-01	4.014802E-01
4.774001E-01	4.911956E-15
9.822457E-01	-5.018711E-13
9.744726E-01	5.292850E-13
7.858957E-01	-3.231030E-14

ROOTS: K = .02

REAL	IMAGINARY
9.022944E-01	-4.182112E-01
9.022944E-01	4.182112E-01
4.611410E-01	2.139395E-15
9.842804E-01	-3.399471E-13
9.725573E-01	3.746179E-13
7.968099E-01	-3.450530E-14

ROOTS: K = .021

REAL	IMAGINARY
9.042351E-01	-4.338763E-01
9.042351E-01	4.338763E-01
4.470187E-01	6.344000E-16
9.856676E-01	-2.996953E-13
9.712728E-01	3.315348E-13
8.053854E-01	-2.875770E-14

ROOTS: K = .022

REAL	IMAGINARY
9.062091E-01	-4.486374E-01
9.062091E-01	4.486374E-01
4.344820E-01	-3.939000E-15
9.867416E-01	-1.425239E-13
9.702880E-01	1.640585E-13
8.123227E-01	-2.142660E-14

ROOTS: K = .023

REAL	IMAGINARY
9.081717E-01	-4.626228E-01
9.081717E-01	4.626228E-01
4.231772E-01	-6.845635E-15
9.876197E-01	-1.064870E-14
9.694876E-01	2.921156E-14
8.180620E-01	-4.242680E-15

ROOTS: K = .024

REAL	IMAGINARY
9.100968E-01	-4.759352E-01
9.100968E-01	4.759352E-01
4.128619E-01	1.040071E-13
9.883609E-01	6.204886E-12
9.688148E-01	-6.506492E-12
8.228962E-01	1.154250E-14

APPENDIX C

THE SIMULATION PROGRAM

```
10 ! ****************************************************************
20 ! *
50 ! *                  THE SIMULATION PROGRAM
60 ! *
70 ! *****************************************************************
90    IMAGE 8(DDD.DDD,2X)
100   T=.010                   ! Sampling Interval.
110   Saturation=S=10          ! Maximum and minimum output voltage from the
                                 power supply to the cart.
120   Outresolution=.010       ! Resolution of the Output Control Signal 'U'
130   Velresolution=.030       ! Resolution of the Signal,Vt, from Tachometer.
140   Posresolution=.020       ! Resolution of the Signal,Vp,from Potentiomete
150   Bias=.030                ! Bias for the reading of the pendulum position
160   K2=256                   ! This gain is used in the microcomputer.
170   INTEGER Pi,Vi,Ci5
180   Vp(1)=Vp(2)=Vp(3)=Vp(4)=Vpi(3)=Vpi(4)=.3
190   Kp=1.45                  ! Gain of the potentiometer.
200   G1d=1.1                  ! Coefficients of Transfer Funtion H1(Z)
210   G1n=.85
220   K1=.75
230   H2d=.85
240   M=Z=C0(3)=C0(4)=C0(5)=C1(3)=C1(4)=C2(3)=C2(4)=U(1)=U(2)=0
250 ! U(3)=U(4)=U(5)=Vt(3)=Vt(4)=Vt(5)=Us(1)=Us(2)=Us(3)=Us(4)=
      =Vi(2)=Vi(3)=Vi(4)=0
260   PLOTTER IS 13,"GRAPHICS"
270   GRAPHICS
280 ! PLOTTER IS 7,5,"9872A"
290   PRINTER IS 7,5
300 ! PRINT "VS5"
310   LOCATE 0,123,0,100
320   CSIZE 3.1,.45
330   SCALE -.5,5,-.32,.32
340   AXES 1,.1,0,0
350   FRAME
360   PRINTER IS 16
370   FOR I=1 TO 5
380   MOVE I-.12,-.04
390   LABEL I
400   NEXT I
410   FOR I=-.3 TO .3 STEP .1
420   MOVE -.3,I
430   LABEL I
440   NEXT I
450 ! ****************************************************************
460 ! Statement 510 to 750 are developed in Chapter 3.  The coefficient
470 ! of the transfer function are calculated in this part of the program
480 ! The nomenclature used for calculating the coefficients are similar
490 ! to that used in chapter 3.
500 ! *****************************************************************
510   A=10.692
520   B=47.409293
530 N11=1.96733491*EXP(-A*T)*COS(B*T)+.2017093*EXP(-A*T)*SIN(B*T)+.032650
540 N10=.9673491*EXP(-2*A*T)+EXP(-A*T)*(.0326509*COS(B*T)+.2017093*SIN(B*
550   D11=2*EXP(-A*T)*COS(B*T)
560   D10=EXP(-2*A*T)
570   A1=4.009816
580   A2=-4.385888
```

```
590    A3=.376
600    A4=.7998549
610  ! N2=(A1+A3)*EXP(-5*T)+(A2+A3)*EXP(5*T)+(A1+A2)*2*EXP(-A*T)*COS(B*T)+
       +EXP(-A*T)*(A3*COS(B*T)+A4*SIN(B*T))
620  ! N1=2*(EXP(-5*T)*A1+EXP(5*T)*A2)*EXP(-A*T)*COS(B*T)+(A1+A2)*
     *EXP(-2*A*T)+A3+(EXP(5*T)+EXP(-5*T))*(A3*COS(B*T)+A4*SIN(B*T))*EXP(-A
630  ! N0=(EXP(-5*T)*A1+EXP(5*T)*A2)*EXP(-2*A*T)+EXP(-A*T)*(A3*COS(B*T)+
       +A4*SIN(B*T))
640    D33=EXP(-5*T)+EXP(5*T)+2*EXP(-A*T)*COS(B*T)
650    D32=1+(EXP(5*T)+EXP(-5*T))*2*EXP(-A*T)*COS(B*T)+EXP(-2*A*T)
660    D31=EXP(-2*A*T)*(EXP(-5*T)+EXP(5*T))+2*EXP(-A*T)*COS(B*T)
670    D30=EXP(-2*A*T)
680    N33=N2
690    N32=N2+N1
700    N31=N1+N0
710    N30=N0
720    N21=1+EXP(-A*T)*(COS(B*T)-.2255254*SIN(B*T))-2*EXP(-A*T)*COS(B*T)
730    N20=EXP(-2*A*T)-EXP(-A*T)*(COS(B*T)-.2255254*SIN(B*T))
740    D21=D11
750    D20=D10
760  ! Statement 790 to 930 prints the transfer function of the system and
770  ! the various constant of the transfer functions.
780    PRINT "Vt(z)/U(z)=.92(Z^2-N11*Z+N10)/(Z^2-D11*Z+D10)"
790    PRINT "N11=",N11,"N10=",N10,"D11=",D11,"D10=",D10
800    PRINT "Wc(z)/Vt(z)=.0655009*(N21*Z+N20)/(Z^2-D21*Z+D20)"
810    PRINT "N21=",N21,"N20=",N20,"D21=",D21,"D20=",D20
820  ! PRINT "Vp(z)/U(z)=383.85*Kp*10^(-5)*(N33*Z^3-N32*Z^2+N31*Z-N30)/
       /(Z^4-D33*Z^3+D32*Z^2-D31*Z+D30)"
830  ! PRINT "N33=",-N33,"N32=",-N32,"N31=",-N31,"N30=",-N30,"D33=",D33,
       "D32=",D32,"D31=",D31,"D30=",D30
840    MOVE 1.7,.3
850   ! Statement 1040 TO 1220 prints the various practical limitation
860   ! and the compensating transfer function used on the plot.
870    LABEL "SAMPLING TIME     =";T
880    LABEL "Saturation volts =";Saturation
890    LABEL "Outresolution     =";Outresolution
900    LABEL "Velresolution     =";Velresolution
910    LABEL "Posresolution     =";Posresolution
920    LABEL "Bias in Pot volts=";Bias
930    MOVE 1,-.12
940    LABEL " G1(z)=C1(z)/Vpi(z))=K2*(Z-G1n)/(Z-G1d)"
950    LABEL " H2(Z)=C2(z)/Vti(z)=K1*(Z^2-2*H2d*Z+H2d^2)/(Z^2-C*Z+C1)"
960    LABEL ""
970    LABEL "K2=";K2,"Kp=";Kp
980    LABEL "G1n=";G1n
990    LABEL "G1d=";G1d
1000   LABEL ""
1010   LABEL "K1=";K1
1020   LABEL "H2d=";H2d
1030 ! ***********************************************************
1040 ! The equations used from lines 1070 to 1220 are developed in
1050 ! Chapter 5.
1060 ! ***********************************************************
1070   FOR X=0 TO 300000
1080   FOR Y=0 TO 10
1090   M=M+T
1100   C1(4)=G1d*C1(3)-K2*(Vpi(4)-G1n*Vpi(3))
```

```
1110 ! C2(4)=2.69*C2(3)-2.58*C2(2)+.89*C2(1)+K1*(Vti(4)-H2d*2*Vti(3)+
       H2d^2*Vti(2))
1120  U(4)=C1(4)-C2(4)
1130  Us(4)=U(4)
1140 ! Statements 1160 and 1170 simulates the saturation of the output
1150 ! voltage from the D/A converter.
1160  IF Us(4)>S THEN Us(4)=S
1170  IF Us(4)<-S THEN Us(4)=-S
1180 ! Statement 1220 to 1230 simulates the resolution of the D/A Convert
1190  Ui=Us(4)/Outresolution
1200  Us(4)=Ui*Outresolution
1210  Vt(5)=D11*Vt(4)-D10*Vt(3)+.92*(Us(4)-N11*Us(3)+N10*Us(2))
1220   P(5)=D33*P(4)-D32*P(3)+D31*P(2)-D30*P(1)-383.5*Kp*10^(-5)*
      *(N33*Cs5(4)-N32*Cs5(3)+N31*Cs5(2)-N30*Cs5(1))
1230 ! Statements 1270 to 1310 simulates the resolution of the A/D
1240 ! converters for the reading of the velocity of the cart and
1250 ! pendulum position.  The bias in the reading of the pendulum
1260 ! position is also included.
1270  Vpi=Vp(5)/Posresolution
1280  Vpi(5)=Vpi*Posresolution
1290  Vpi(5)=Vpi(5)+Bias
1300  Vti=Vt(5)/Velresolution
1310  Vti(5)=Vti*Velresolution
1320 ! Statements 1340 to 1550 shifts the values of the various varaibles
1330 ! from K+1 instant to K instant.
1340  Vpi(3)=Vpi(4)
1350  Vpi(4)=Vpi(5)
1360  Vti(2)=Vti(3)
1370  Vti(3)=Vti(4)
1380  Vti(4)=Vti(5)
1390  C1(3)=C1(4)
1400  Vp(1)=Vp(2)
1410  Vp(2)=Vp(3)
1420  Vp(3)=Vp(4)
1430  Vp(4)=Vp(5)
1440  Vt(2)=Vt(3)
1450  Vt(3)=Vt(4)
1460  Vt(4)=Vt(5)
1470  C2(1)=C2(2)
1480  C2(2)=C2(3)
1490  C2(3)=C2(4)
1500  U(1)=U(2)
1510  U(2)=U(3)
1520  U(3)=U(4)
1530  Us(1)=Us(2)
1540  Us(2)=Us(3)
1550  Us(3)=Us(4)
1570  PLOT M,Vp(5)
1580  NEXT Y
1590  PRINT USING 90;M,Vp(5),Vt(5),D,C1(4),C2(4),U(4),Us(4)
1600  NEXT X
1610  END
```

APPENDIX D

MEMORY MAP

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3E00	Coeff. of C1(3)	Coeff. of Vp(3)	Coeff. of C2(3)	Coeff. of C2(2)	Coeff. of C2(1)	Coeff. of Vt(3)	Coeff. of Vt(2)	Initial Value To Bias the Pendulum Position Reading				Amp. and Channel To read Vp(4).				Amp. and Channel To read Vt(4)
3E10	Sign of C1(3)	Sign of Vp(3)	Sign of C2(3)	Sign of C2(2)	Sign of C2(1)	Sign of Vt(3)	Sign of Vt(2)	Exponent of C1(3)	Exponent of C2(3)	Exponent of C2(2)	Exponent of C2(1)	Stored number of Sampling-Intervals Passed				
3E20	Magnitude of C1(3)		Magnitude of Vp(3)		Magnitude of C2(3)		Magnitude of C2(2)		Magnitude of C2(1)		Magnitude of Vt(3)		Magnitude of Vt(2)			
3E30																
3E40	Display & Store Alternate Values	Output Value to D/A is stored		Number of Sampling Intervals to be Executed												INIT. FOR Subroutine 'DISPLAY'
3E50	Init. & store Address Location of C2,C1, Vp,Vt and U	Starting Storage Location for Vt(4)		Starting Storage Location for Vp(4)		Starting Storage Location for C1(4)		Starting Storage Location for C2(4)		Starting Storage Location for U(4)						
3E60	Value of U(4) in 24 Bits, and in 2's complement															
3E70	Incomplete value of C1(4) in 24 bits and in 2's comp.			24 bit value of C2(4) in 2's complement			24 bit value of C1(4) in 2's complement					Bias value for Pendulum Position		Bias value for Speed of Armature		Stores 'A'-Reg. when input from Key

Memory Map of Page 3E

LIST OF PROGRAMS AND SUBROUTINE

Address	
3000 - 33AD	MAIN PROGRAM
3400 - 341B	MULT
341F - 342D	STAD
3431 - 3440	COMPLEMENT
3444 - 345E	EXPONENT
3462 - 34A4	TRANS
34A8 - 34DE	SHIFT
34DF - 34F7	STOP
3530 - 363D	DISPLAY
3640 - 36E6	STORE
3700 - 37CB	INITI
37CC - 37E4	CHECK
37E5 - 37EF	DEC
37F0 - 37FA	INC
3814 - 384A	SCREEN
3850 - 38D3	PENDIS

APPENDIX E

MICROCOMPUTER PROGRAM LISTING

```
*                          MAIN PROGRAM

***********************************************************
* THIS IS THE MAIN PROGRAM TO STABILIZE AN INVERTED PEN-
* DULUM MOUNTED ON A CART BY DIGITAL CONTROL.

***********************************************************

MULT    EQU  3400H
STAD    EQU  341CH
COMP    EQU  342EH
EXP     EQU  3441H
LAST    EQU  3700H
PENDIS  EQU  3850H
        ST   3001H        # OPCODE STORED FROM 3001H
        LD   A,7FH        # OUTPUT CONTROL VOLTAGE=0 VOLTS
        LD   (0FFF2H),A
        LD   (0FFF1H),A
        LD   A,0          # MEMORY LOACATION 3E00H TO 3E80H
        LD   B,80H        # IS INITIALISED TO 0
        LD   HL,3E00H
LOOP    LD   (HL),A
        INC  HL
        DJNZ LOOP
        LD   A,12H        # CHANNEL 1 AND AMPLIFICATION 4
        LD   (3E0BH),A    # TO READ POSITION OF PENDULUM
        LD   A,0          # CHANNEL 0 AND AMPLIFICATION 1
        LD   (3E0FH),A    # TO READ VELOCITY OF CART
        CALL INITI        # INITIALIZE PENDULUM POSITION
        CALL PENDIS       # TO DISPLAY PENDULUM POSITION
        LD   A,0FFH       # INITIALISE COUNTER
        OUT  10H
        CALL 00034E       # CLEAR SCREEN
        LD   A,8DH        # STORE COEFF OF C1(3)
        LD   (3E00H),A
        LD   A,0D9H       # STORE COEFF OF P(3)
        LD   (3E01H),A
        LD   A,0ACH       # STORE COEFF OF C2(3)
        LD   (3E02H),A
        LD   A,0A5H       # STORE COEFF OF C2(2)
        LD   (3E03H),A
        LD   A,39H        # STORE COEFF OF C2(1)
        LD   (3E04H),A
        LD   A,36H        # STORES COEFF OF V(3)
        LD   (3E05H),A
        LD   A,17H        # STORES COEFF OF V(2)
        LD   (3E06H),A
        LD   A,01H        # INITIALISE FOR SUBROUTINE STORE
        LD   (3E50H)      # AND DISPLAY
        LD   (3E40H),A
        LD   HL,0FFFH     # SAMPLING EVENTS TO TAKE PLACE
        LD   (3E43H),HL
        LD   A,03H        # INITILISATION FOR SUB DISPLA
        LD   (3E4FH),A
        LD   A,30H        # TIMER COUNTER IS INITI IN MODE 0
```

```
0560 *          ... CONT OF MAIN PROGRAM
0570
0580        OUT  17H
0590        LD   A, 0
0600        OUT  14H
0610        LD   A, 0
0620        OUT  14H
0630 MAIN   LD   BC, 0007H
0640        LD   DE, 3E30H
0650        LD   HL, 3E00H
0660        LDIR
0670        LD   A, 0          # C' H' L'  REGISTER IS INITIALISED
0680        LD   C, A          # TO 0
0690        LD   H, A
0700        LD   L, A
0710        EXX
0720        LD   DE, (3E24H) # LOADS MULTIPLICAND OF C2(3)
0730        LD   IX, 3E32H   # LOADS ADDRESS OF MULTIPLIER
0740        CALL MULT        # MULTIPLY C2(3) AND COEFF OF C2(3)
0750        LD   A, (3E18H)  # LOADS EXPONENT OF C2(3)
0760        LD   B, A
0770        INC  B
0780        CALL EXP
0790        LD   A, (3E12H)  # LOADS SIGN OF C2(3)
0800        CALL STAD
0810        LD   DE, (3E26H) # LOADS MULTIPLICAND OF C2(2)
0820        INC  IX
0830        CALL MULT        # MULTIPLY C2(2) AND COEFF OF C2(2)
0840        LD   A, (3E19H)  # LOADS EXPONENT OF C2(2)
0850        LD   B, A
0860        INC  B
0870        CALL EXP
0880        LD   A, (3E13H)  # LOAD SIGN OF C2(2)
0890        INC  A
0900        CALL STAD
0910        LD   DE, (3E28H) # LOAD MULTIPLICAND OF C2(1)
0920        INC  IX          # ADDRESS OF MULTIPLIER
0930        CALL MULT        # MULTIPLY C2(1) AND COEFF OF C2(1)
0940        LD   A, (3E1AH)  # LOAD EXPONENT OF C2(1)
0950        LD   B, A
0960        INC  B
0970        CALL EXP
0980        LD   A, (3E14H)  # LOAD SIGN OF C2(1)
0990        CALL STAD
1000        LD   DE, (3E2AH) # LOADS MULTIPLICAND OF V(3)
1010        INC  IX          # ADDRESS OF V(3)
1020        CALL MULT
1030        LD   A, (3E15H)  # LOADS SIGN OF V(2)
1040        INC  A
1050        CALL STAD
1060        LD   DE, (3E2CH) # LOADS MULTIPLICAND OF V(2)
1070        INC  IX          # ADDRESS OF MULTIPLIER IN IX REG
1080        CALL MULT
1090        LD   A, (3E16H)  # LOADS SIGN OF V(2)
```

```
     *           ... CONTD OF MAIN PROGRAM


     STAD    EQU   341CH
     SHIFT   EQU   34A5H
             ST    3100H       # OPCODE STORED FROM 3100H
             CALL  STAD
             EXX
             LD    B,05H       # LOAD COUNTER BY 5
             LD    A,C         # CHECK IF SIGN +VE OR -VE
             AND   80H
             JR    NZ,LOOPG    # IF -VE THEN JUMP TP LOOPG
     LOOPI   SRL   C           # DIVIDE CHL REGISTER BY 32
             RR    H
             RR    L
             DJNZ  LOOPI
             JR    LOOPJ
     LOOPG   NOP
     LOOPK   SRL   C           # DIVIDE CHL REGISTER BY 32
             RR    H
             RR    L
             DJNZ  LOOPK
             LD    A,C         # OR 0FCH BECAUSE CHL REGISTER IS
             OR    0FCH        # -VE AND IS IN 2'COMPLEMENT
             LD    C,A
     LOOPJ   LD    (3E73H),HL
             LD    A,C
             LD    (3E75H),A
             CALL  SHIFT
     LOOPE   IN    A,10H       # WAIT IF 10 MILLI SECS (SAMPLING
             AND   A           # TIME OVER)
             JR    Z,LOOPE
             LD    A,30H       # INITIALISE TIMER COUNTER IN MODE 0
             OUT   17H
             LD    A,50H       # LOADS MOST SIGNIFICANT BYTE OF
             OUT   14H         # TIMER COUNTER AND THEN THE LEAST
             LD    A,60H       # SIGNIFICANT BYTE TO MAKE SAMPLING
             OUT   14H         # TIME 10 MILLI SECS.
             LD    A,(3E0FH)   # INITIALISE A/D TO READ VELOCITY
             LD    (0FFF3H),A
     LOPF    LD    A,(0FFF1H)  # THIS AND THE NEXT LOOP CHECKS
             AND   80H         # WHETHER THE A/D CONVERSION IS
             JR    Z,LOPF      # OVER
     LOOPF   LD    A,(0FFF1H)
             AND   80H
             JR    NZ,LOOPF
             LD    A,(0FFF2H)  # CHECK SIGN
             AND   08H
             JR    Z,LAMP      # IF SIGN -VE THEN JUMP TO LAMP
             LD    A,(0FFF0H)  # V(4) IS MULTIPLIED BY -1.
             LD    L,A         # CONVERTS SHIFTED BINARY INTO
             LD    A,(0FFF2H)  # 2'COMPLEMENT FORMAT.
```

```
1620 *          ... CONTD OF MAIN PROGRAM.
1630
1640         AND  07H
1650         LD   H,A
1660         LD   A,(3E0BH)
1670         LD   (0FFF3H),A
1680         LD   DE,(3E7DH)
1690         SBC  HL,DE        # SUBTRACT BIAS
1700         LD   A,H
1710         AND  80H
1720         JR   Z,B1
1730         ADD  HL,DE
1740         LD   A,E
1750         SUB  L
1760         LD   L,A
1770         JR   B2
1780 B1      LD   (3E2AH),HL # STORE MAGNITUDE OF V(4) IN V(3)
1790         LD   A,0        # STORE +VE SIGN FOR V(3)
1800         LD   (3E15H),A
1810         LD   DE,(3E73H)
1820         LD   A,(3E75H)
1830         ADD  HL,DE
1840         ADC  0
1850         LD   (3E73H),HL
1860         LD   (3E75H),A
1870         JR   LOOPP
1880 LAMP    LD   A,(0FFF0H) # IV V(4) IS -VE IT IS CONVERTED IN
1890         CPL               # IN 2' COMPLEMENT
1900         LD   L,A
1910         LD   A,(0FFF2H)
1920         CPL
1930         AND  07H
1940         LD   H,A
1950         LD   A,(3E0BH)
1960         LD   (0FFF3H),A
1970         LD   DE,(3E7DH) # BIAS IS ADDED TO V(4)
1980         ADD  HL,DE
1990 B2      LD   (3E2AH),HL # STORE THE MAGNITUDE OF V(4) IN
2000         LD   A,01H      # V(3).  STORE -VE SIGN FOR V(4)
2010         LD   (3E15H),A  # IN V(3)
2020         LD   A,L
2030         CPL
2040         LD   L,A
2050         LD   A,H
2060         CPL
2070         LD   H,A
2080         LD   DE,(3E73H)
2090         LD   A,(3E75H)
2100         ADD  HL,DE
2110         ADC  0FFH
2120         LD   (3E73H),HL # STORE C2(4)
2130         LD   (3E75H),A
2140 LOOPP   NOP
```

```
*              .... CONTD OF MAIN PROGRAM.


MULT    EQU  3400H
COMP    EQU  342EH
STAD    EQU  341CH
DISPLA  EQU  3530H
STORE   EQU  3640H
STOP    EQU  34DFH
        ST   31E0H
        LD   DE,(3E20H)  # LOAD MULTIPLICAND OF C1(3)
        LD   IX,3E30H    # LOAD ADDRESS OF C1(3) IN IX REG
        CALL MULT        #MULTIPLY C1(3) AND COEFF OF C1(3)
        LD   B,0F7H
        LD   A,(3E17H)
        ADD  B
        CPL
        LD   B,A         # LOAD COUNTER WITH EXPONENT
LOOP0   SRL  C           # DIVIDE THE RESULT OF C1(3)
        RR   H           # BY THE REMAINDER.
        RR   L
        DJNZ LOOP0
        LD   A,(3E10H)   # TEST WHETHER C1(3) IS +VE OR -VE
        RRA              # IF -VE THEN TAKE 2'COMPLEMENT
        JR   NC,LOOPB    # IF NOT -VE THEN JUMP TO LOOPB
        CALL COMP
LOOPB   EXX              # RESULT IS STORED IN C'H'L' REG
        LD   DE,(3E22H)  # LOADS MULTIPLICAND OF P(3)
        INC  IX          # ADDRESS OF P(3) IN IX REGISTER
        CALL MULT        # MULTUPLY P(3) AND COEFF OF P(3)
        LD   B,03H       # DIVIDE THE RESULT OF P(3) WHICH
LOOPM   SRL  C           # IS STORED IN CHL REGISTER BY 8
        RR   H
        RR   L
        DJNZ LOOPM
        LD   A,(3E11H)
        CALL STAD
        EXX
LOOPN   LD   (3E70H),HL
        LD   A,C         # MEMORY LOCATION 3E72H, 3E71H AND
        LD   (3E72H),A   # 3E70H
        LD   HL,(3E41H)  # INITILISATION FOR SUB DISPLAY
        LD   A,0A0H      # INITIALISE COUNTER FOR DELAY
        LD   A,(3E40H)   # THIS PART OF THE PROGRAM STORES
        INC  A           # AND DISPLAYS ALTERNATE VALUES OF
        AND  01H         # DATA ON SCREEN AND IN MEMORY
        LD   (3E40H),A
        JR   Z,SKIP
        CALL DISPLA
        CALL STORE
        LD   HL,(3E1BH)
        INC  HL
        LD   (3E1BH),HL
```

```
*             .... CONTD OF MAIN PROGRAM.

              LD    B,75H          # DELAY TIME SO THAT THE A/D HAS
      SKIP    CALL  STOP           # ENOUGH TIME FOR SETTLING TO
              DJNZ  SKIP           # READ CHANNEL FOR PENDULUM POSI-
              LD    A,(3E0BH)      # TION.  INITIALIZE A/D TO READ
              LD    (0FFF3H),A     # PENDULUM POSITION.
      LOOPQ   LD    A,(0FFF1H)     # TEST THE MSB TO CHECK IF A/D
              AND   80H            # CONVERSION IS OVER.
              JR    Z,LOOPQ
      LOPQ    LD    A,(0FFF1H)     #
              AND   80H
              JR    NZ,LOPQ
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP
              ST    3270H
      TRANS   EQU   345FH
      MAIN    EQU   306DH
              LD    A,(0FFF2H)     # TEST IF VALUE OF P(4) IS +VE OR
              AND   08H            # -VE
              JR    Z,LOOPR        # IF P(3) -VE THEN JUMP TO LOOPR
              LD    A,(0FFF0H)     # LOADS THE LEAST SIGNIFICANT BYTE
              LD    L,A            # OF P(3) IN L REG. LOAD THE MOST
              LD    A,(0FFF2H)     # SIGNIFICANT NIBBLE IN H REGISTER
              AND   07H            # RESET THE UPPER NIBBLE
              LD    H,A
              LD    A,(3E0FH)      # INITIALISE THE A/D TO READ
              LD    (0FFF3H),A     # VELOCITY OF CART
              LD    DE,(3E7BH)     # LOAD BIAS FOR PENDULUM POSITION
              SBC   HL,DE          # SUBTRACT BIAS FROM VALUE OF P(3)
              LD    A,H            # CHECK IF MAGNITUDE OF BIAS IS
              AND   80H            # GREATER THAN THE VALUE OF P(3)
              JR    Z,B3           # IF BIAS LESS THAN P(3) THEN B3
              ADD   HL,DE          # IF BIAS > THAN P(4) THEN
              LD    A,E            # SUBTRACT P(3) FROM BIAS AND S
              SUB   L              # STORE -VE SIGN FOR P(4)
              LD    L,A
              JP    B4
      B3      LD    (3E22H),HL     # STORE MAGNITUDE OF P(4)
              LD    A,0            # STORE +VE (00H) FOR SIGN OF P(4)
              LD    (3E11H),A
              SLA   L              # MULTIPLY P(4) BY 8
              RL    H
              SLA   L
              RL    H
              SLA   L
              RL    H
              LD    A,L
              CPL
              LD    L,A
```

```
2555 *           .... CONTD OF MAIN PROGRAM.
2560        LD   DE,(3E70H)
3220        LD   A,H
3230        CPL
3240        LD   H,A
3250        LD   C,0FFH
3260        LD   DE,(3E70H)
3270        LD   A,(3E72H)
3280        ADD  HL,DE
3290        ADC  C
3300        LD   C,A
3310        LD   (3E70H),HL   # STORE THE VALUE OF C1(3) IN
3320        LD   (3E72H),A    # MEMORY LOCATION 3E72H, 3E71H AND
3330        LD   (3E76H),HL   # 3E70H ALSO STORE IT IN 3E78H,
3340        LD   (3E78H),A    # 3E77H AND 3E76H
3350        JR   LOOPS
3360 LOOPR  LD   A,(0FFF0H)   # LOADS P(4) MOST SIGNIFICANT BYTE
3370        CPL               # SINCE A/D IS IN SHIFTED BINARY
3380        LD   L,A          # AND VALUE IS -VE COMPLEMENT V
3390        LD   A,(0FFF2H)   # THE VALUE TO GET CORRECT MAGNIT-
3400        CPL               # TUDE.  DO THE SAME FOR UPPER
3410        AND  07H          # NIBBLE.  RESET THE UPPER BYTE
3420        LD   H,A
3430        LD   A,(3E0FH)    # INITIALISE A/D TO READ VELOCITY
3440        LD   (0FFF3H),A   # OF CART
3450        LD   DE,(3E7BH)   # LOAD BIAS OF PENDULUM POSITION
3460        ADD  HL,DE        # SINCE P(4) IS -VE ADD BIAS T
3470 B4     LD   (3E22H),HL   # STORE MAGNITUDE OF P(4) IN P(3)
3480        LD   A,01H        # SINCE SINE -VE STORE (01H) FOR
3490        LD   (3E11H),A    # SIGN OF P(4)
3500        SLA  L            # MULTIPLY P(4) BY 8
3510        RL   H
3520        SLA  L
3530        RL   H
3540        SLA  L
3550        RL   H
3560        LD   DE,(3E70H)   # LOADS INCOMPLETE C3(4)
3570        LD   A,(3E72H)
3580        ADD  HL,DE
3590        ADC  0            # TO GET VALUE OF C3(4)
3600        LD   (3E70H),HL   # STORE C1(4) IN 3E72H, 3E71H AND
3610        LD   (3E72H),A    # 3E70H. ALSO STORE C1(4) IN 3E78H
3620        LD   (3E76H),HL   # 3E77H AND 3E76H.
3630        LD   (3E78H),A
3640        LD   C,A
3650 LOOPS  NOP
3660        LD   DE,(3E73H)   # VA = C1(4) - C1(4).  LOAD C2(4)
3670        LD   A,E          # GET VALUE OF C2(4)
3680        CPL               # TAKE COMPLEMENT AND ADD TO C1(4)
3690        LD   E,A          # C1(4) IS PRESENT IN CHL REGISTER
3700        LD   A,D
3710        CPL
3720        LD   D,A
```

```
3725 *           .... CONTD OF MAIN PROGRAM
3726 *
3730       LD    A,(3E75H)
3740       CPL
3750       ADD   HL,DE
3760       ADC   C             # VA(4) IS IN AHL REGISTER
3770       LD    (3E62H),A     # VA(4) IS STORED IN 3E62H, 3E61H
3780       LD    (3E60H),HL    # AND 3E60H
3790       LD    C,A           # VA(4) IS IN 2'S COMPLEMENT.
3800       AND   80H           # CHECK IF VA(4) IS +VE OR -VE.
3810       JR    NZ,LOOPU      # IF VA(4) IS -VE THEN JUMP TLOOPU
3820       LD    A,C           # VA(4) IS PRESENT IN 24 BITS OF
3830       AND   0FFH          # CHL REGISTER. IF VA(4) HAS A
3840       JR    NZ,SAT        # MAGNITUDE > THEN 11 BITS (2048)
3850       LD    A,H           # IN BINARY THEN THE OUTPUT TO D/A
3860       AND   0F8H          # IS SATURATED TO +VE 10 VOLTS.
3870       JR    NZ,SAT        # CHECK FOR 1'S IN TOP 13 BITS OF
3880       LD    A,L           # CHL REGISTER.
3890       CPL                 # 000H=10 VOLTS AND 3FFH=-10 VOLTS
3900       AND   0CH           # FOR D/A TO CART. CONVERT VA(4)
3910       LD    B,04H         # ACCORDINGLY. DIVIDE VA(4) BY 16
3920 BEG   SLA   L
3930       RL    H
3940       DJNZ  BEG
3950       LD    L,A
3960       LD    A,H
3970       CPL
3980       AND   7FH
3990       JR    FINAL
4000 SAT   LD    L,0           # THIS GIVES THE MAXIMUM OUTPUT
4010       LD    A,0           # VOLTAGE TO D/A.( LOADS A,H 00H)
4020       JR    FINAL
4030 LOOPU LD    A,C           # IF MAGNITUDE OF VA(4) IS
4040       CPL                 # GREATER THEN 2048 IN BINARY OR
4050       AND   7FH           # MORE THAN 11 BITS THEN THE OUT-
4060       JR    NZ,SAT1       # PUT VOLTAGE TO D/A TO MOTOR IS
4070       LD    A,H           # SATURATED TO -10 VOLTS.
4080       CPL                 # SINCE VA(4) IS -VE AND 2'S COMP
4090       AND   0F8H          # CHECK IF ANY 0'S IN MOST
4100       JR    NZ,SAT1       # SIGNIFICANT BITS. IF TRUE THEN
4110       LD    A,L           # JUMP TO SAT1.
4120       CPL                 # 000H=10 VOLTS AND 3FFH=-10 VOLTS
4130       AND   0CH           # HENCE CONVERT ACCORDINGLY.
4140       LD    B,04H         # DIVIDE VA(4) BY 16
4150 NEG   SLA   L
4160       RL    H
4170       DJNZ  NEG
4180       LD    L,A
4190       LD    A,H
4200       CPL
4210       OR    80H
4220       JR    FINAL
4221       NOP
```

```
4222 *          .... CONTD OF MAIN PROGRAM.
4223 *
4230 SAT1   LD   L,0CH        # OUTPUT TO D/A IS=-10 VOLTS HENCE
4240        LD   A,0FFH       # STORE ALL 1'S IN A AND C REGI
4250 FINAL  LD   (0FFF1H),A   # THE MSB 8 BITS OF THE D/A IS
4260        LD   H,A          # ADDRESSED BY 0FFF1H AND THE 2
4270        LD   A,(3E0DH)    # LSB BITS ARE THE THIRD AND THE
4280        OR   L            # FOURTH BITS. AND IT IS ADDRESSED
4290        LD   (0FFF2H),A   # BY 0FFF2H
4300        SRL  L
4310        SRL  L
4320        LD   (3E41H),HL   # STORE OUTPUT TO D/A
4330        LD   A,(3E75H)    # LOAD C2(3) IN AHL REGISTER
4340        LD   HL,(3E73H)
4350        LD   IX,3E18H     # ADDRESS FOR EXPONENT OF C2(3)
4360        LD   IY,3E12H     # ADDRESS FOR SIGN OF C2(3)
4370        CALL TRANS
4380        LD   (3E24H),HL   # STORE MAGNITUDE OF C2(3)
4390        LD   A,(3E72H)    # LOAD THE VALUE OF C1(3) IN
4400        LD   HL,(3E70H)   # AHL REGISTER
4410        LD   IX,3E17H     # ADDRESS FOR EXPONENT OF C1(3)
4420        LD   IY,3E10H     # ADDRESS FOR SIGN OF C1(3).
4430        CALL TRANS
4440        LD   (3E20H),HL   # STORE MAGNITUDE OF C1(3).
4450        JP   MAIN
4560
4570 ***************************************************************
```

```
*                    SUBROUTINE MULT.

*************************************************************
* THIS SUBROUTINE PERFORMS A 16 BY 8 BIT MULTIPLICATION.
* THE 16 BIT MULTIPLICANT IS PLACED IN DE REGISTER AND
* THE 8 BIT MULTIPLIER IS INDIRECTLY ADDRESED BY IX RESG.
* BEFORE THIS SUBROUTINE IS CALLED.  THE FINAL RESULT
* OF MULTIPLICATION IS PLACED IN CHL REGISTER.

*############################################################

        ST   3400H
STOP    EQU  34DFH
        CALL STOP
        LD   B,08H        # COUNTER IS INITILISED.
        LD   A,0          # INITILISE A,H,L AND C REGI TO 0.
        LD   H,A
        LD   L,A
        LD   C,A
LOOP1   SRL  (IX+0)       # SHIFT RIGHT MULTIPLIER INDIRECTLY
        JR   NC,SKIP      # ADDRESSED BY IX REG.
        ADD  HL,DE
        ADC  C
SKIP    SLA  E          # THE MULTIPLICAND IS ROTATED LEFT
        RL   D          # THROUGH CARRY. THE CARRY IS SHIFTED
        RL   C          # LEFT IN THE 'E' REGISTER.
        DJNZ LOOP1
        LD   C,A
        RET
*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


*                    SUBROUTINE STAD.

*************************************************************
* THE SIGN BIT OF A 23 BIT VALUE IS PLACED IN 'A' REG.
* AND THE MAGNITUDE IS PLACED IN CHL REGISTER.  IF THE
* SIGN IS -VE THEN A 2'COMPLEMENT IS TAKEN BY SUBROUTINE
* COMP.  IT THEN ADDS CHL REG TO C'H'L' REG.  THE RESULT
* IS PLACED IN C'H'L' REGISTER.

*############################################################

        RRA                 # CHECK SIGN.
        JR   NC,LOOPC       # IF SIGN +VE THEN JUMP TO LOOPC.
        CALL 342EH          % REG. IF THE SIGN BIT IS -VE A
LOOPC   LD   A,C            % 2'COMP IS TAKEN OF THE CHL REGS
        PUSH HL             # SAVE HL ON STACK.
        EXX
        POP  DE             # GET VALUE FROM STACK IN DE.
        ADD  HL,DE
        ADC  C
        LD   C,A
        EXX                 # EXCHANGE B,C,D,E,H, AND L REGI
        RET                 # WITH B',C',D',E',H' AND L' REG.
*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
*                          SUBROUTINE COMP

**********************************************************
* THIS SUBROUTINE TAKES THE 2' COMPLEMENT OF CHL REGISTER
* AND STORES IT BACK IN CHL REGISTER.

**********************************************************

        LD   A,L
        CPL
        ADD  01H
        LD   L,A
        LD   A,H
        CPL
        ADC  0
        LD   H,A
        LD   A,C
        CPL
        ADC  0
        LD   C,A
        RET

**********************************************************



*                          SUBROUTINE EXP

**********************************************************
* THIS SUBROUTINE DIVIDES THE CHL REGISTER BY 8.  IT THEN
* MULTIPLIES THE CHL REGISTER BY 2 EXPONENT(VALUE).  THIS
* (VALUE) IS PLACED IN CHL REGISTER.  THE FINAL RESULT IS
* STORED IN CHL REGISTER.

**********************************************************

        SRL  C        # CHL REGISTER IS DIVIDED BY 8.
        RR   H
        RR   L
        SRL  C
        RR   H
        RR   L
        SRL  C
        RR   H
        RR   L
LOP     SLA  L
        RL   H
        RL   C
        DJNZ LOP
        RET

**********************************************************
```

```
*                       SUBROUTINE TRANS
***********************************************************
* CONVERTS C3 AND C4 FROM 24 BITS AND 2'COMPLEMENT WHICH
* IS PLACED IN AHL REGISTER INTO 15 BIT MAGNITUDE,
* SIGN BYTE AND EXPONENT BYTE.
* THE SIGN BYTE IS INDIRECTLY ADDRESSED BY IY REGISTER AND
* THE EXPONENT BYTE IS INDIRECTLY ADDRESSED BY IX REGISTER
* THE 15 BIT MAGNITUDE IS STORED IN HL REGISTER.
***********************************************************
        LD   D,A        # LOAD MOST SIGNIFICANT BYTE.
        SLA  D          # CHECK SIGN
        JR   NC,LOOPX   # IF +VE SIGN THEN JUMP TO LOOPX
        LD   A,01H      # SIGN -VE AND HENCE STORE 01H
        LD   (IY+0),A   # INDIRECTLY ADDRESED BY IY REGISTER
        LD   A,D
        RRCA
        LD   C,A        # TAKE 2'COMPLEMENT OF AHL REGISTER
        LD   A,L        # TO GET MAGNITUDE.
        CPL
        ADD  01H
        LD   L,A
        LD   A,H
        CPL
        ADC  0
        LD   H,A
        LD   A,C
        CPL
        ADC  0
        AND  7FH
        JR   LOO
LOOPX   LD   D,A
        LD   A,0         # SIGN +VE AND HENCE STORE 00H
        LD   (IY+0),A    # INDIRECTLY ADDRESED BY IY REG.
        LD   A,D
LOO     LD   C,A
        LD   D,A
        LD   A,09H
        LD   B,08H
LOOPY   SLA  C
        JR   C,LOM
        DEC  A
        DJNZ LOOPY
LOM     LD   B,A         # STORE EXPONENT VALUE INDIRECTLY
        LD   (IX+0),A    # ADDRESED BY IX REGISTER.
LOOPZ   SRL  D           # IF MAGNITUDE GREATER THAN 15 BITS
        RR   H           # THEN SAVES THE FIRT 15 BITS IN HL
        RR   L           # REGISTER.
        DJNZ LOOPZ
LO      RET
***********************************************************
```

```
*                     SUBROUTINE SHIFT
**********************************************
* THIS SUBROUTINE SHIFTS THE MAGNITUDE AND SIGN OF V(3)
* TO MEMORY LOCATION OF V(2).  MAGNITUDE, SIGN AND EXPO
* OF C2(2) TO MEMORY LOCATION OF C2(1).  MAGNITUDE , SIGN
* AND EXPONENT OF C2(3) TO MEMORY LOCATION OF C2(2)
**********************************************
          LD   HL,(3E2AH)    # MAGNITUDE OF V(3) TO V(2)
          LD   (3E2CH),HL
          LD   HL,(3E26H)    # MAGNITUDE OF C4(2) TO C4(1)
          LD   (3E28H),HL
          LD   HL,(3E24H)    # MAGNITUDE OF C2(3) TO C2(2)
          LD   (3E26H),HL
          LD   A,(3E15H)     # SIGN OF V(3) TO V(2)
          AND  01H
          LD   (3E16H),A
          LD   A,(3E13H)     # SIGN OF C2(2) TO C2(1)
          AND  01H
          LD   (3E14H),A
          LD   A,(3E12H)      # SIGN OF C2(3) TO C2(2)
          AND  01H
          LD   (3E13H),A
          LD   A,(3E19H)      # EXPONENT OF C2(2) TO C2(1)
          LD   (3E1AH),A
          LD   A,(3E18H)      # EXPONENT OF C2(3) TO C2(2)
          LD   (3E19H),A
          RET
**********************************************


*                     SUBROUTINE STOP

**********************************************
* IF THE LETTER 'S' IS PRESSED FROM THE KEYBOARD THEN
* THE EXECUTION OF THE PROGRAM IS STOPED.  AT THE SAME
* INSTANT THE OUTPUT VOLTAGE TO THE CART IS MADE 0.  THE
* PAST VALUES OF THE POSITION OF PENDLUM, VOLTAGE OUT TO
* THE CART AND THE VELOCITY OF THE CART ARE DISPLAYED ON
* SCREEN. IF KEY 'A' IS PRESSED THEN THE CONTROLS ARE
* PASSED TO RESTART.
**********************************************
          LD   A,(3E7FH)    # REGISTER A IS SAVED
          IN   0,A          # INPUT FROM KEY BOARD.
          CP   'S'          # CHECK IF KEY 'S' IS DEPRESSED.
          JR   Z,END        # IF YES THEN JUMP TO END.
          LD   A,(3E7FH)    # LOADS REGISTER A WITH ORIGINAL
          RET               # VALUE.
END       IN   0,A
          CP   'A'          # CHECK IF KEY 'A' IS DEPRESSED.
          LD   A,7FH        # OUTPUT TO D/A TO CART =0 VOLTS
          LD   (0FFF1H),A
          JR   NZ,END       # IF KEY 'A' IS DEPRESSED TTHEN
          RST  48D          # RST.
**********************************************
```

```
*                              SUBROUTINE DISPLAY

***********************************************************
* THIS SUBROUTINE DISPLAYS ON SCREEN THE CONTROL VOLTAGE
* OUTPUT (U), PENDULUM POSITION (P), SPEED OF THE A
* ARMATURE OF THE D.C. MOTOR IN HEXIDECIMAL.   THE M
* VALUES ARE DISPLAYED WITH SIGNS.

***********************************************************

             ST    3530H
STOP         EQU   34DFH
             CALL  STOP
             LD    A,0D6H       # ASCII CODE FOR 'V'
             CALL  000372
             LD    A,0BDH       # ASCII CODE FOR '='
             CALL  000372
             LD    A,H
             AND   80H          # TEST SIGN OF CONTROL VOLTAGE
             JR    NZ,POST      # IS -VE SIGN THEN JUMP TO POST
             LD    A,H
             CPL
             AND   7FH
             LD    H,A
             LD    A,L
             CPL
             LD    L,A
             LD    A,0ABH       # ASCII CODE FOR '+'
             JR    NET
POST         LD    A,0ADH       # ASCII CODE FOR '-'
NET          CALL  000372
             LD    A,H          # DISPLAY OUTPUT CONTROL VOLTAGES
             AND   7FH          # IN HEXA DECIMAL OF THE VALUE
             LD    H,A          # PRESENT IN HL REGISTER.
             LD    A,0
             SLA   H
             RLA
             SLA   H
             RLA
             OR    0B0H
             CP    0BAH
             CALL  P,HEX
             CALL  000372
             LD    A,L
             AND   03H
             ADD   H
             LD    B,A
             SRL   A
             SRL   A
             SRL   A
             SRL   A
             OR    0B0H
             CP    0BAH
```

```
             CALL P.HEX
             CALL 000372
             LD   A,B
             AND  0FH
             OR   0B0H
             CP   0BAH
             CALL P.HEX
             CALL 000372
             CALL 000370
             CALL STOP
             LD   HL,(7E22H)   # LOADS MAGNITUDE OF POSITION
             LD   A,0D0H      # OF PENDULUM.  ASCII CODE FOR 'P'
             CALL 000372
             LD   A,0BDH      # ASCII CODE FOR '='
             CALL 000372
             LD   A,(7E11H)
             AND  01H         # TEST SIGN BIT
             JR   NZ,POST1    # IF -VE THEN DISPLAY '-' ON SCREEN
             LD   A,0ABH      # ASCII CODE FOR '+'
             JR   NET1
      POST1  LD   A,0ADH      # ASCII CODE FOR '-'
      NET1   CALL 000372
             LD   A,H         # DISPLAY MAGNITUDE FOR POSITION OF
             AND  07H         # PENDULUM IN HEX OF THE VALUE S
             OR   0B0H        # STORED IN HL REGISTER.
             CALL 000372
             LD   A,L
             SRL  A
             SRL  A
             SRL  A
             SRL  A
             OR   0B0H
             CP   0BAH
             CALL P.HEX
             CALL 000372
             LD   A,L
             AND  0FH
             OR   0B0H
             CP   0BAH
             CALL P.HEX
             CALL 000372
             CALL 000370
             CALL STOP
             LD   A,0D3H      # ASCII CODE FOR 'S'
             CALL 000372
             LD   A,0BDH      # ASCII CODE FOR '='
             CALL 000372
             LD   A,(7E15H)   # LOADS SIGN OF BELOCITY
             AND  01H         # TEST SIGN
             JR   NZ,POST2    # IF -VE SIGN THEN DISPLAY '-'
             LD   A,0ABH      # ASCII CODE FOR '+'
             JR   NET2
```

```
7800 POST2  LD   A,0ADH      # ASCII CODE FOR '-'
7810 NET2   CALL 000373
7820        LD   HL,(3E2AH)
7830        LD   A,H         # DISPLAY THE MAGNITUDE FOR VE
7840        AND  07H         # VELOCITY OF CART IN HEX OF THE
7850        OR   0B0H        # VALUE PRESENT IN HL REGISTER.
7860        CALL 000373
7870        LD   A,L
7880        SRL  A
7890        SRL  A
7900        SRL  A
7910        SRL  A
7920        OR   0B0H
7930        CP   0BAH
7940        CALL P,HEX
7950        CALL 000373
7960        LD   A,L
7970        AND  0FH
7980        OR   0B0H
7990        CP   0BAH
8000        CALL P,HEX
8010        CALL 000373
8020        CALL 000370
8030        LD   A,(3E4FH)
8040        DEC  A
8050        LD   (3E4FH),A
8060        JR   NZ,SKIP
8070        CALL 000370
8080        LD   A,03H
8090        LD   (3E4FH),A
8100        CALL STOP
8110 SKIP   RET
8120 HEX    ADD  7
8130        RET
8140
8150
8160 ************************************************************
```

```
8170 *                          SUBROUTINE STORE
8180
8190 ***************************************************************
8200 * STORE VALUES OF THE SPEED OF THE ARMATURE FROM 3E80H
8210 * TO 4000H AND THE PENDULUM POSITION FROM 2E90H TO 3000H
8220 * EACH VALUE IS STORED IN 3 BYTES.  THE FIRST TWO BYTES
8230 * CONTAINS THE MAGNITUDE AND THE THIRD CONTAINS THE SIGN.
8240 * 01H INDICATES -VE SIGN WHILE 00H INDICATE +VE SIGN.
8250 * C1 IS STORED FROM 3900H TO 3B00H. C2 FROM 3B00H TO 3C80H
8260 * VA FROM 3C80H TO 3E00H.  C1, C2, VA ARE STORED IN 3
8270 * BYTES IN 2' COMPLEMENT FORMAT.  AFTER THE ENTIRE MEMORY
8280 * SPACE FOR THE VALUES ARE FULL, THE COUNTER IS INITIAL-
8290 * IZED TO STORE THE VALUES FROM THE BEGINNING ADDRESS
8291 * THIS SUBROUTINE ALSO STOPS THE EXECUTION OF THE MAIN
8300 * PROGRAM AFTER A FIXED NUMBER OF CYCLES.
8310
8320 ***************************************************************
8330
8340 STOP    EQU   34DFH
8350 END     EQU   34ECH
8360         ST    3E40H
8370 STORE   LD    A,(3E50H)      # CHECK IF ENTIRE MEMORY IS
8380         DEC   A              # FILLED UP.
8390         LD    (3E50H),A
8400         JR    NZ,INITI
8410         LD    A,80H
8420         LD    (3E50H),A
8430         LD    HL,3E80H     # INITIALIZE ADDRESS FOR V(3)
8440         LD    (3E51H),HL
8450         LD    HL,2E90H     # INITIALIZE ADDRESS FOR P(3)
8460         LD    (3E53H),HL
8470         LD    HL,3900H     # INITIALIZE ADDRESS FOR C1
8480         LD    (3E55H),HL
8490         LD    HL,3B00H     # INITIALIZE ADDRESS FOR C2
8500         LD    (3E57H),HL
8510         LD    HL,3C80H     # INITIALIZE ADDRESS FOR VA
8520         LD    (3E59H),HL
8530 INITI   LD    A,(3E15H)    # LOADS SIGN OF V(3)
8540         LD    DE,(3E2AH)   # LOAD MAGNITUDE OF V(3).
8550         LD    HL,(3E51H)   # LOADS ADDRESS OF V(3).
8560         LD    (HL),E       # STORE MAGNITUDE AND SIGN OF V(3)
8570         INC   HL
8580         LD    (HL),D
8590         INC   HL
8600         LD    (HL),A
8610         INC   HL
8620         LD    (3E51H),HL   # STORE UPDATED ADDRESS BACK.
8630         CALL  STOP
8640         LD    A,(3E11H)     # LOADS SIGN OF P(3).
8650         LD    DE,(3E22H)    # LOADS MAGNITUDE OF P(3)
8660         LD    HL,(3E53H)    # LOADS ADDRESS OF P(3)
8670         LD    (HL),E        # STORE MAGNITUDE AND SIGN OF P(3
8680         INC   HL
8690         LD    (HL),D
```

```
8700          INC   HL
8710          LD    (HL),A
8720          INC   HL
8730          LD    (3E53H),HL   # STORE UPDATED ADDRESS BACK
8740          LD    A,(3E78H)    # LOAD MOST SIGNIFICANT BYTE OF C1
8750          LD    DE,(3E79H)   # LOAD THE LAST TWO BYTES OF C1
8760          LD    (HL),E       # STORE MAGNITUDE OF C1
8770          LD    (HL),E
8780          INC   HL
8790          LD    (HL),D
8800          INC   HL
8810          LD    (HL),A
8820          INC   HL
8830          LD    (3E55H),HL    # STORE UPDATED ADDRESS OF C1
8840          CALL  STOP
8850          LD    A,(3E75H)     # LOAD MS BYTE OF C2
8860          LD    DE,(3E73H)    # LOAD THE LAST TWO BYTES OF C2
8870          LD    HL,(3E75H)    # LOAD ADDRESS OF C2
8880          LD    (HL),E        # STORE VALUE OF C2
8890          INC   HL
8900          LD    (HL),D
8910          INC   HL
8920          LD    (HL),A
8930          INC   HL
8940          LD    (3E57H),HL    # STORE UPDATED ADDRESS OF VA
8950          LD    A,(3E83H)     # LOAD M.S. BYTE OF VA
8960          LD    DE,(3E80H)    # LOAD LAST TWO BYTES OF V1
8970          LD    (HL),E        # LOAD ADDRESS OF VA
8980          LD    (HL),E        # STORE VALUE OF VA IN MEMORY
8990          INC   HL
9000          LD    (HL),D
9010          INC   HL
9020          LD    (HL),A
9030          INC   HL
9040          LD    (3E59H),HL
9050          CALL  STOP
9060          LD    HL,(3E43H)     # LOAD COUNTER - NO OF SAMPLES
9070          LD    (3E43H),HL     # STORE BACK COUNTER.
9080          LD    (3E43H),HL
9090          LD    A,H
9100          AND   0FFH           # CHECK IF COUNTER=0
9110          JR    Z,END1
9120          RET                  # IF COUNTER=0 THEN STOP THE
9130 END1     LD    A,L            # HALT THE EXECUTION OF PROGRAM
9140          AND   0FFH
9150          JR    Z,END2
9160          RET
9170 END2     JP    END
9180
9190 ****************************************************************
```

```
*                          SUBROUTINE LAST

**********************************************************
* THIS SUBROUTINE IS USED TO BIAS THE PENDLUM POSITION
* READIND AND STORE THE CORRECTION FACTOR FOR THE VELOCITY
* READING FOR THE CART IN STATIONARY CONDITION.  THE
* BIASING IS DONE BY DEPRESSING THE KEY 'I' AND 'D' WHICH
* INCREASES AND DECREASES THE BIASING VOLTAGE FROM THE
* D/A.  TO RETURN TO THE MAIN PROGRAM KEY 'E' SHOULD BE
* DEPRESSED.
* THIS SUBROUTINE HAS ACCESS TO SIX MORE SUBROUTINE WHICH
* ARE 1) DEL1, 2) INC, 3) DEC, 4) END, 5) DELAY AND
* 6) SCREEN.

**********************************************************

        ST    3700H
        LD    HL,01CDH        # LOADS INITIAL VALUE FOR BIAS
        LD    (3E07H),HL
        LD    HL,0
        LD    (3E7BH),HL
        LD    (3E7DH),HL
LOOP    LD    HL,(3E07H)      # CONVERT THE LAST 10 BITS IN
        LD    B,L             # HL REG IN 8 BITS IN H REG AND
        SLA   A               # LAST 2 BITS OF L REGISTER.
        RL    H
        SLA   L
        RL    H
        SLA   L
        RL    H
        SLA   L
        RL    H
        SLA   L
        RL    H
        SLA   L
        RL    H
        LD    A,H
        LD    (0FFF0H),A      # OUTPUT D/A
        LD    A,B
        LD    (0FFF2H),A
        LD    A,0D0H          # ASCII CODE FOR 'P'
        CALL  000372
        LD    A,0C2H          # ASCII CODE FOR 'B'
        CALL  000372
        LD    A,0BDH          # ASCII CODE FOR "=".
        CALL  000372
        LD    A,(3E0BH)       # INITIALISE A/D TO READ PENDLUM
        LD    (0FFF3H),A
LOOP1   LD    A,(0FFF1H)      # CHECK WHETHER A/D CONVERSION IS
        AND   80H
        JR    Z,LOOP1
LOOP2   LD    A,(0FFF1H)
```

```
        AND   80H
        JR    NZ,LOOP2
        LD    A,(0FFF2H)    # READ MOST SIGNIFICANT NIBBLE.
        AND   0FH
        LD    H,A
        LD    A,(0FFF0H)
        LD    L,A
        CALL  SCREEN
        LD    A,(3E0FH)     # INITIALISE A/D TO READ TVELOCITY
        LD    (0FFF3H),A
        CALL  DEL1
        LD    DE,07FFH      # SUBTRACT BIAS AND ADD PREVIOUS
        LD    A,(3E7BH)     # BIAS TO IT. THEN DIVIDE BY 2.
        ADD   0
        SBC   HL,DE
        ADD   L
        SRL   A
        LD    (3E7BH),A     # STORE BIAS.
        CALL  DEL1
        LD    A,0D6H         # ASCII CODE FOR 'V'
        CALL  000372
        LD    A,0C2H         # ASCII CODE FOR 'B'
        CALL  000372
        LD    A,0BDH         # ASCII CODE FOR '='
        CALL  000372
        LD    A,(3E0FH)      # INITIALISE A/D TO READ VELOCITY
        LD    (0FFF3H),A
        CALL  DEL1
        CALL  DEL1
        LD    A,(0FFF2H)     # READ MOST SIGNIFICANT NIBBLE.
        AND   0FH
        LD    H,A
        LD    A,(0FFF0H)    # LOAD LEAST SIGNIFICANT BYTE
        LD    L,A
        CALL  SCREEN
        LD    DE,07FFH      # SUBTRACT BIAS FOR VELOCITY. ADD
        LD    A,(3E7DH)     # PREVIOUS BIAS. DIVIDE BY 2
        ADD   0
        SBC   HL,DE
        ADD   L
        SRL   A
        LD    (3E7DH),A     # STORE BIAS OF VELOCITY.
        LD    A,(3E0BH)     # INITIALISE A/D TO READ POSITION
        LD    (0FFF3H),A    # OF PENDLUM.
        CALL  DEL1
        CALL  DEL1
        CALL  DEL1
        CALL  DEL1
        JP    LOOP
```

```
*               SUBROUTINE DEL1

* THIS SUBROUTINE CHECKS IF ANY INPUT FROM THE KEY BOARD
* IS MADE TO EITHER INCREASE OR DECREASE THE BIAS VOLTAGE
* FOR THE READING OF THE POSITION OF THE PENDLUM OR TO
* RETURN BACK TO THE MAIN PROGRAM.
DEL1    LD   B,3FH          # LOAD VALUE IN COUNTER.
JM      IN   0,A
        CP   'D'            # CHECK IF KEY 'D' IS DEPRESSED.
        CALL Z,DEC
        IN   0,A
        CP   'I'            # CHECK IF KEY 'I' IS DEPRESSED
        CALL Z,INC
        IN   0,A
        CP   'E'            # CHECK IF KEY 'E' IS DEPRESSED.
        JR   Z,END
        DJNZ JM             # DECREMENT COUNTER. IN NZ JP DEL1
        RET


*                      SUBROUTINE DEC

* DECREMENTS THE OUTPUT VOLTAGE FROM THE D/A WHICH IS USED
* TO BIAS THE POSITION OF THE PENDLUM
DEC     LD   HL,(3E07H)
        INC  HL
        LD   (3E07H),HL
        CALL DELAY
        RET


*                  SUBROUTINE INC

* INCREASES THE VOLTAGE OUTPUT FROM THE D/A TO BIAS THE
* READING OF THE PENDLUM POSITION.
INC     LD   HL,(3E07H)
        DEC  HL
        LD   (3E07H),HL
        CALL DELAY
        RET


*                      SUBROUTINE END

* SAVE THE LAST TWO BITS OF THE BIAS OUTPUT FOR THE
* BIAS OUTPUT VOLTAGE TO THE PENDLUM READING AND RETURNS
* BACK TO THE MAIN PROGRAM.
END     LD   A,(3E07H)
        AND  03H
        JP   3021H
DELAY   LD   C,4FH
LAB     DEC  C
        JR   NZ,LAB
        RET
```

```
*                      SUBROUTINE SCREEN

* DISPLAYS ON SCREEN IN HEX THE VALUE WHICH IS PLACED
* IN THE LAST 12 BITS OF THE HL REGISTER.
SCREEN LD   A,H
       AND  7FH
       OR   0B0H
       CP   0BAH
       CALL P,HEX
       CALL 000372
       LD   A,L
       SRL  A
       SRL  A
       SRL  A
       SRL  A
       OR   0B0H
       CP   0BAH
       CALL P,HEX
       CALL 000372
       LD   A,L
       AND  0FH
       OR   0B0H
       CP   0BAH
       CALL P,HEX
       CALL 000372
       CALL 000370
       CALL 000370
       RET
HEX    ADD  07H
       RET
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&



*                    SUBROUTINE PENDIS.

*************************************************************
* THIS SUBROUTINE DISPLAYS THE POSITION OF THE PENDLUM.
* ON THE SCREEN.  HENCE THE PENDLUM POSITION CAN BE
* INITIALISED AT TIME T=0.  IF KEY "R" IS DEPRESSED
* THEN THE EXECUTION OF THE MAIN PROGRAM IS BEGUN.

#############################################################

SCREEN EQU  3814H
       ST   3850H
       LD   A,(3E0BH)      # SELECT CHANNEL A& AMPLIFICATION
       LD   (0FFF3H),A     # TO READ POSITION OF PENDLUM.
       LD   A,05H
DEL1   LD   B,0FFH         # DELAY.
DEL    NOP
       DJNZ DEL
       DEC  A
       JR   NZ,DEL1
```

```
1310 AGAIN  LD    A,0D0H       # ASCII CODE FOR FOR 'P'.
1320        CALL  000372
1330        LD    A,0BDH       # ASCII CODE FOR '='.
1340        CALL  000372
1350        LD    A,(3E0BH)
1360        LD    (0FFF3H),A
1370 POST   LD    A,(0FFF1H)   # TEST THE MSB TO CHECK IF THE
1380        AND   80H          # A/D CONVERSION IS OVER.
1390        JR    Z,POST
1400 POST1  LD    A,(0FFF1H)
1410        AND   80H
1420        JR    NZ,POST1
1430        LD    A,(0FFF2H)   # LOADS MOST SIGNIFICANT NIBBLE.
1440        AND   08H
1450        JR    Z,SKIP
1460        LD    A,0          # SAVE THE SIGN IN MEMORY 3E11H.
1470        LD    (3E11H),A
1480        LD    A,0ABH       # ASCII CODE FOR '+'.
1490        CALL  000372
1500        LD    A,(0FFF2H)
1510        AND   07H
1520        LD    H,A
1530        LD    A,(0FFF2H)   # LOAD LEAST SIGNIFICANT BYTE.
1540        LD    L,A
1550        LD    DE,(3E7BH)
1560        SBC   HL,DE
1570        LD    A,H
1580        AND   80H
1590        JR    Z,SKIP2
1600        LD    HL,0
1610        JR    SKIP2
1620 SKIP   LD    A,01H        # SAVE THE SIGN.
1630        LD    (3E11H),A
1640        LD    A,0ADH       # ASCII CODE FOR '-'.
1650        CALL  000372
1660        LD    A,(0FFF2H)   # LOADS MOST SIGNIFICANT NIBBLE.
1670        CPL
1680        AND   07H
1690        LD    H,A
1700        LD    A,(0FFF0H)   # LOADS LEAST SIGNIFICANT BYTE.
1710        CPL
1720        LD    L,A
1730        LD    DE,(3E7BH)
1740        ADD   HL,DE
1750 SKIP2  LD    (3E22H),HL
1760        CALL  SCREEN
1770        IN    0,A          # LOADS THE INPUT FROM KEY-BOARD
1780        CP    'R'          # CHECK TO SEE WHETHER'R'WAS
1790        JR    Z,END        # DEPRESSED.  IF TRUE THEN JUMPS
1800        JR    AGAIN        # BACK TO THE MAIN PROGRAM.
1810 END    RET
1820 *%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

ACKNOWLEDGEMENT

The author wishes to extend his sincere appreciation and gratitude to Dr. J. Garth Thompson for his continuous guidance and advise throughout the graduate program. The author would also like to thank Dr. Ralph Turnquist and Dr. R. R. Gallagher for serving on the advisory committee.

The author also expresses his thanks to Dr. Paul L. Miller, Professor and Head of Mechanical Engineering Department, for extending financial support during the graduate studies.

The author is also thankful to Mr. B. Shirley and Mr. D. Marvin for their help in aiding this research.

Finally, the author wishes to extend his sincere appreciation to Mr. Gopal Ananth for proof reading this thesis.

DESIGN AND IMPLEMENTATION OF A MICROCOMPUTER, CONTROLLER FOR STABILIZING AN INVERTED PENDULUM MOUNTED ON A CART.

by

PARAG BRIJENDRA BHARGAVA

B.E., Bombay University, India, 1980

AN ABSTRACT OF A THESIS

Submitted in partial fulfillment of the requirement for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1983

ABSTRACT

In recent years, especially since the advent of the microcomputer, great advances have been made in the field of discrete control systems. A real-time, discrete controller was developed and implemented to stabilize a naturally unstable system. An inverted pendulum mounted on a motor driven cart was used.

A mathematical model of the system was developed and the parameters of the model were obtained by frequency response analysis. The discrete model was obtained by applying the Z-transforms. The naturally unstable system was stabilized by using compensators developed by the Z-plane analysis technique. Simulation results were obtained for various values of the system gain.

A Z-80 microcomputer, was used as the real-time controller for the experimental investigation. The results of the experimental investigation compare favorably with the result predicted by the analysis and the simulation. Recommendations for further study are provided.