A NO LOAD SIMULATION MODEL OF A DC DRIVE SYSTEM

by

Guy Mandel Dudley

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

APPROVED:

Dr. L.L. Grigsby, Chairman

Dr. C.E. Nunnally

Dr. R.G. Mitchiner

Dr. F.C. Brockhurst

May, 1983

Blacksburg, Virginia

ACKNOWLEDGEMENTS

I would like to express my appreciation to the Department of Electrical Engineering, Virginia Polytechnic Institute and State University, for allowing me to conduct this research. I also thank the Industrial Drives Division of Kollmorgen for the use of their DC drive system.

Mr. Don Betz, Mr. Bill Shumate, and Mr. Dennis Newman of the Electrical Engineering shop have been very supportive throughout this entire research effort, to them I owe many thanks. Also, I would like to thank Ms. Anitta Chapman, Ms. Vicki Trump, and Ms. Pat Wojciechouski for their aid in the preparation of this document. Furthermore, I am grateful to each of the hard working members of my committee, Dr. C.E. Nunnally, Dr. R.G. Mitchiner, and Dr. F.C. Brockhurst, for their personal contributions to this research. Also, I would like to extend my gratitude to the chairman of my committee, Dr. L.L. Grigsby, for the guidance, patience and encouragement that he extended to me as a graduate student, without which this research would not have been possible. Finally, I thank my parents, Mr. and Mrs. Guy Danion Dudley, for their everpresent concern and support.

TABLE OF CONTENTS

Appendix

LIST OF FIGURES

LIST OF TABLES

Chapter I

INTRODUCTION

> 'Electric servo motors are used as feed drives, for example, machine tools, handling systems and industrial robots. They are usually the positioning modules of a positional servo loop with numerical control, and produce the relative movements between tool and workpiece. They accelerate and decelerate machine rails, for instance, via a ballscrew, and absorb the forces generated during workpiece machining without allowing any changes of position. The requirements on such a drive are very high.' [1]

The concluding sentence of the preceding quotation emphasizes the demands on the performance of a DC drive system. Therefore, if these systems are to continue to satisfy these high requirements, the design and development of these systems must take advantage of the powerful attributes of Modern Control Theory. Brogan [2], has indicated that the stimulus for the development of Modern Control Theory has come from several factors: a)the necessity of dealing with more realistic models of systems, b)the shift in emphasis towards optimal control and optimal system design, c)the continuing developments in digital computer technology, d)the shortcomings of previous approaches, e)a recognition of the applicability of well-known methods in other fields of knowledge.

In DC drive systems, the classical specifications such as percent overshoot and settling time, may also be matched in importance by such optimal criteria as minimum energy, minimum time of operatation or minimum cost. An excellent example of such a case is given in the work by Safiuddin [3]. In this work, optimal control techniques are applied to a DC drive system. In the selection of the performance index, the author explains that while minimizing the time to reach the final state may be a desirable objective for certain systems, minimizing the energy losses during acceleration and/or deceleration is a preferred objective for generalized applications. Safiuddin also notes that the energy loss in a DC drive system can be classified into four categories as listed below [3].

*1. Excitation losses in transformers and motor windings

2. Diode/thyristor and brush drop losses in power converters and DC motors

3. I^2R losses in the electrical circuits

4. Friction losses in the bearings, couplings and rotating parts, etc.

For such a type of drive system, the author suggests that the integral of the current squared over the acceleration period would provide the best performance index, and the minimization of this index would minimize the losses.

Another major area of Modern Control Theory is digital control. Because of the rapid advances in LSI technology, smaller, faster and lower costing microprocessors and peripheral circuits are available. Thus, the application of microprocessor technology to DC drive systems is becoming more and more commonplace. Examples of the application of microprocessors to drive systems are given in references [4],[5],[6],[7],[8],[9]. The digital control examples illustrate systems where finite - settling time (deadbeat) control [6], linear digital regulator [10], digital PID (Proportional - Integral - Derivative) control [10], as well as other techniques have been applied. However, it has been shown [5], that the microprocessor computation time may not be able to be neglected when the processor is placed in a drive system control loop.

In each of the previously discussed examples, the state variable model for the drive system was given before the control algorithm was designed. The purpose of this thesis is to report on the development of a simulation model of a specific DC drive system. The development of such a model is the first step in applying Modern Control Theory. The modeling technique that is used in this work is a technique for simulation of nonlinear, as well as, linear networks that was developed by Kuelz [11]. A few other papers [12],[13],[14],[15] address this technique in detail. This method uses 4 matrices (A,J,N,X) to describe a system. The A matrix describes the system's dynamics between samples. The Jump matrix, J, handles both the real and the introduced sampling in the system at every sample instant; any state variable not associated with a nonlinearity has no jump at the sample instant. The Nonlinear description matrix, N, handles the implementation of each nonlinearity. This matrix contains such elements as saturation limits, coefficents of nonlinear equations, and counters, to describe a nonlinearity. The X vector contains the system states at each sample period. This technique allows a user to develop a new nonlinear simulation device to fit a given system. In the work that follows, 3 new simulation devices were developed to model the drive system. Once a good simulation model has been obtained, then other areas of Modern Control Theory may be applied to the system.

Chapter II

OVERVIEW OF THE DRIVE SYSTEM OPERATION

The purpose of this section is to discuss the general operation of the drive system to be modeled. The drive system is composed of a Kollmorgen TPAR-3330 Phase Angle (Silicon Controlled Rectifier-SCR) amplifier and a Kollmorgen TT-2953B Samarium Cobalt permanent magnet DC motor. The drive system may be classified as a four-quadrant drive, because the armature voltage supply can supply positive and negative voltage, as well as, positive and negative current. The system is a closed loop system composed of a rate loop with a subordinate current loop (figure 2.1). The electronics in the system may be separated into the following functional modules:

1. Interlock
2. Rate Loop
3. Current Loop
4. Pulse Generators and SCR Bridge

as shown in figure 2.1.

The first functional module to be discussed is the Interlock card. This card has the general responsibility of protecting the drive system. The Interlock card serves three basic functions. It operates the dynamic brake (to stop the motor) if any of the line power is lost. In order to reduce transient surges to the system's electronics, the Interlock card also supplies a small delay when the system is first connected to the source power. The final function of the Interlock card is to stop the drive system in an orderly fashion in case the motor exceeds the rated speed (overspeeds). When the Interlock card stops the drive because of an overspeed condition, the drive cannot be started again until the primary power (line power) has been removed. The removal of the line power indicates that the drive system will be troubleshot to find the fault in the system. The function of the Rate Loop card is to compare the commanded speed of the motor to the actual speed of the motor, and to create a current command signal for the Current Loop card. The commanded speed signal for the drive system is the signal R, in figure 2.1; it is input into an inverting differential amplifier and is scaled before it is summed with the negative of the tachometer signal.

The Current Loop card is the next module to be discussed. The Current Loop card performs two basic functions. First, it controls how much current is delivered to the motor, and it also monitors the motor speed and adjusts the allowable change in current based on the present speed of the motor. The Current Loop card has three inputs:

1. the current command (from the Rate Loop card)
2. a signal proportional to the motor current
3. the tachometer signal.

The outputs of the Current Loop card are the reverse firing signal (fires the reverse SCRs) and the forward firing signal (fires the forward SCRs). The Pulse Generator card accepts the Current Loop card output and converts it to a pulse, whose position, with respect to line voltage crossover, depends on the value of the input signal. Each Pulse Generator contains two channels, both of which are practically identical. One of the channels is used to fire the positive SCRs and the other channel is used to fire the negative SCRs. As the pulse position advances, the current in the motor increases.

As the current in the DC motor increases (decreases) the motor speed will increase (decrease). The output of the tachometer generator is fed back to the Rate Loop card, thus closing the velocity loop. A signal proportional to the motor current is obtained by sampling the voltage across a current sensing resistor that is connected in series with the motor armature leads (figure 2.1). This current signal is fed back to the Current Loop card to close that loop.

The purpose of this section was to supply the reader with a very brief overview of the drive system operation. More detailed information about the system operation may be obtained from the installation manuals [16] and [17].

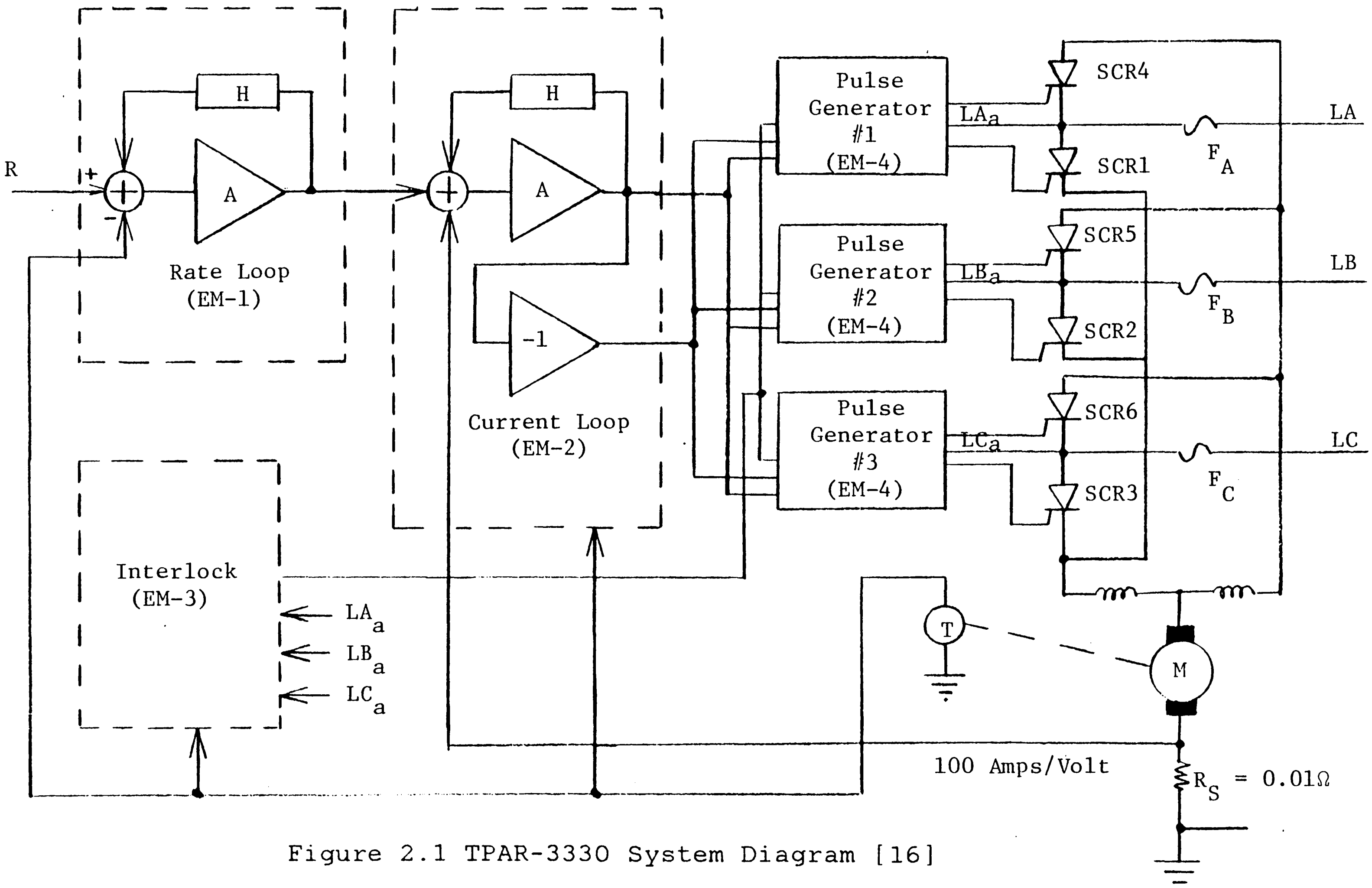


Figure 2.1 TPAR-3330 System Diagram [16]

Chapter III

DEVELOPMENT OF SPECIFIC MODELS

The development of the drive system simulation model was done by functional modules. This section discusses the procedure that was followed, and the assumptions made in the modeling process for each module. In all models, the Interlock circuit is assumed to be in the normal operation mode.

3.1 DC MOTOR MODEL

The circuit diagram for a separately excited, armature controlled, DC motor is given in figure 3.1[18]. If the system is considered to be a speed controlled system, it may be represented by two state variables. The independent state variables for the system are the armature current in amps, (X_1), and the velocity in radians per second, (X_2). The state equation is given by:

$$\dot{\underline{X}} = \begin{bmatrix} \frac{-F}{J} & \frac{K_T}{J} \\ \frac{-K_B}{L_a} & \frac{-R_a}{L_a} \end{bmatrix} \underline{X} + \begin{bmatrix} 0 \\ \frac{1}{L_a} \end{bmatrix} u \qquad (3.1.1)$$

A block diagram representation of this state equation is shown in figure 3.2 The values for these constants were taken from the data sheet that was supplied with the motor [19]. The numerical values and units for the DC motor are the following:

Armature resistance	R_a = 0.199 ohms (12.5%)
Armature inductance	L_a = 0.81 mH (30.0%)
Torque sensitivity	K_T = 0.368 Nm/amp (10.0%)
Back emf constant	K_B = 38.48 V/Krpm (10.0%)
Rotor inertia	$J_m = 2.74\text{x}10^{-3}$ Kg m^2
Viscous damping	F = 0.152 Nm/Krpm
Tachometer sensitivity	K_G = 12.5 V/Krpm (10.0%)

The back emf constant (K_B), viscous damping constant (F), and the tachometer sensitivity (K_G) needed to have their units changed in order to have a consistent set. Therefore, the following calculations were made:

$$K_B = 38.48 \frac{\text{V}}{\text{k rpm}} \times \frac{\text{1K}}{1000} \times 1 \frac{\text{rev}}{2\pi \text{ rad}} \times 60 \frac{\text{sec}}{\text{1 min}} \tag{3.1.2}$$

$$K_B = 0.368 \frac{\text{V}}{\text{rad/sec}}$$

$$F = 0.152 \frac{N \cdot m}{k\ rpm} \times \frac{1K}{1000} \times \frac{60\ sec}{1\ min} \times \frac{1\ rev}{2\pi\ rad} \tag{3.1.3}$$

$$F = 0.00145\ N \cdot m \cdot sec/rad$$

$$K_G = \frac{12.5\ V}{k\ rpm} = \frac{12.5\ V}{1000\ rev/min} \cdot 60\ \frac{sec}{min} \cdot 1\ \frac{1\ rev}{2\pi\ rad} \tag{3.1.4}$$

$$K_G = 0.11937\ \frac{V}{rad/sec}$$

By substitution of the appropriate values, the state equation becomes that of equation 3.1.5

$$\dot{\underline{X}} = \begin{bmatrix} -0.529 & 134.30 \\ -454.32 & -245.68 \end{bmatrix} \underline{X} + \begin{bmatrix} 0 \\ 1234.57 \end{bmatrix} U \tag{3.1.5}$$

Equation 3.1.5 yields the block diagram of figure 3.3.

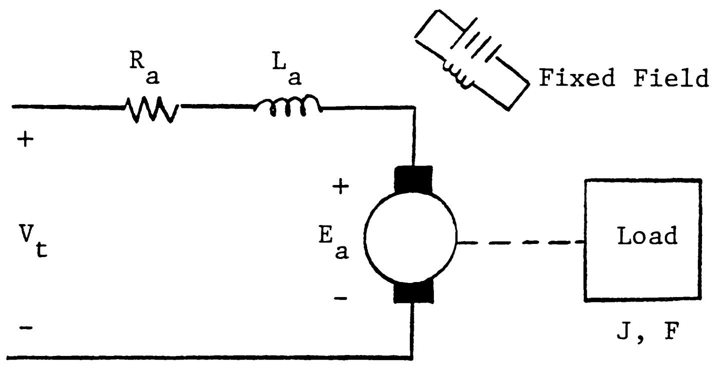


Figure 3.1 Separately Excited DC Motor Circuit Diagram

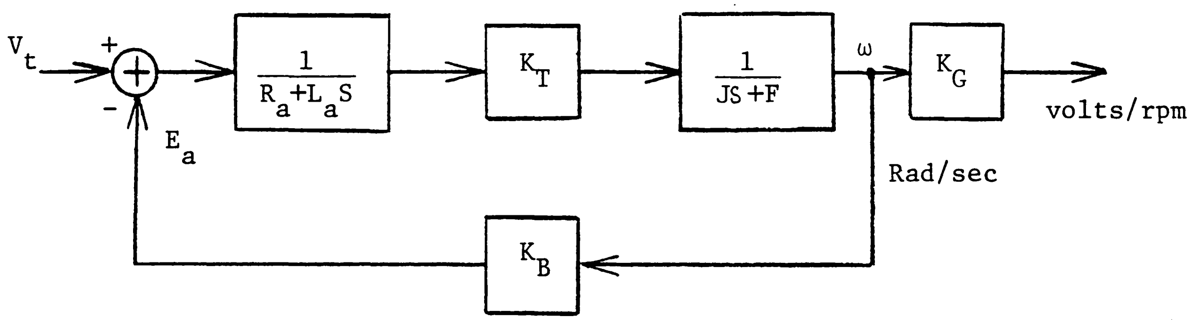


Figure 3.2 DC Motor Block Diagram With General Parameters

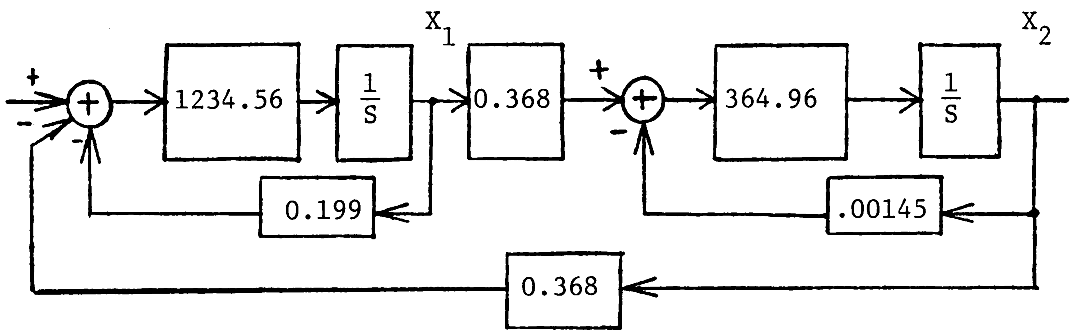


Figure 3.3 DC Motor Block Diagram With Calculated Parameters

3.2 RATE LOOP CARD SIMULATION MODEL - DIFFERENTIAL AMPLIFIER SIMULATION MODEL

The function of the Rate Loop card was briefly explained in chapter 2. The first part of the Rate Loop Card is the differential amplifier network of figure 3.4. This network supplies the negative difference between the input high and the input low as its output. The input high in this network is the speed reference voltage for the drive system. The differential amplifier is a very common circuit in industrial equipment because it discriminates against DC variations, drifts and noise [20]. However, the capacitor,C6, in this circuit makes the analysis of this circuit a formidable task. Using the labels of figure 3.4, the following equations may be obtained by using admittances and nodal analysis.

$$V_3 = \frac{Y_6 V_4 + Y_{14} V_5}{Y_6 + Y_{13} + Y_{14}} + \frac{Y_{13} V_1}{Y_6 + Y_{13} + Y_{14}} \qquad (3.2.1)$$

$$V_4 = \frac{Y_{13} V_2}{Y_6 + Y_{13} + Y_{14}} + \frac{Y_6 V_3}{Y_6 + Y_{13} + Y_{14}} + \frac{Y_{14} V_5}{Y_6 + Y_{13} + Y_{14}} \qquad (3.2.2)$$

$$V_5 = \frac{Y_{14} V_4}{Y_{14} + Y_{18}} \qquad (3.2.3)$$

$$V_0 = \frac{(Y_{14} + Y_8)V_5}{Y_{18}} - \frac{Y_{14}\ V_3}{Y_{18}} \tag{3.2.4}$$

Equations 3.2.1 through 3.2.4 may now be solved for the output voltage (V_0) in terms of the input signals V_1 and V_2. The resulting transfer function, with component values substituted, is given as:

$$V_0 = \frac{-1338.5}{(S+1336)} V_1 + \frac{1338.5}{(S+1336)} V_2 \tag{3.2.5}$$

A simulation diagram is then developed from the transfer function 3.2.5 and it is shown in figure 3.5. The time constant of each of the loops is approximately 0.75 milliseconds, and the DC steady state value of the simulation diagram is approximately -1. x (V_1-V_2).

In order to examine the response of this network, a simulation of the response to a +5 vdc signal on the high input and a +3 vdc signal on the low input was run. The results are shown in figure 3.6, and they show that the output has an exponential response and that it settles to a value of -2.0 vdc in approximately 4 milliseconds. The output of this differential amplifier is connected to a potentiometer whose wiper feeds the velocity loop. This potentiometer sets the value of the Speed Scale Factor (SCF). The SCF determines the volts/rpm value of the reference input. The value of SCF was determined to be 0.442 for this drive system. This value was determined by measuring the ratio of the wiper voltage to the output of the differential amplifier at any point. For this system, the SCF was determined from the -5 vdc operating point.

$$SCF = \frac{\text{wiper voltage}}{\text{differential op amp output}} = \frac{-2.21}{-5.0} = 0.442 \qquad (3.2.6)$$

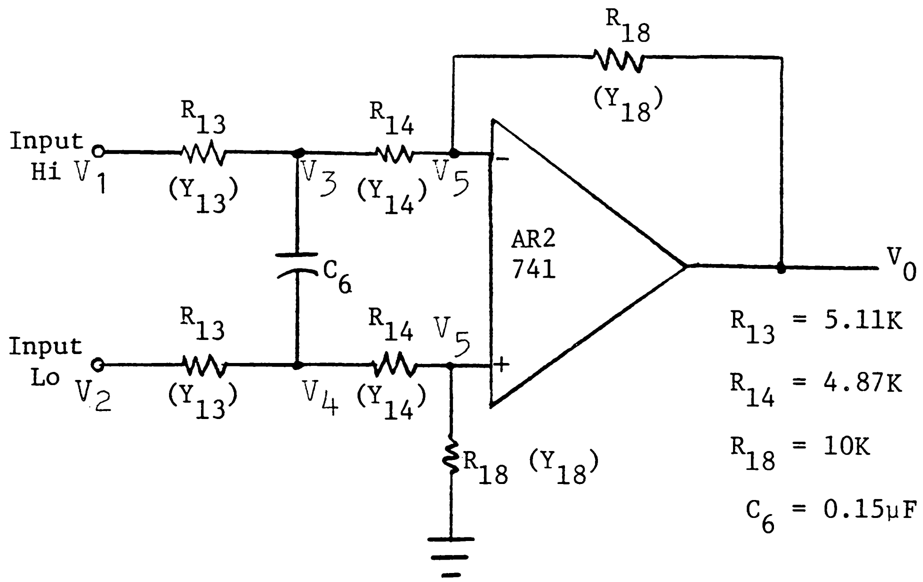


Figure 3.4 Differential Amplifier Network

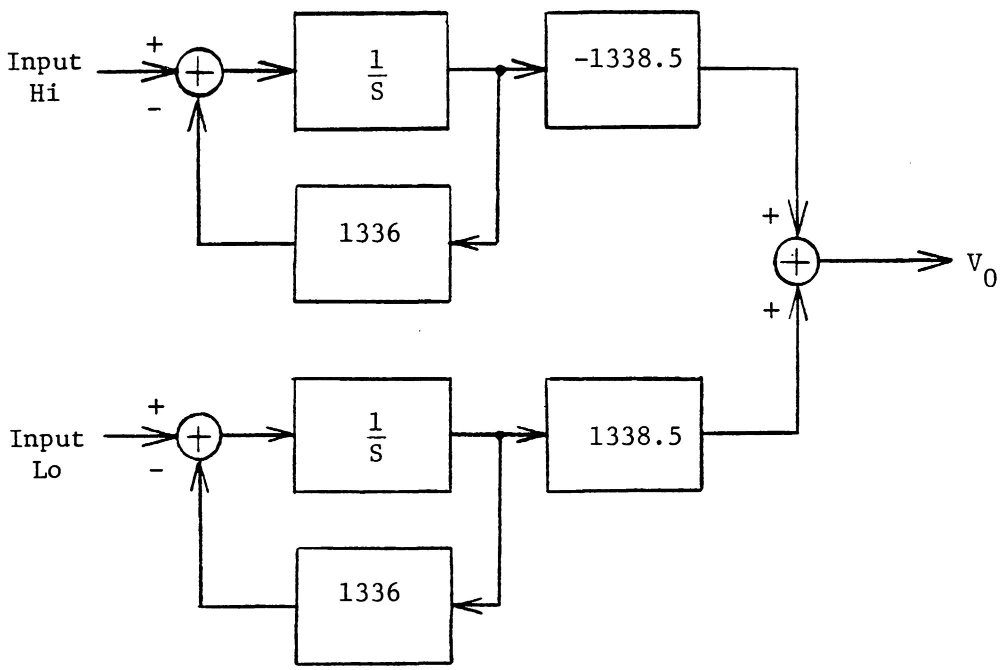


Figure 3.5 Differential Amplifier Simulation Diagram

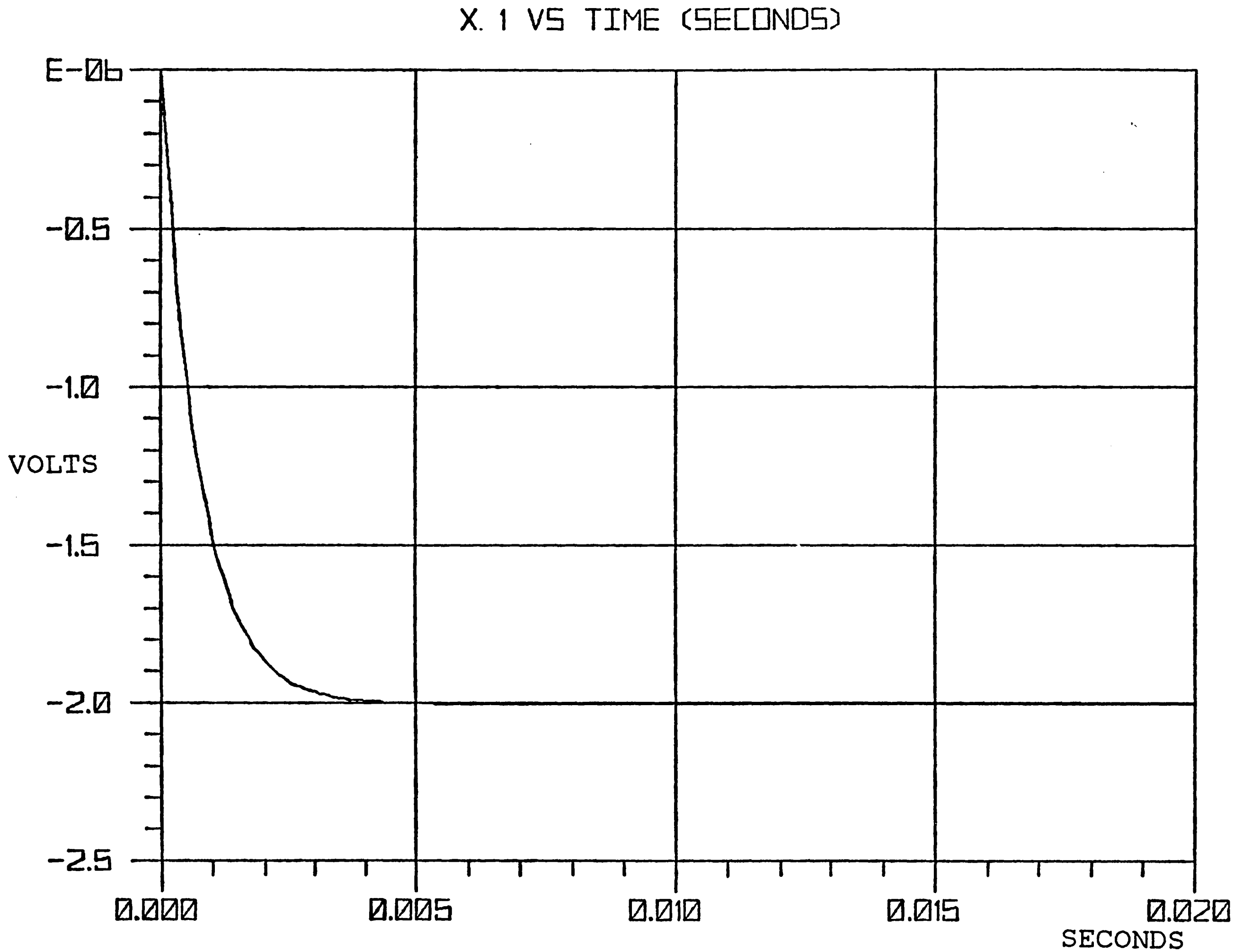


Figure 3.6 Simulated Differential Amplifier Response

3.3

RATE LOOP CARD - VELOCITY LOOP AMPLIFIER SIMULATION MODEL

The velocity loop amplifier sums the voltage from the speed scale potentiometer with the negative of the tachometer voltage. The electrical network that generates this function is shown in figure 3.7. The transfer function of this amplifier circuit was determined by nodal analysis to be:

$$V_0 = \frac{[K_1(S+T_1)(S+T_2) + K_2(S)(S+T_2) + K_3(S)(S+T_1)]\cdot V_1}{S(S+T_1)(S+T_2)} + \frac{[K_4(S+T_1) + K_5(S)(SCF)]\cdot V_2}{(S)(S+T_1)} \tag{3.3.1a}$$

This may be separated by partial fractions and expressed as:

$$V_0 = \left[\frac{K_1}{S} + \frac{K_2}{(S+T_1)} + \frac{K_3}{(S+T_2)}\right] \cdot V_1 + \left[\frac{K_4}{S} + \frac{K_5}{S+T_1}\right] \cdot (SCF) \cdot V_2 \tag{3.3.1b}$$

The values for the constants in the transfer function, in general component terms, are as folows:

$$T_1 = \frac{(C_2+C_4)}{C_2C_4C_7} \tag{3.3.2}$$

$$T_2 = \left(\frac{1}{C_1R_3}\right) \tag{3.3.3}$$

$$T_3 = \left(\frac{1}{C_2R_7} \right) \tag{3.3.4}$$

$$K_1 = \frac{-1}{R_2(C_2+C_4)} \tag{3.3.5}$$

$$K_2 = \frac{-(R_2+R_3)\left(-T_1+\frac{1}{R_7C_2}\right)\left(-T_1+\frac{1}{C_1(R_2+R_3)}\right)}{(C_4R_2R_3)(-T_1)(-T_1+T_2)} \tag{3.3.6}$$

$$K_3 = \frac{-(R_2+R_3)\left(-T_2+\frac{1}{R_7C_2}\right)\left(-T_2+\frac{1}{C_1(R_2+R_3)}\right)}{(C_4R_2R_3)(-T_2)(-T_2)(-T_2+T_1)} \tag{3.3.7}$$

$$K_4 = \frac{-T_3}{R_1C_4T_1} \tag{3.3.8}$$

$$K_5 = \frac{-(-T_1+T_3)}{R_1C_4(-T_1)} \tag{3.3.9}$$

The simulation model may now be developed from equation 3.3.1b, and it is shown in figure 3.8. It is important to note that the summing junction is connected to a nonlinear device that saturates at + or - 12.5 vdc. This simulates the actual circuit implementation of this transfer function. Because it is implemented with an op amp, the output can never reach a value of more than approximately + or - 12.5 vdc. By substituting in the values for each component in equations 3.3.2 to 3.3.9, the following values may be obtained:

$$
\begin{aligned}
T_1 &= 1408. \\
T_2 &= 303.03 \\
T_3 &= 30.30 \\
K_1 &= -238.07 \\
K_2 &= -27995.28 \\
K_3 &= 3399.70 \\
K_4 &= -1964.21 \\
K_5 &= -89329.943
\end{aligned}
\tag{3.3.10}
$$

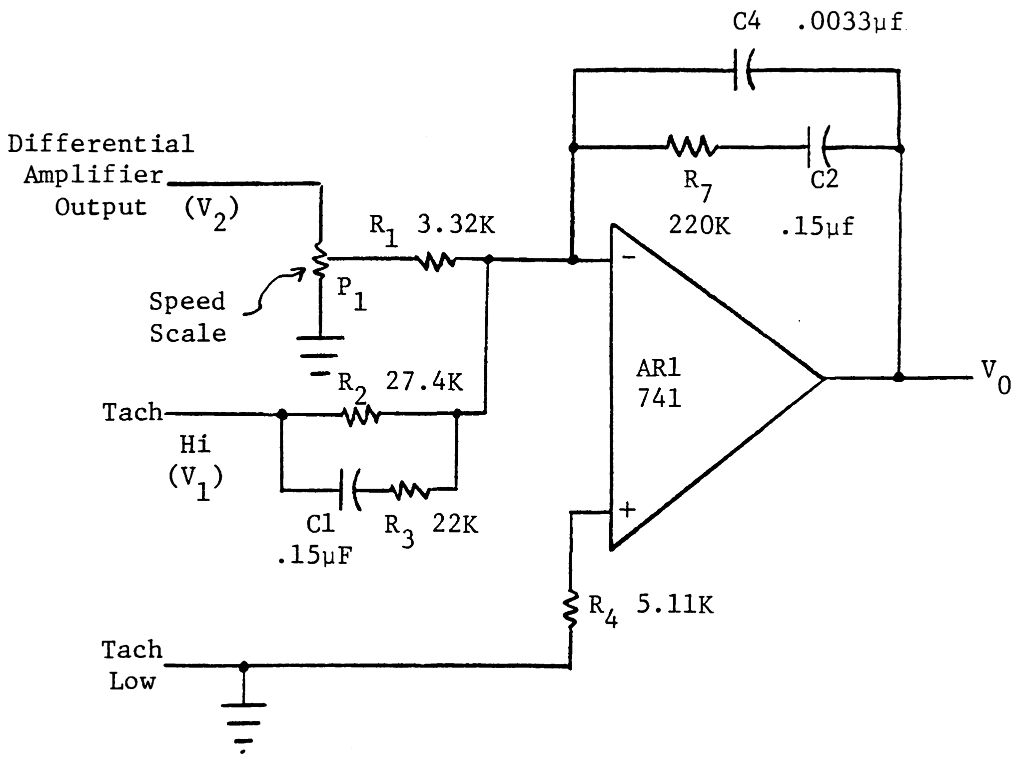


Figure 3.7 Velocity Loop Amplifier Network

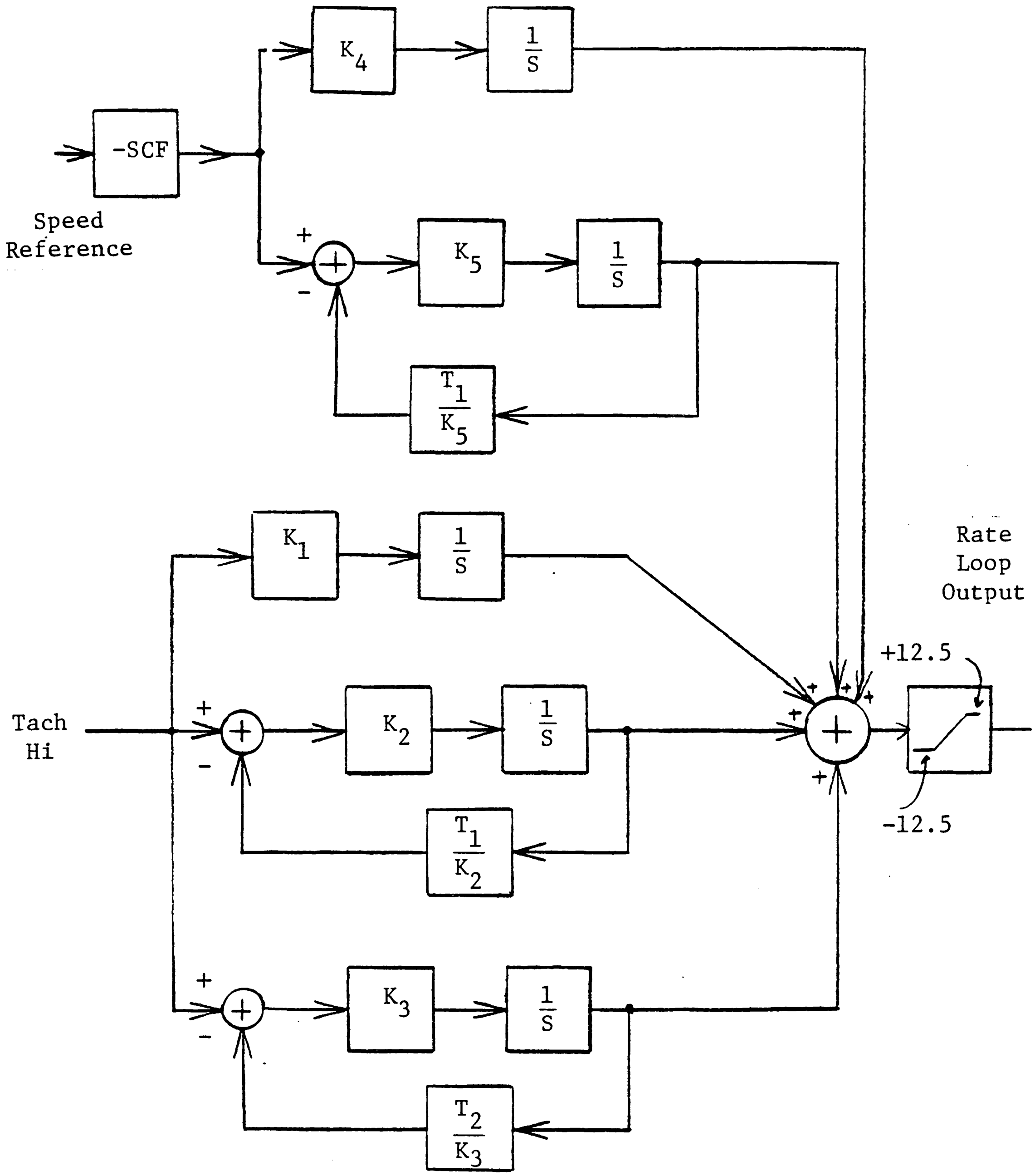


Figure 3.8 Velocity Loop Amplifier Simulation Model

From the closed loop data, the DC value of the tachometer voltage is -18.16 vdc when the reference voltage, the input of the differental amplifier, is -5.0 vdc. A simulation was run to examine the response of this network. Figure 3.9 shows that with the application of pure DC signals, the output of the amplifier saturates in less than 2 ms. This result is quite predictable when a closer look is taken at the velocity loop amplifier network (figure 3.7). This network shows that the circuit has no DC feedback because the capacitors C4 and C2 are in the feedback paths. Mathematically, this may be shown as follows. The Velocity Loop transfer function is given in equation 3.3.1a, and when the appropriate values are substituted in, the transfer function may be written as:

$$V_0(S) = \frac{-2.52\times10^4\,(S+30.3)(S+135)}{S(S+1408)(S+303.03)}\,V_1 + \frac{-91274.19(S+30.3)(-SCF)}{S(S+1408)}\,V_2 \tag{3.3.11}$$

When a -18.16 vdc step is applied as V_1 and a -5.0 vdc step is applied as V_2, the resulting response will have a K*t*u(t) term; this will make the output increase with time for constant inputs. Now, by examining an oscillogram of the tachometer voltage, figure 3.10, it may be seen that at a shaft velocity of zero rpm, the tachometer signal actually is a sinusoidal type of signal of about 0.08 mv, zero to peak, with a frequency of about 180 Hz. This signal is due to the fact that the drive system is using a 3 phase SCR bridge as the power supply. This signal's magnitude is much less than the magnitude of those signals normally observed at the tachometer output (0 to 30vdc). Therefore, this signal may easily remain unobserved unless a very sensitive oscilloscope setting is used when monitoring the tachometer voltage. To include the 180 Hz signal in this simulation, a sinusoidal generator was added in parallel with the simulated tachometer input. With only this addition, the simulated velocity loop output signal is as indicated in figure 3.11. The output of this simulation shows that the Rate Loop signal actually has an oscillatory output when there is zero error between the commanded speed and the actual motor speed (steady state). The oscillogram in figure 3.12 shows the actual Rate Loop signal output at zero error. It should be noted that this signal is not a pure sinusoid,as is shown in the simulation, because the actual tachometer signal is composed of other harmonics (figure 3.10). However, this clearly shows that the 180 Hz ripple on the tachometer voltage may not be neglected if a realistic simulation is to be generated.

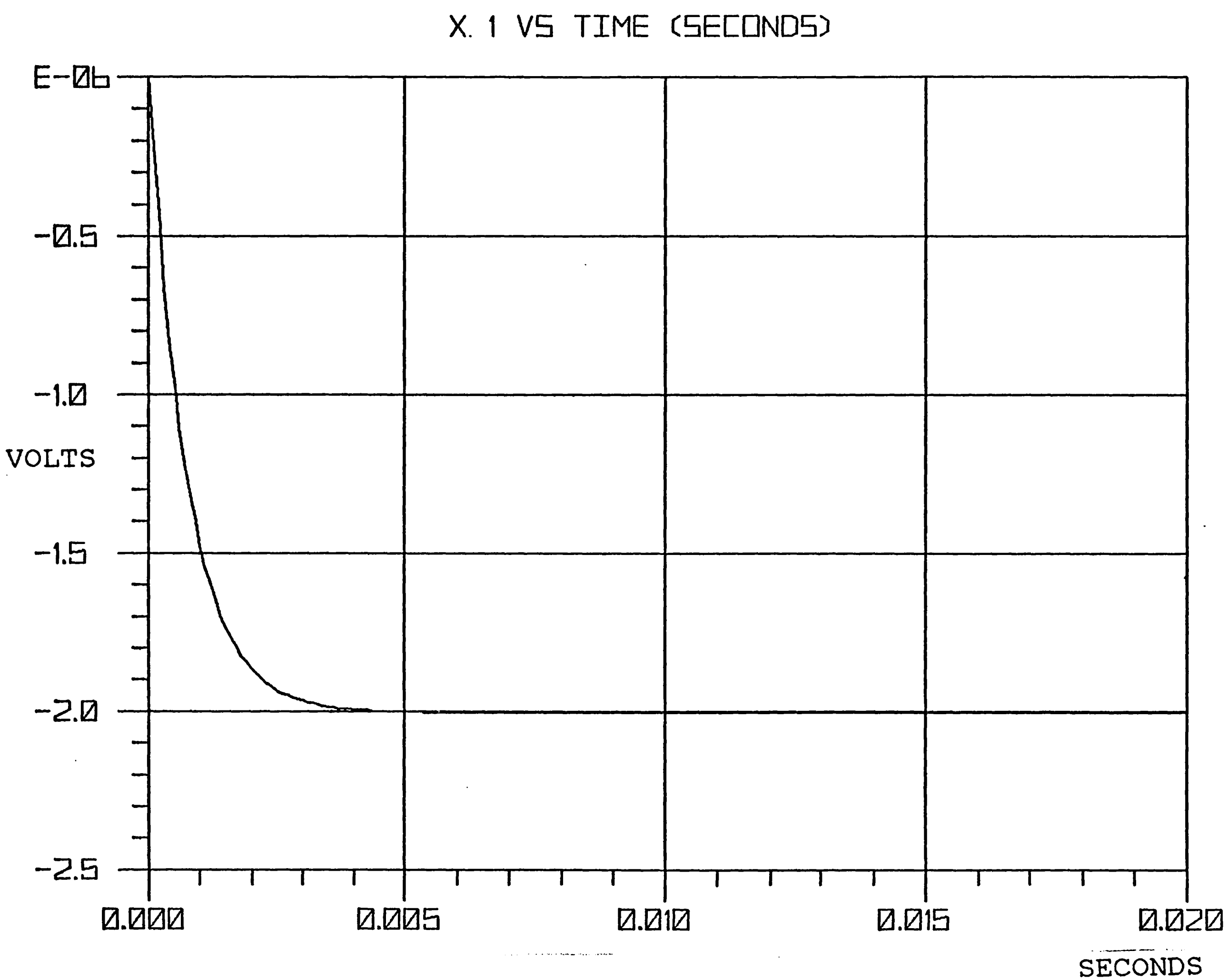
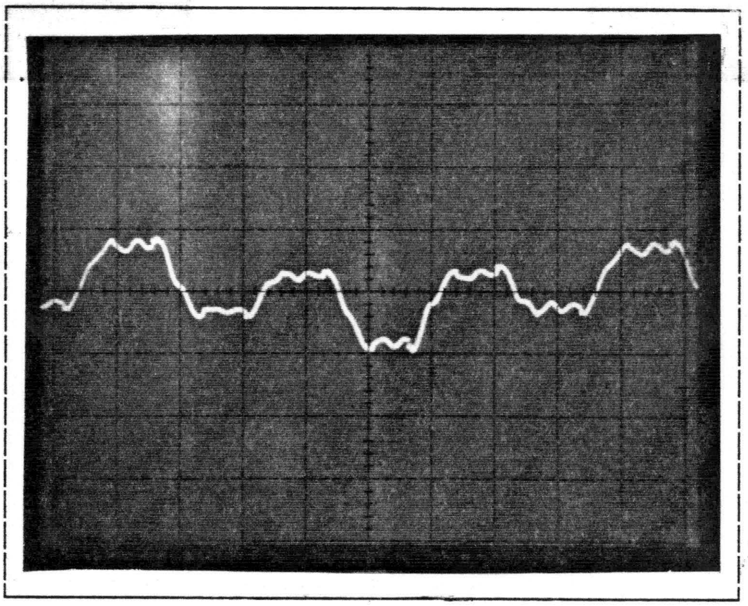


Figure 3.9 Velocity Loop Simulation Response to DC Inputs

0



VERT. = 0.1 V/DIV
HORIZ. = 2.0 ms/DIV

Figure 3.10 Actual Tachometer Voltage at Zero Speed

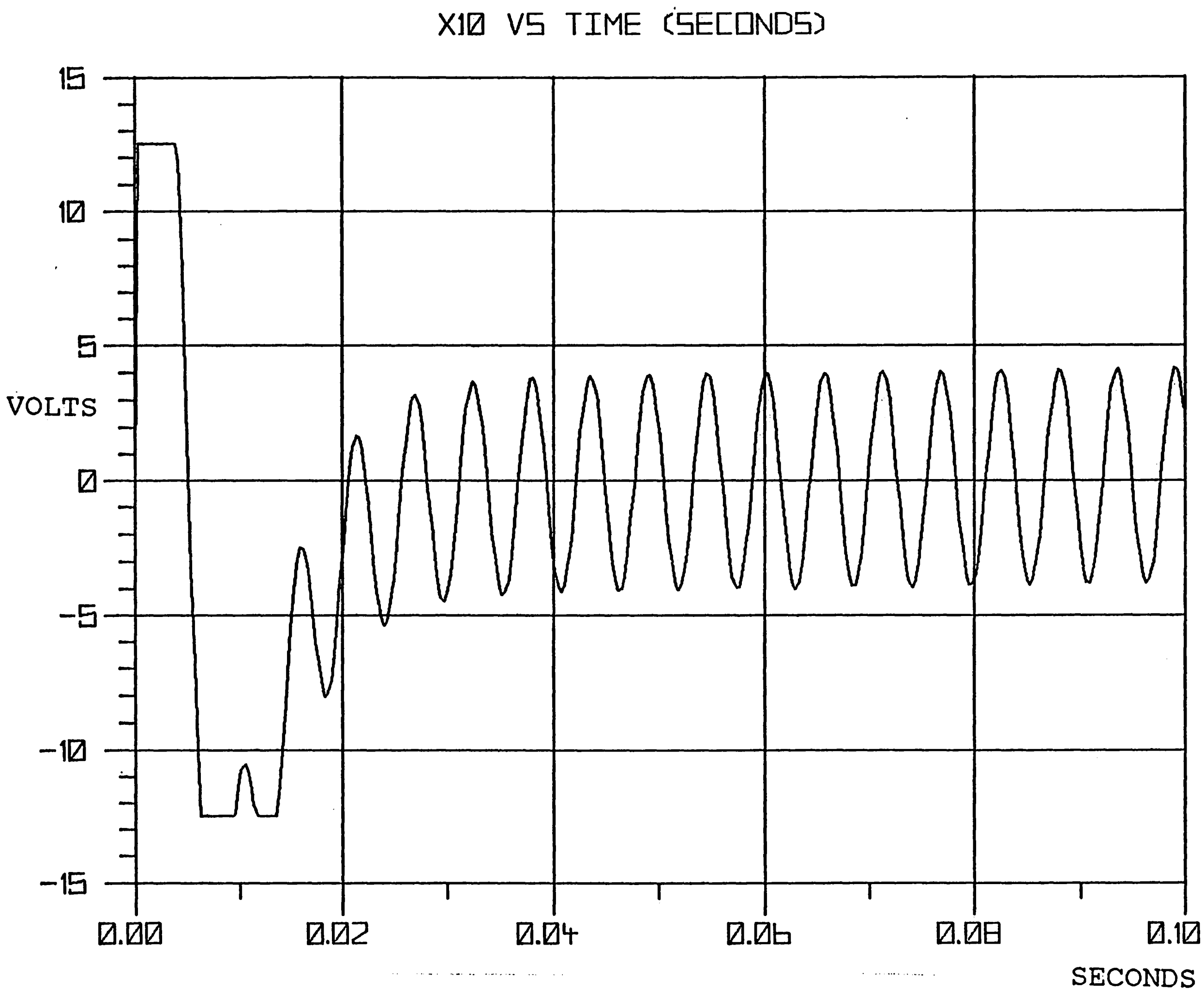
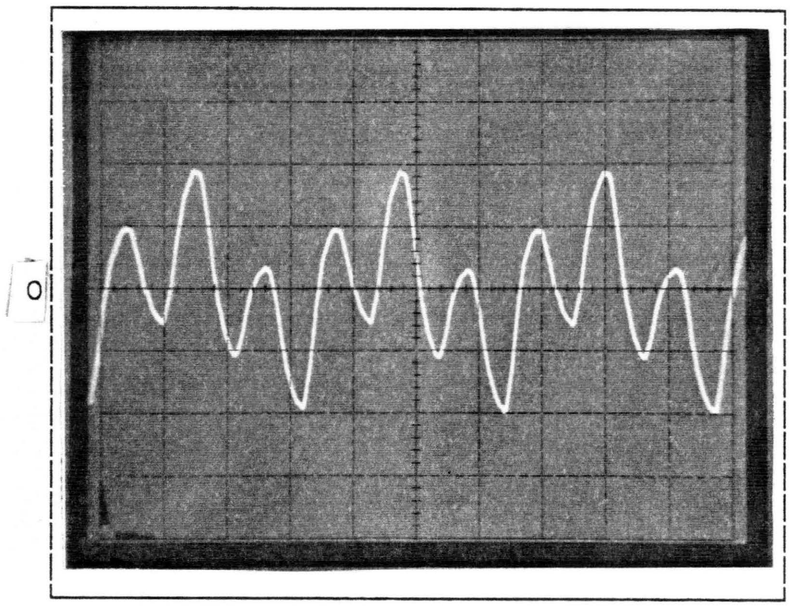


Figure 3.11 Velocity Loop Simulation Response to DC Plus Small Sinusoidal Input



VERT. = 1.0 V/DIV
HORIZ. = 5.0 ms/DIV

Figure 3.12 Actual Velocity Loop Output at Zero Error

3.4 CURRENT LOOP CARD - CURRENT STEP SIMULATION MODEL

This section will describe the development of the simulation model for the Current Loop card. The function of the Current Loop card is to create the reference voltages for the Pulse Generators by comparing the Rate Loop card output with the motor current. The Current Loop card limits the magnitude of the Rate Loop input, based on the speed, in order to extend the DC motor brush lives [21]. This task is accomplished by the current step network of figure 3.13. The AR3 op amp's inverting input is connected to 2 zener diodes (breakdown voltage of 27 vdc) that are connected back to back, which are in turn connected to the tachometer. If the tachometer voltage is less than the zener diode breakdown voltages, then the AR3 op amp's output will be zero volts; this will keep transistors Q2 and Q3 turned off, which means that the input rate error will be limited to the voltage drop across diodes cr8 and cr9 (approximately + or - 9.4 vdc). If the tachometer voltage is greater than the zener diodes' breakdown voltages, the AR3 op amp output will be at a value of plus or minus saturation (+ or - 12.5) and this value will turn Q2 or Q3 on. In this case, one of these transistors will be on and saturated, giving an approximate + or - 3.0 volt path to ground. In this case, the input rate error signal will be limitd to that 3.0 volt magnitude. In order to model this function, a new simulation device had to be designed. A flowchart for this function and its associated simulation symbol are shown in figure 3.14. This simulation device samples the tachometer signal and decides which limits to use on the input rate error at that sample; after the comparisons are made, the device returns control back to the simulation loop (see Appendix B - device 10).

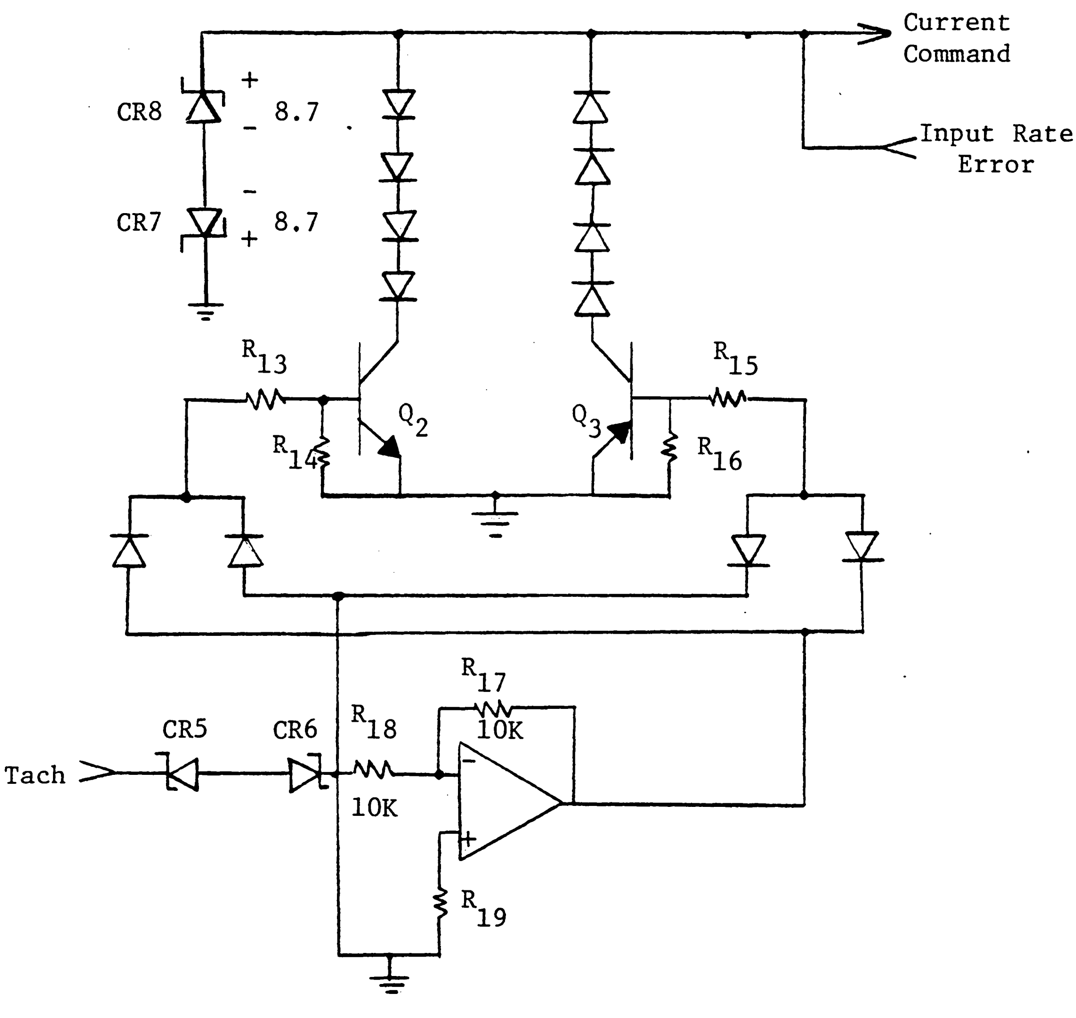
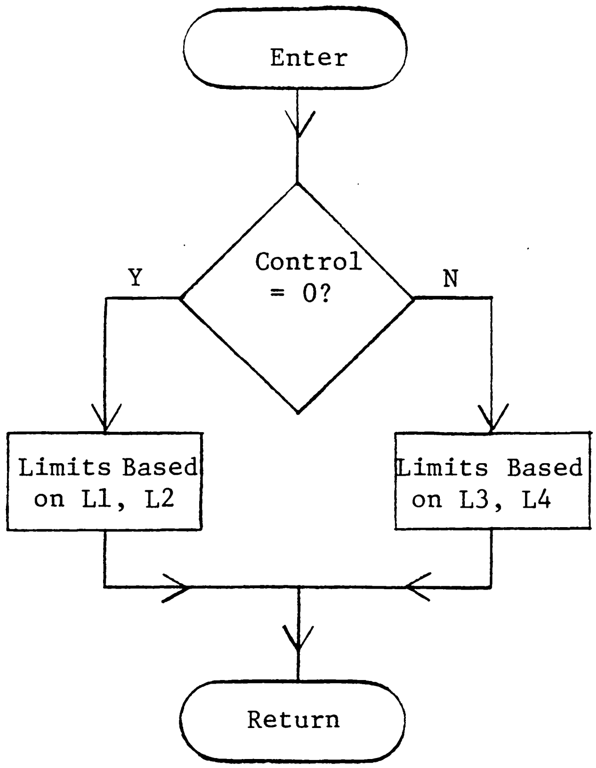


Figure 3.13 Current Step Network



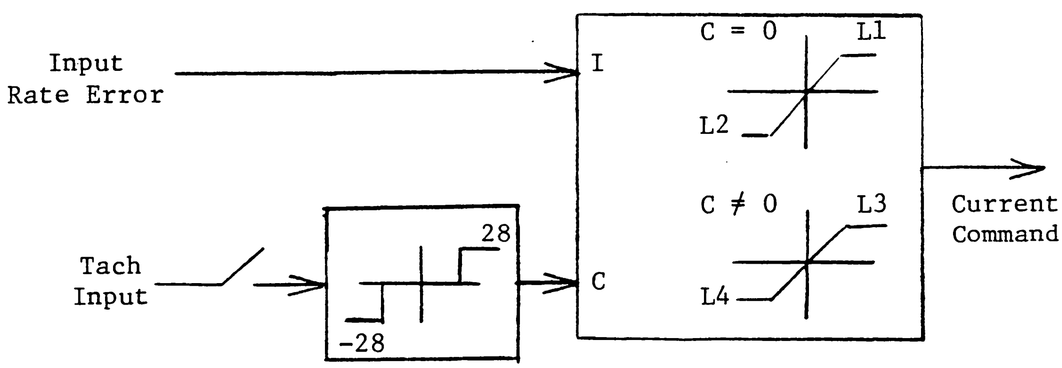


Figure 3.14 Current Step Simulation Flowchart

3.5 THE CURRENT LOOP CARD - CURRENT LOOP AMPLIFIER SIMULATION MODEL

The Current Loop amplifier, figure 3.15, compares the current command signal (the limited Rate Loop output signal) to the actual motor current. The amplified value of this input appears at the output and is diode limited. This signal is then used as a reference voltage to fire the SCRs. The transfer function for this network may be derived by applying nodal analysis at the inverting input of the amplifier. The resulting transfer function is given in equation 3.5.1.

$$V_0(S) = \frac{-[R_5C_2S+1]}{R_1C_2S}\,V_1(S) - \frac{[R_5C_2S+1]}{R_2C_2S}\,V_2(S) + \left[\frac{R_5C_2S+1}{R_1C_2S} + \frac{R_5C_2S+1}{R_1C_2S} + 1\right]V_3(S) \tag{3.5.1}$$

It should now be noticed that V_3 is the motor current low; since the motor current is obtained by placing a 0.01 ohm resistor between the low motor terminals and ground, the motor current low is equal to the power system neutral (zero volts). After setting V_3 equal to zero and factoring equation 3.5.1, the transfer function becomes equation 3.5.2 and a simulation model for this equation is shown in figure 3.16.

$$V_0(S) = \frac{-R_5}{R_1}\left[1 + \frac{1}{R_5C_2S}\right]V_1(S) - \frac{R_5}{R_2}\left[1 + \frac{1}{R_5C_2S}\right]V_2(S) \tag{3.5.2}$$

It should be clear that this Current Loop network is a Proportional - Integral controller (PI) as is common in DC drive system current loops [22]. Since the output of this loop is limited by diode cr1, the simulation model used had to have this capability. Rather than create one device to handle the PI summation and another one to handle the output limit, one device was used to handle both functions. The inputs into the device were accomplished by making the Jump matrix entries for this device equal to the appropriate constant times the respective state variable. One simulation was run in order to verify the response of the Current Loop network. For this simulation, the current command was a +1 volt step input and the motor current high (MCH) was a -1 volt step input. By using the values for the K's shown in figure 3.16, the output equation for the Current Loop becomes equation 3.5.3.

$$V_0 = -0.1 \left[1 + \frac{202.02}{S}\right]\left[\frac{1}{S}\right] - 1.83 \left[1 + \frac{202.02}{S}\right]\left[\frac{-1}{S}\right] \qquad (3.5.3)$$

After simplification, equation 3.5.3 may be written as equation 3.5.4.

$$V_0 = 1.73 \left[\frac{1}{S} + \frac{202.02}{S^2} \right] \qquad (3.5.4)$$

The time domain representation for the output equation is:

$$V_0(t) = 1.73[1 + 202.02\ t] \qquad (3.5.5)$$

A simulation of this network was run for 500 samples at a sample rate of 0.001 seconds (figure 3.17). The state variables had the following values after the 500 samples

$X_1 = -10.099$

$X_2 = 184.838$

$X_3 = 12.5$

$X_4 = +1.0$

$X_5 = -1.0$.

The simulation device (a saturating summer), X_3, output was determined from the equation 3.5.6

$$X_3 = X_1 + X_2 + K_6\ X_4 + K_8\ X_5 \qquad (3.5.6a)$$

$$X_3 = X_1 + X_2 + (-0.1)X_4 + (-1.83)X_5 \qquad (3.5.6b)$$

$$X_3 = 176.468 > 12.5 \quad \text{therefore} \quad X_3 = +12.5 \tag{3.5.6c}$$

From the time domain equation (3.4.16)

V_o (t)= 1.73 (1 + 202.02 x 0.5)= 176.477.

This value is very close to the simulated value of X_3.

This section has shown the analysis of the circuitry on the Rate Loop and Current Loop cards. It has also shown the simulation models for each of these functional blocks. It has also been shown that DC inputs to these models will eventually cause saturation unless some values cancel exactly. Thus, DC feedback signals from the tachometer voltage or motor current are unacceptable and will not suffice for a realistic simulation. Therefore, the next section on the Pulse Generators and the SCR Bridge could not be neglected.

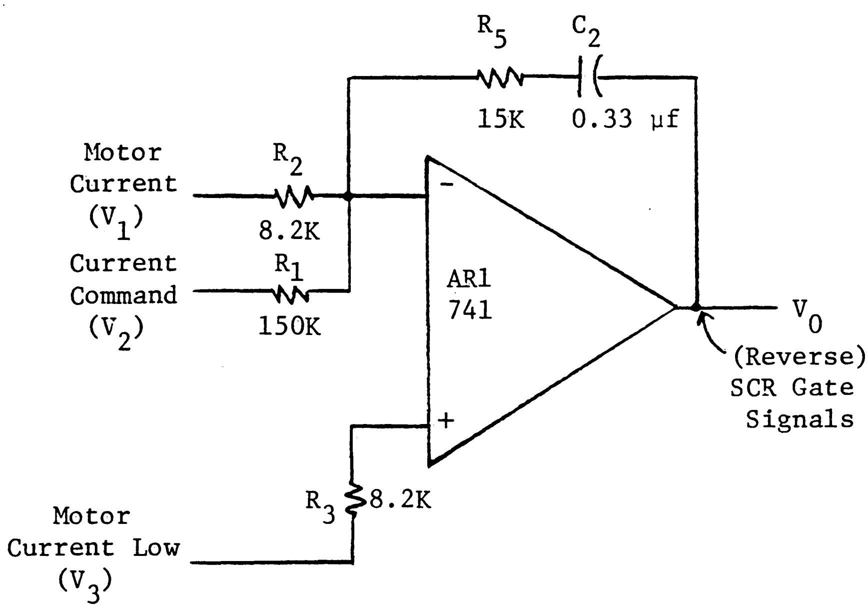


Figure 3.15 Current Loop Amplifier Network

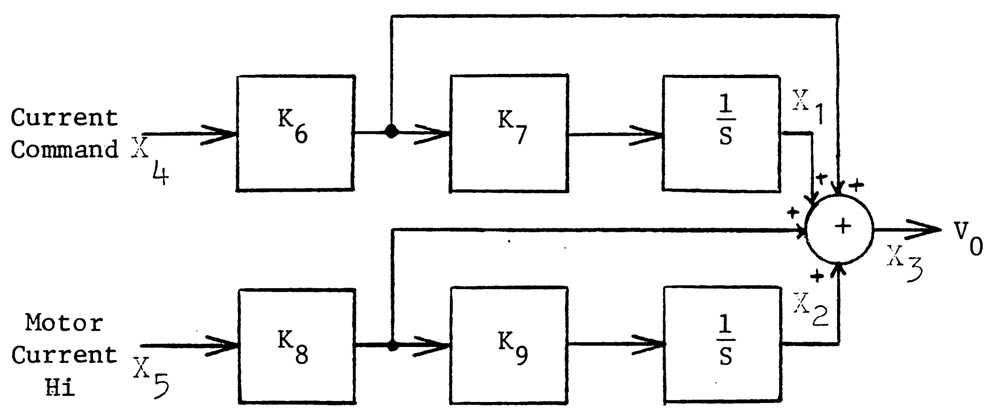


Figure 3.16 Current Loop Amplifier Simulation Model

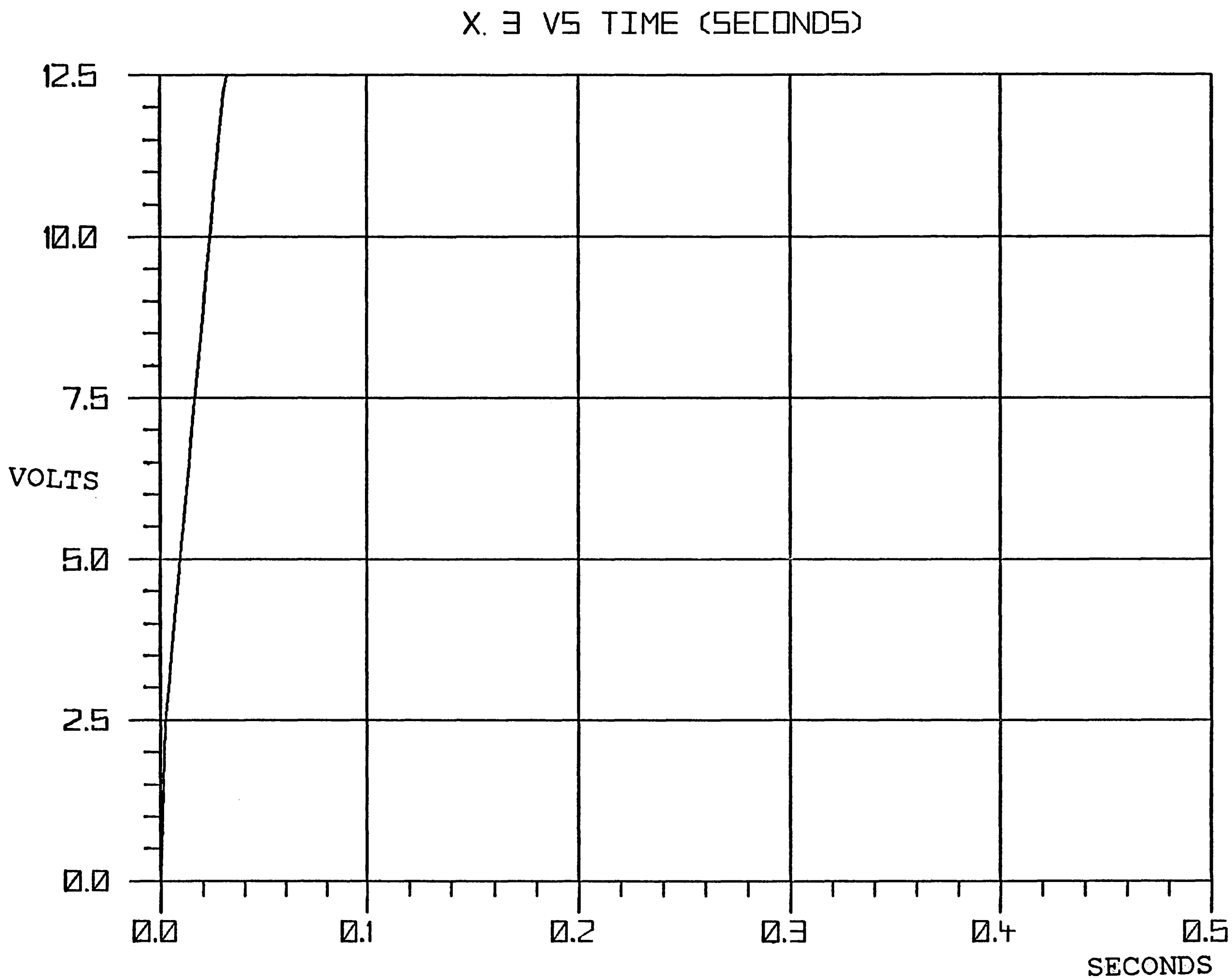


Figure 3.17 Current Loop Amplifier Simulation

(T=0.001 seconds / samples=500)

Chapter IV

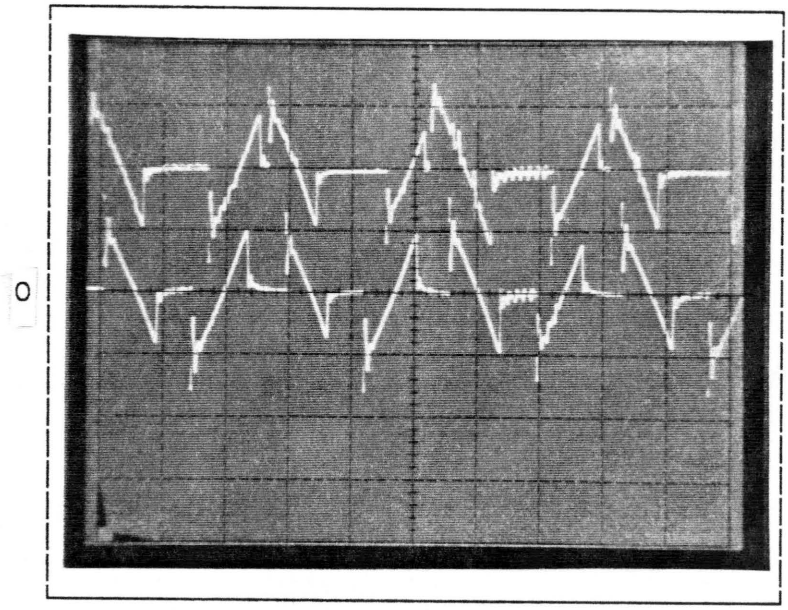
THE POWER STAGE - PULSE GENERATORS AND THE SCR BRIDGE

The power stage is composed of the Pulse Generator module and the SCR bridge (figure 2.1). The Pulse Generator module contains six similar circuits, each whose function is to generate the firing pulses for its respective SCR. The bridge is composed of six SCRs connected in a dual converter with circulating current configuration [23]. The output of the bridge is taken from the center tap of the smoothing inductor (choke) that connects the positive and negative sides of the bridge. This output supplies the voltage and current necessary to run the DC motor. An oscillogram of the output voltage of the SCR bridge when the motor is at rest, and when the motor is running at 1000 rpm [ccw] is shown in figure 4.1.

4.1 DC COMPONENT OF THE TERMINAL VOLTAGE

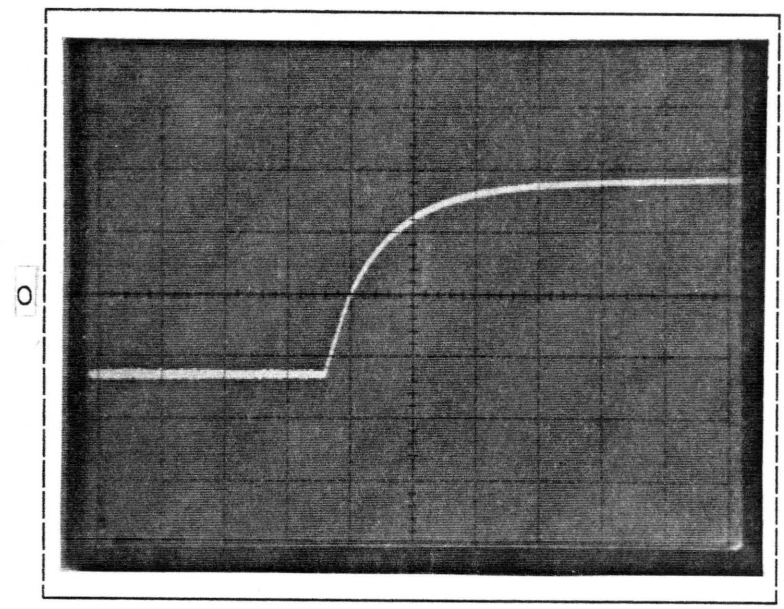
The oscillogram of figure 4.1 clearly shows the shape of the motor terminal voltage (V_t in figure 3.1). The initial step in modeling this stage is to develop a transfer function relationship between the output of the Current Loop and the terminal voltage. The work done by Nunnally[24], Sen[23], and Pelly[25] offer linear models for power stages when the appropriate SCR firing scheme is used.

Unfortunately, our system does not use this exact (cosine) firing scheme. Much of the other work that considers DC drive systems, such as Grotstollen [26], Safiuddin [3], McMurray [27], and Bonnert [22], model this stage as a first order transfer function with the time constant being related to the number of SCRs. Another model for SCR bridge systems is a transport lag representation as used by Parrish and McVey [28]. The oscillogram of figure 4.2 shows the open loop response of the DC component of the terminal voltage. This response was obtained by opening the system loop between the Current Loop output and the Pulse Generator inputs (figure 2.1). Several different magnitudes of steps were used, and each resulted in a similar response. Figure 4.2 shows the open loop response of the terminal voltage as the motor speed changes from 750 rpm [cw] to 1000 rpm [ccw]. The open loop response of this stage does approximate a first order system response and the time constant of this response was calculated to be approximately 200 ms. Next, the open loop gain from the Current Loop output to the choke needed to be calculated. Both the input signal to the Pulse Generators and the DC component of the bridge output were measured using a Fluke 8000A multimeter, therefore only DC values were observed. The data for this gain was recorded for a range of shaft velocity of 2600 rpm clockwise to 2600 rpm counterclockwise. This data is plotted in figure 4.3. The data may be approximated by the two straight line segments shown in figure 4.3. The slopes and y intercepts for these two lines were obtained by using the least squares linear regression routine on a Texas Instruments SR-51A calculator [29]. The simulation device was designed using the flowchart of figure 4.4. To simulate the results of figure 4.2, a first order system was simulated with the gain of the system being determined by the characteristic of figure 4.3. A simulation was run with this model as the power stage and the motor model as developed in section 3.1.



VERT. = 20.0 V/DIV
HORIZ. = 2.0 ms/DIV

Figure 4.1 DC Motor Terminal Voltage at 0 and 1000 RPM



VERT. = 20.0 V/DIV
HORIZ. = 200 ms/DIV

Figure 4.2 Pulse Gen. to DC Motor Terminal Voltage - Open Loop Response (750 RPM cw to 1000 RPM ccw)

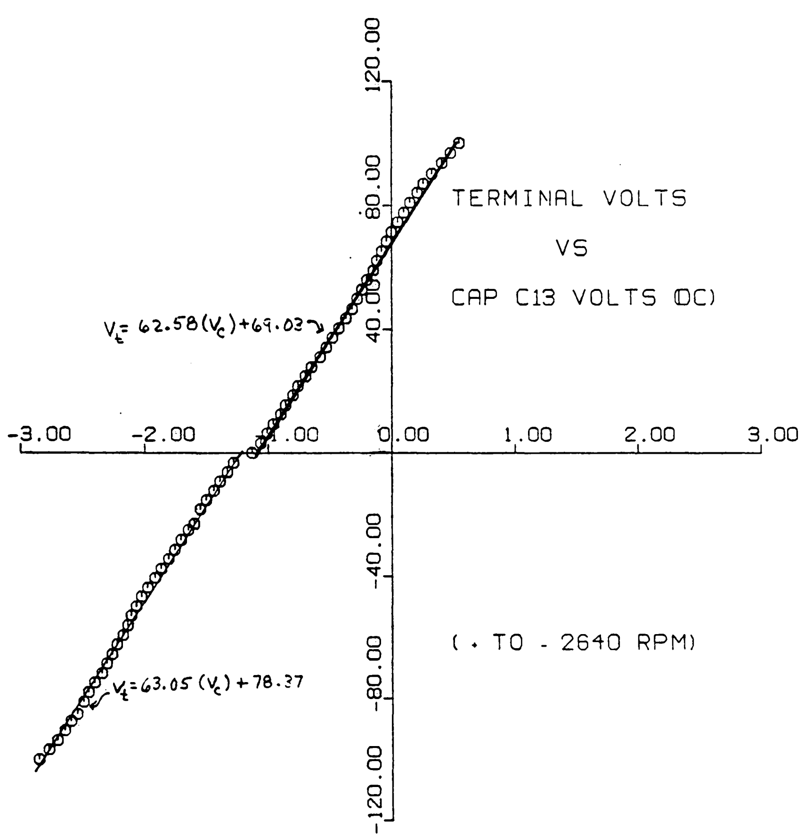


Figure 4.3 Open Loop Gain

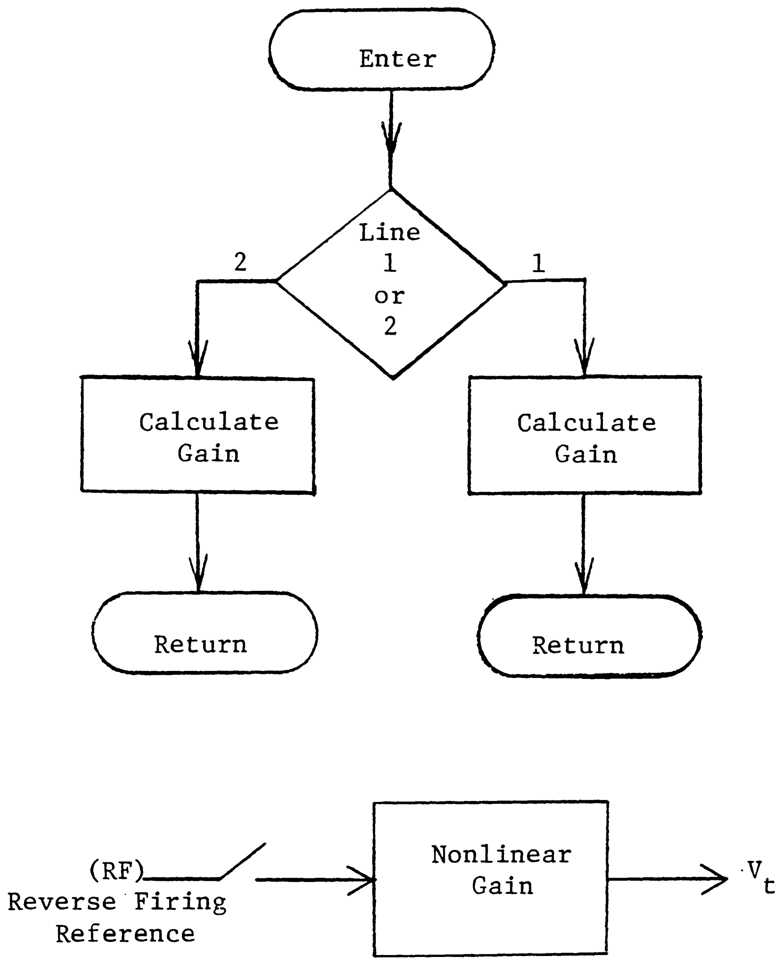


Figure 4.4 Flowchart for DC Gain Simulation Device

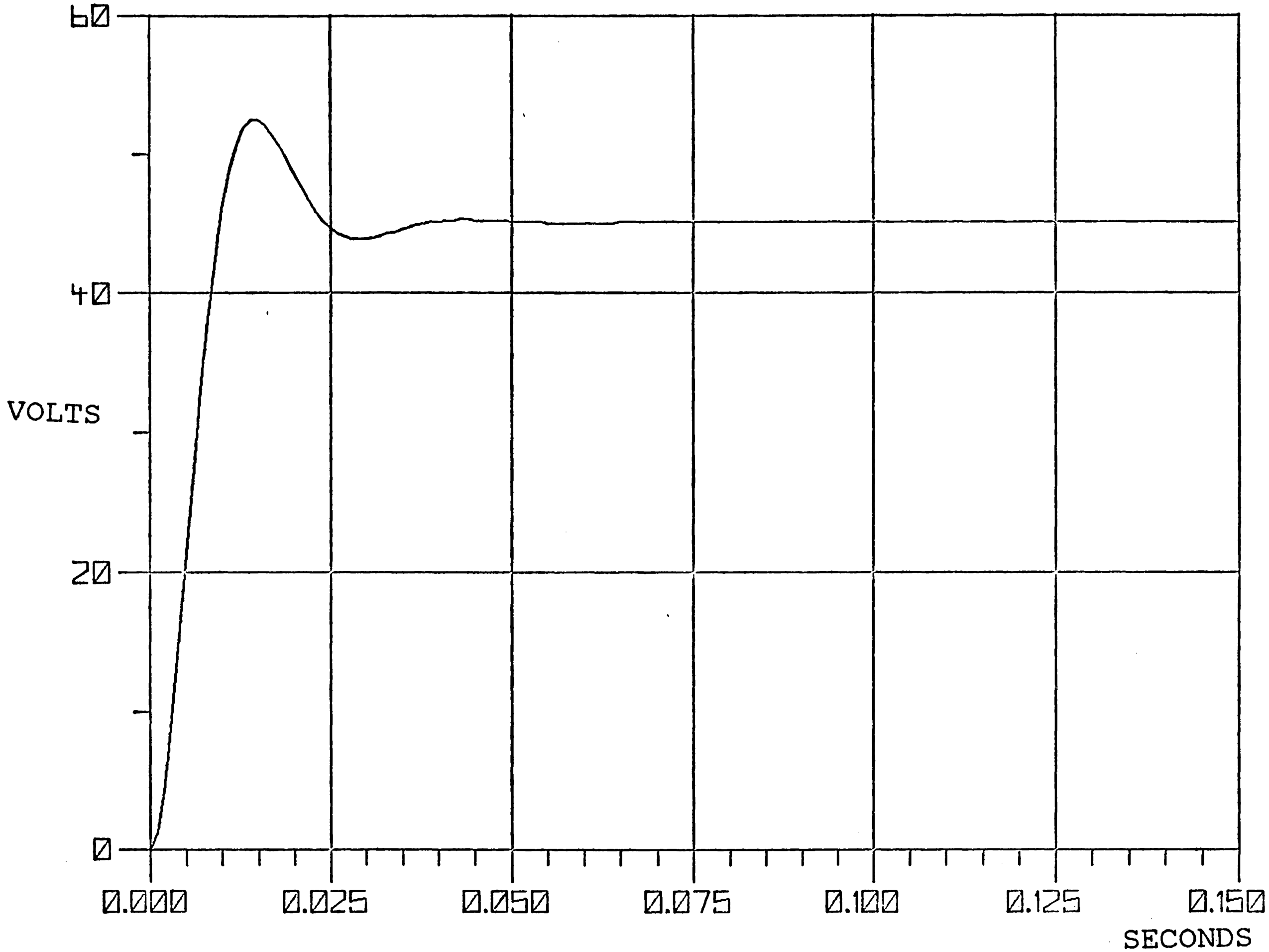


Figure 4.5a Simulated Tachometer Voltage for DC Input

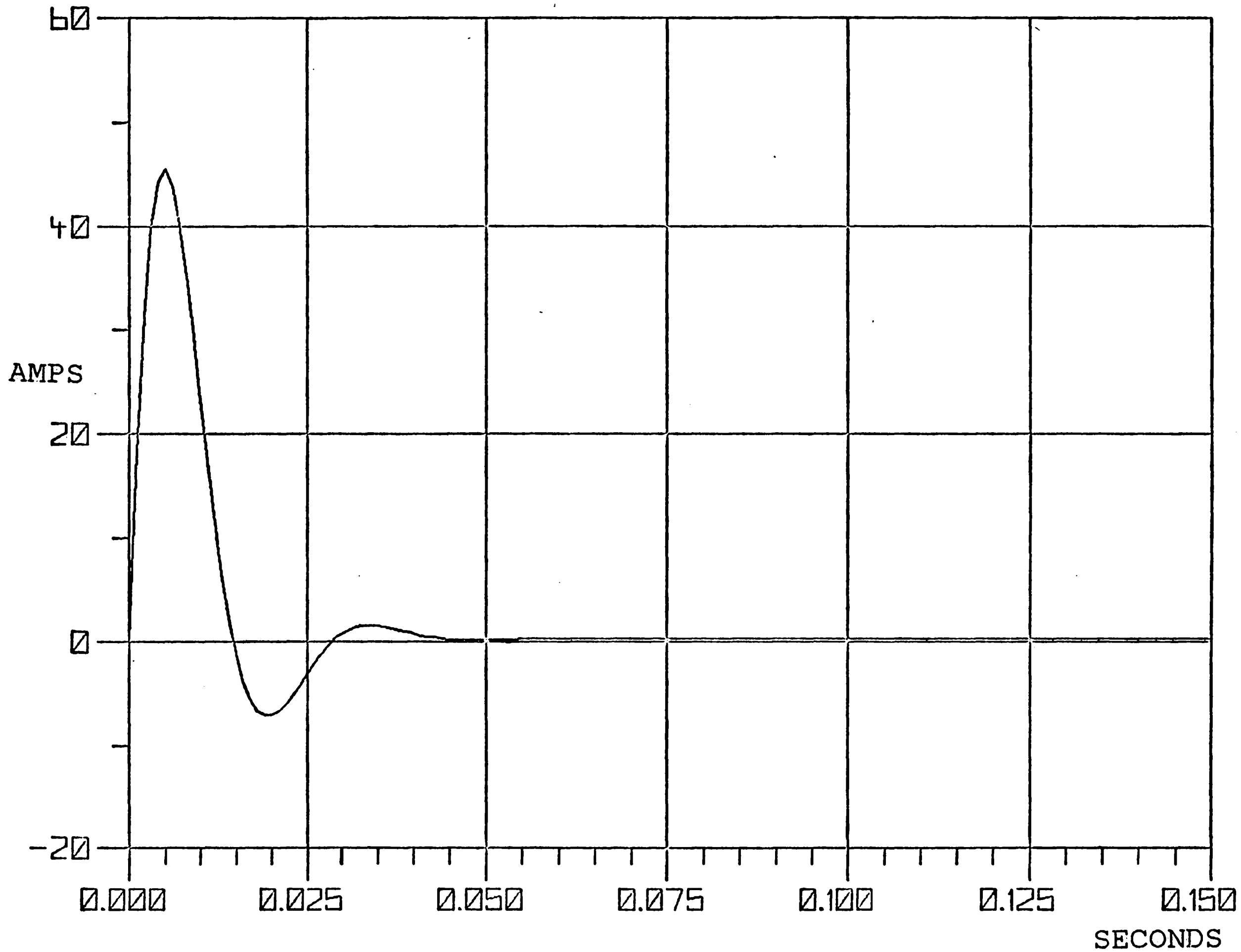


Figure 4.5b Simulated Motor Current for DC Input

The simulation shows, as expected, that all of the steady state variables have constant DC values (figures 4.5a,5b). DC steady state values were shown in section 3 to be undesirable for feedback. Oscillograms of the actual motor current and the actual tachometer voltage are shown in figures 5.6a,c. These oscillograms show that neither of these state variables have pure DC steady state values, therefore, there must be another addition to this power amplifier model.

4.2 SCR SWITCHING - SIMULATION MODEL

Section 4.1 shows that a DC model for the power stage results in motor state variables (current and velocity) that have constant DC values. The DC value of these variables are correct for this type (constant DC) of input, however, when these results are compared to the actual values in figure 5.6, the simulated results do not look very realistic. With this in mind, another simulation device was designed to produce the pulses that are shown on top of the DC value of the terminal voltage in figure 4.1.

The pulses are created by the contributions of each line phase being switched in or out by a SCR. The point in each phase voltage cycle that gets added to the DC value is determined by the output signal of the Current Loop card (section 3.4). This establishes a firing angle for that particular SCR. Figure 4.6 shows a single phase source and a SCR with a firing angle of 45 degrees [20]. The average value of the DC current can be determined as follows:

$$I_{dc} = \frac{1}{2\pi}\int_{\phi}^{\pi} I_m \sin\omega t \, d(\omega t) \qquad (4.2.1)$$

$$= \frac{I_m}{2\pi}(1 + \cos\phi) \qquad (4.2.2)$$

$$= \frac{I_m}{2\pi}(1 - \cos\theta) \qquad (4.2.3)$$

Equation 4.2.3 shows that the amount of current delivered to the source is dependent on the firing angle. The Pulse Generator modules perform the task of establishing the firing angle (figure 2.1). The Current Loop card actually generates two signals, one to fire the forward SCRs and the other to fire the negative SCRs. The forward SCRs conduct current during the positive half cycles of the phase voltage and the reverse SCRs conduct current during the negative half cycles. The signals from the Current Loop card are fed through a RC filter and are compared to a ramp, such that, when the ramp voltage exceeds this reference voltage, a pulse is generated. These pulses continue from this point until approximately 234 degrees from the beginning of the ramp. Each of the six ramps (1 for each SCR) start at approximately 6 degrees after the respective phase has crossed zero (on the way up for the forward SCRs and on the way down for the reverse SCRs). An example is shown in figure 4.7 [16].

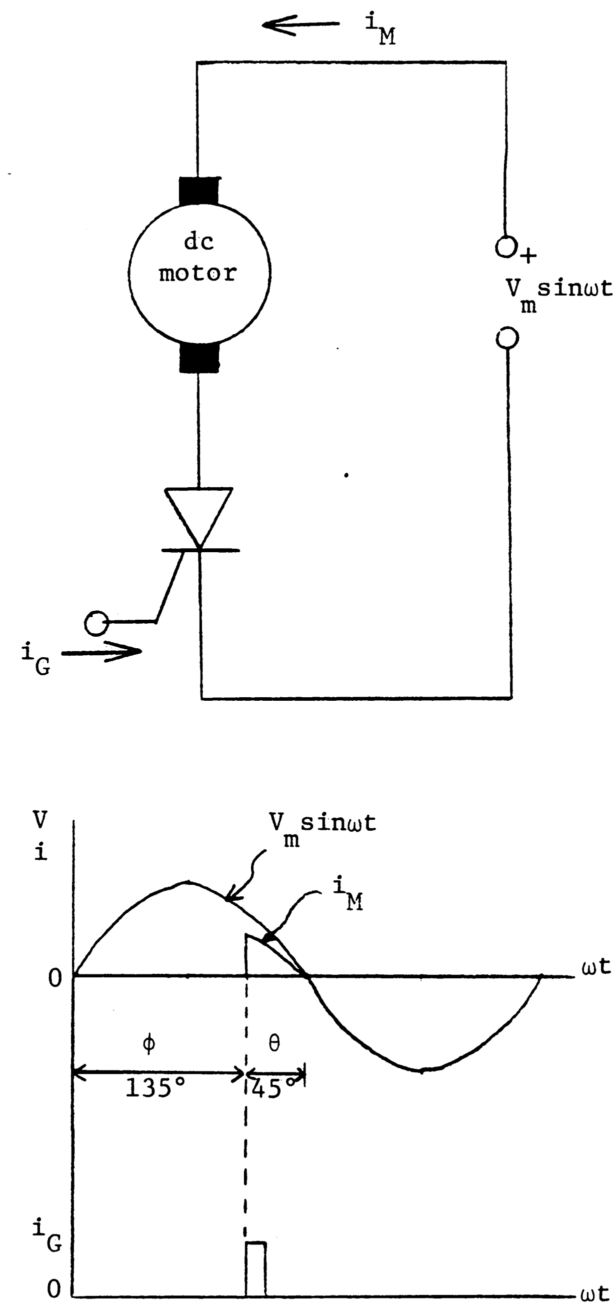


Figure 4.6 Single Phase Source and SCR

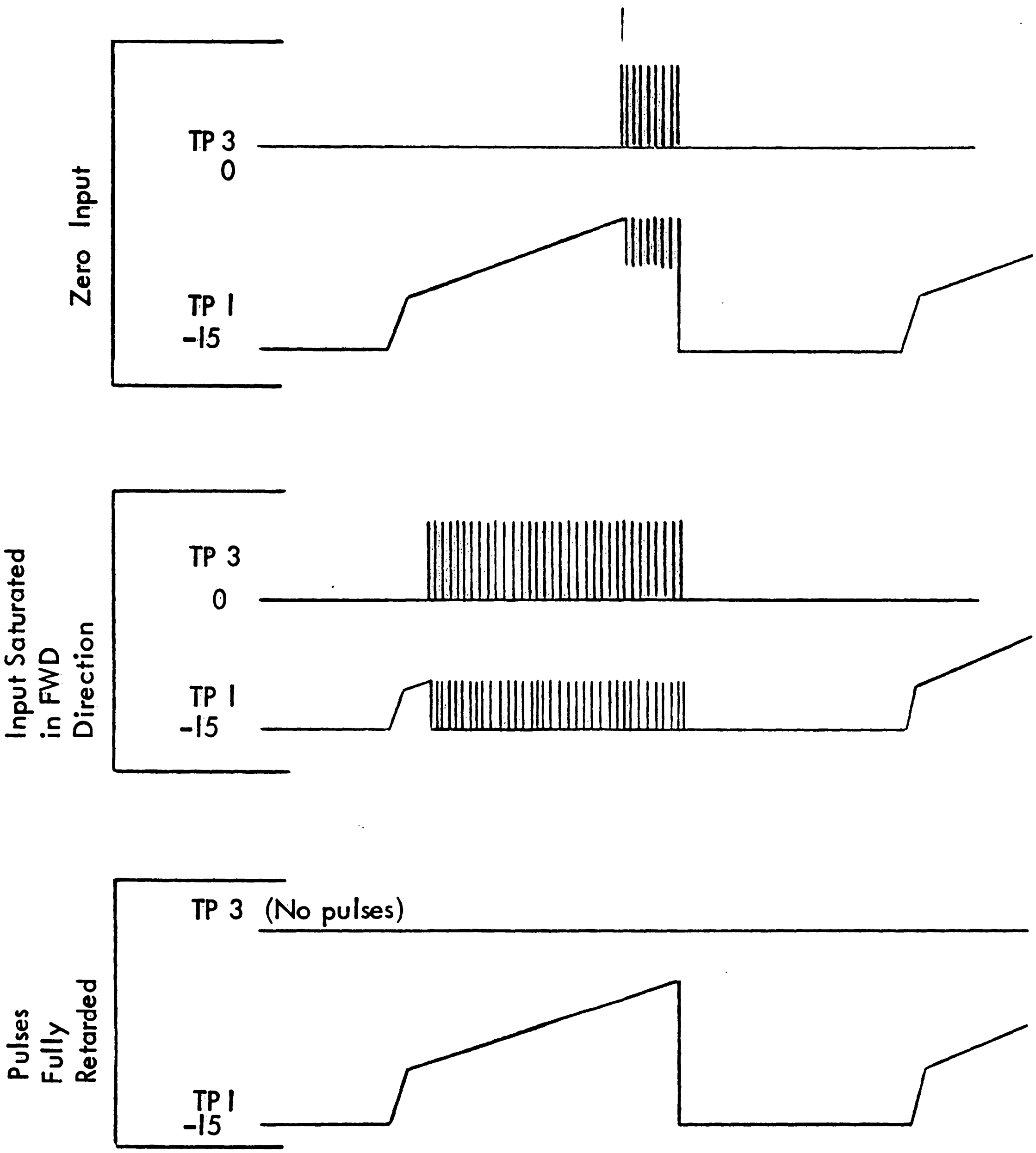


Figure 4.7 TPAR-3330 Firing Strategy [16]

An ideal switching concept was used in order to simulate the action of the SCR bridge. This concept assumes that the load on the SCR bridge is purely a resistive one; in this case, the voltage and the current in each SCR will be in phase. Therefore, the SCR will stop conducting when the phase voltage crosses zero, because the current is crossing zero at the same instant and the SCRs are current controlled devices. An example of this concept is shown in figures 4.8a,b,c. Figure 4.8a shows the circuit model of the SCR bridge and the DC motor as a load and figure 4.8b shows the balanced 3 phase supply. For this discussion, a phase angle of 30 degrees will be used. When a phase angle of 30 degrees is used, each SCR will begin conduction at the point indicated by the arrows in figure 4.8b. When all of these outputs are summed together through the 2.5 mH circulating current choke, the output is of the shape as indicated in figure 4.8c. This is the signal that needs to be summed with the DC component of the terminal voltage that was obtained in section 4.1.

The next step in the modeling process was to develop a method of simulation that performs the procedure that was described in the preceding section. The following functions were needed:

1) a balanced 3 phase voltage supply

2) 6 ramps for SCR firing

3) forward firing pulses from the reverse signal

4) bias current setting

5) all pulses must have a 240 degree range

6) represent the choke

The number of simulation devices would be greatly increased if each of the above mentioned functions were simulated separately. Therefore, one simulation device was designed to include all of the required functions. Figure 4.9 shows a flowchart of the simulation device.

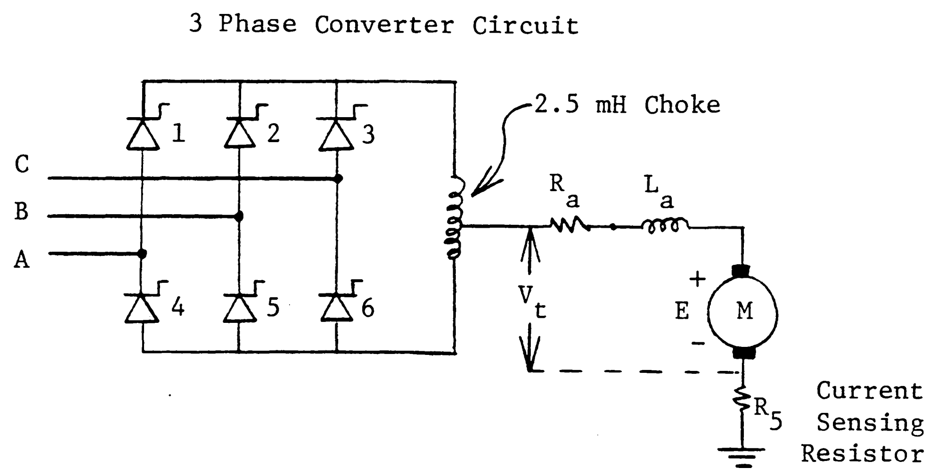


Figure 4.8a Circuit Model of SCR Bridge and DC Motor

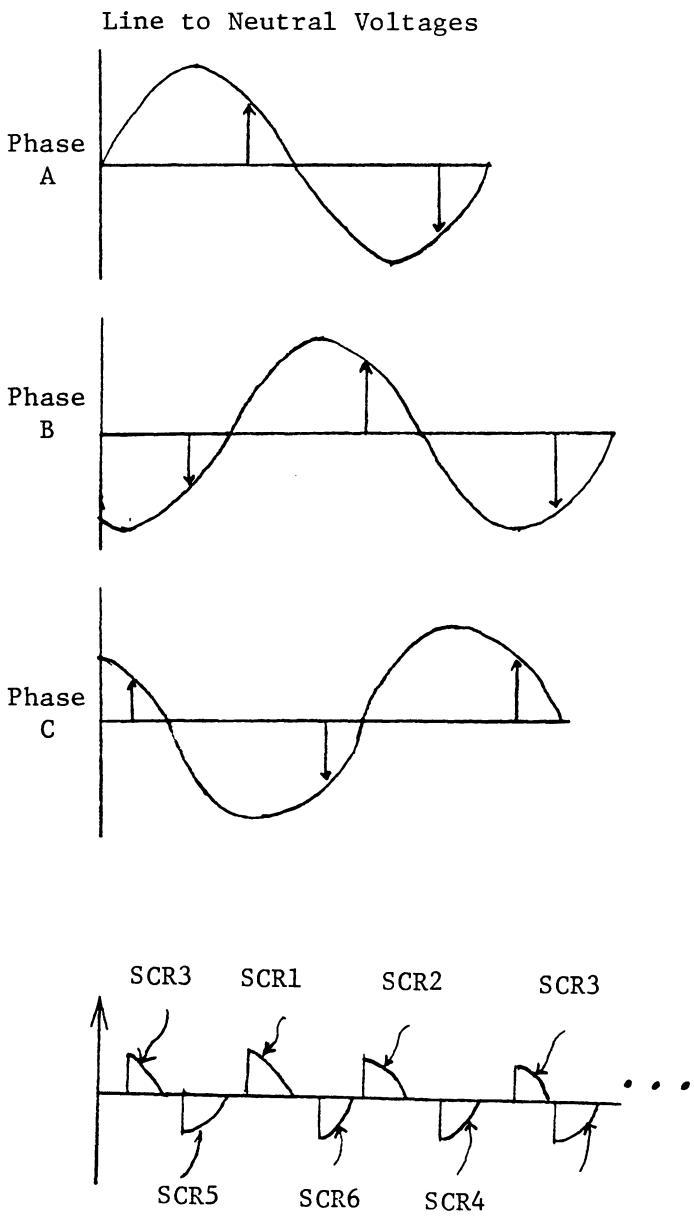


Figure 4.8b,c SCR Switching

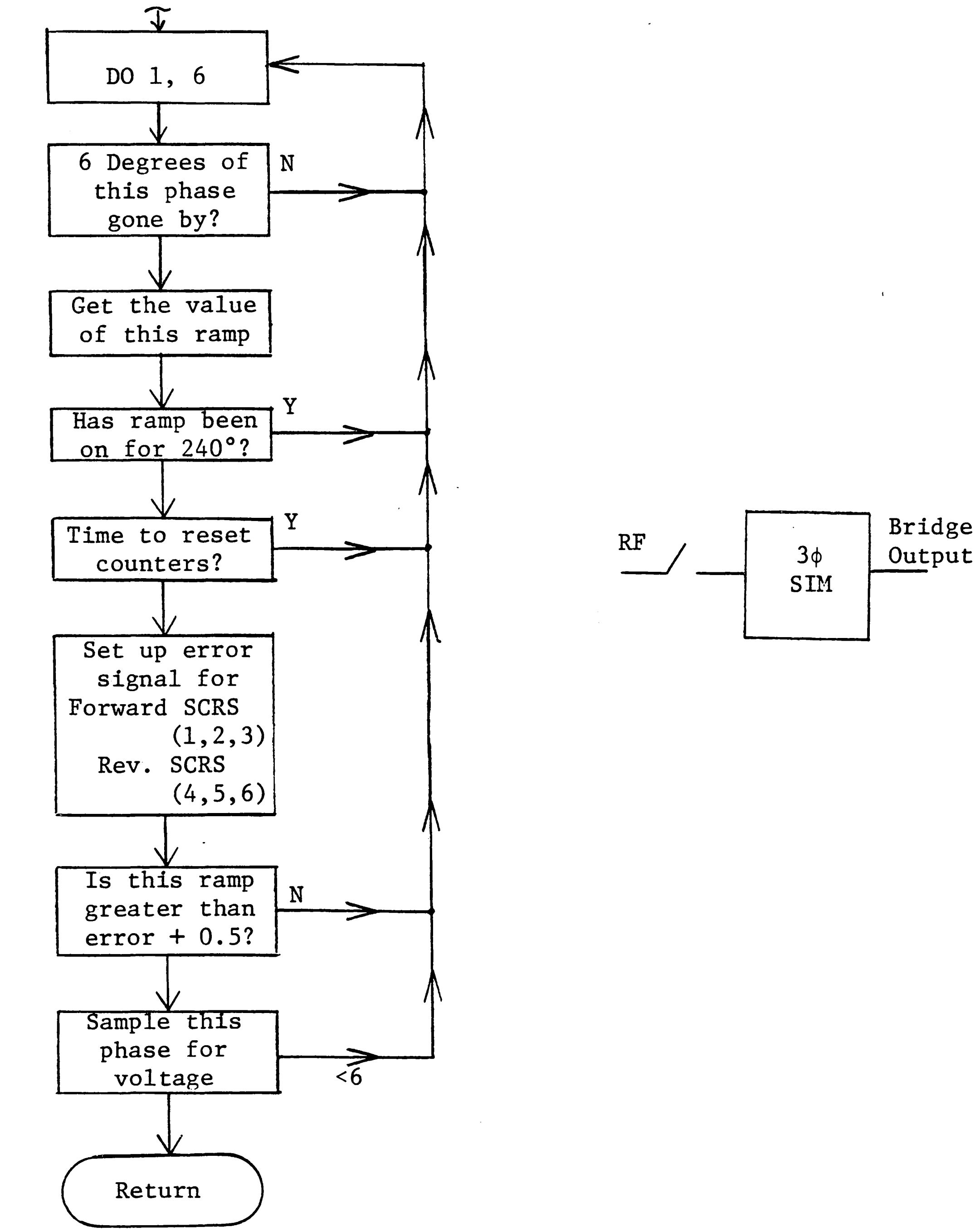


Figure 4.9 Flowchart of a Device to Simulate the SCR Bridge

The actual Pulse Generator circuits have a six degree lagging phase shift with respect to the line crossover. Therefore, the first module in the simulation device develops a six degree phase shift for each ramp, before the ramp may begin to rise. An oscillogram of a ramp signal is shown in figure 4.10a, while a simulated ramp is shown in figure 4.10b. Note that the actual ramp signal has several pulses on it during each cycle, the extra pulses after the first one are to assure that the SCR gate gets a pulse if it missed the first one. In the simulation, the pulse stream was not needed because the software in the simulated SCR device will start to conduct when the ramp crosses the forward or reverse gate signal. Once any SCR is on, in the simulation, it remains on until the input phase to that SCR crosses zero. The forward firing signal also has to be created within this device because the only output from the Current Loop card simulation model is the reverse firing signal. The forward firing signal in the actual drive system is created by the circuit of figure 4.11. The transfer function may easily be determined from the potentiometer setting and the resistor values in this circuit.

The transfer function is determined to be:

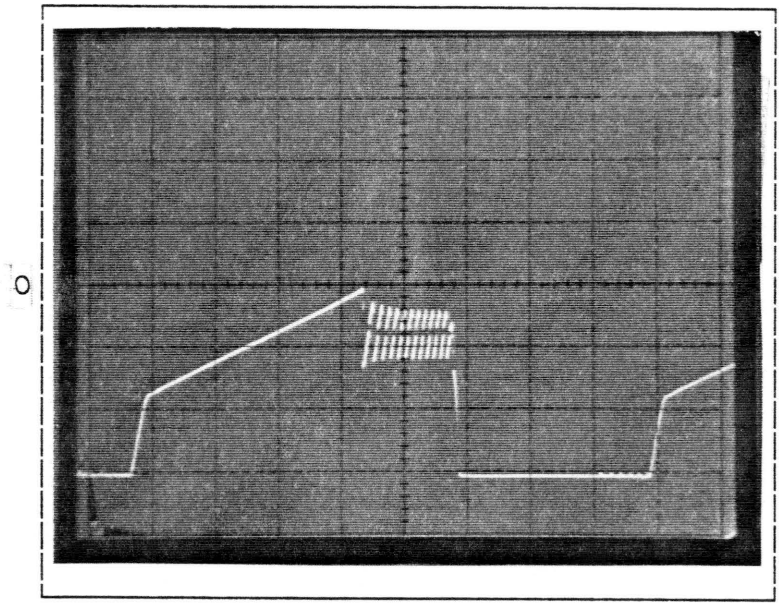
$$V_{FG}(t) = -2.5 - X_{RG}(t) \tag{4.2.4}$$

This equation is easily created within the simulation device because $X_{RG}(t)$ is the input. At each sample time, each of the six ramps are compared to their respective firing signal. If the ramp is greater than the firing signal, the voltage of that SCR is given the value of the phase voltage. The phase voltages in the simulation device assume that the 3 phase voltage applied to the SCRs, figure 4.8a, is the output of a balanced 3 phase ideal voltage source. Therefore, all of the line to neutral voltages may be described by the following equations:

$$\begin{aligned} \text{Phase A} &= 275*\sin\omega t \\ \text{Phase B} &= 275*\sin(\omega t-120°) \\ \text{Phase C} &= 275*\sin(\omega t+120°) \end{aligned} \tag{4.2.5}$$

By using this fact, the ideal supply voltages are easily calculated at any sample instant t. The output of the simulated ideal supply is shown in figure 4.12. Now the output of the simulated bridge may be calculated. An example of the simulation device output is shown in figure

4.13; the input reverse firing signal is equal to -1.13 volts. This is the desired output of the device, because -1.13 is the DC reverse firing signal that is measured in the actual system when the motor speed is equal to zero.



VERT. = 5.0 V/DIV
HORIZ. = 2.0 ms/DIV

Figure 4.10a Actual Firing Ramps

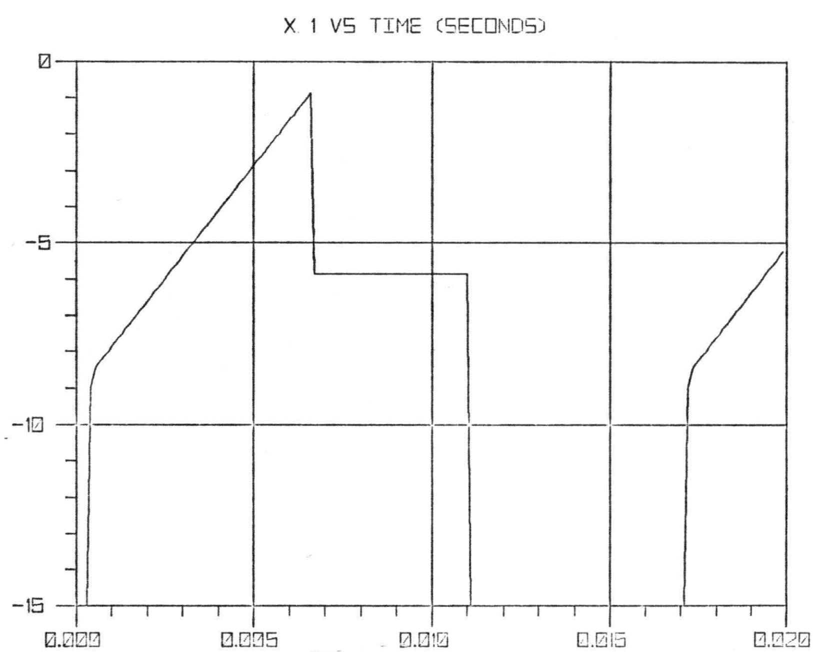


Figure 4.10b Simulated Ramps

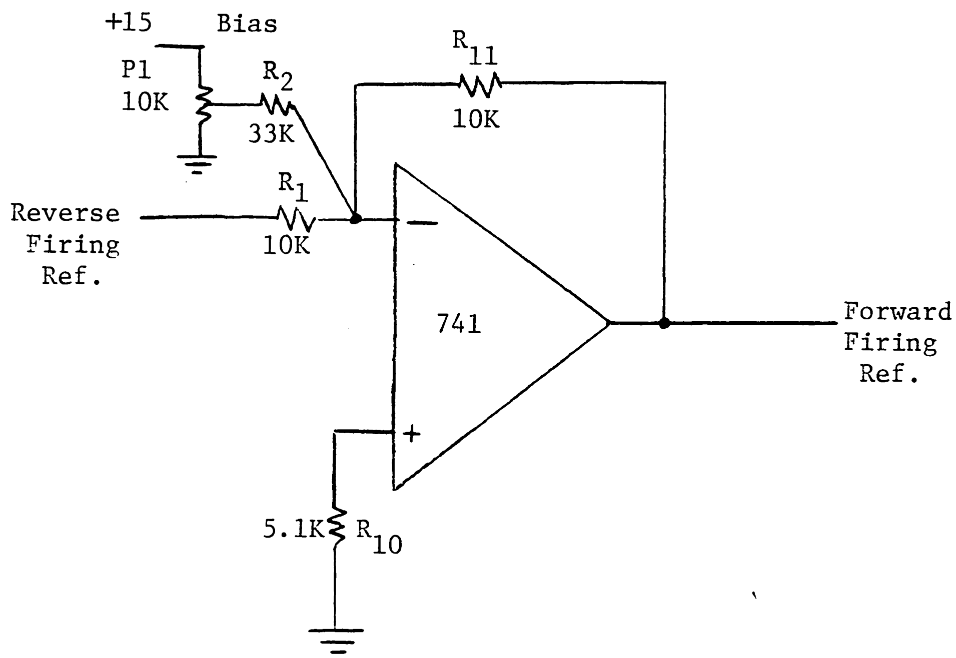


Figure 4.11 Forward SCR Firing Circuit

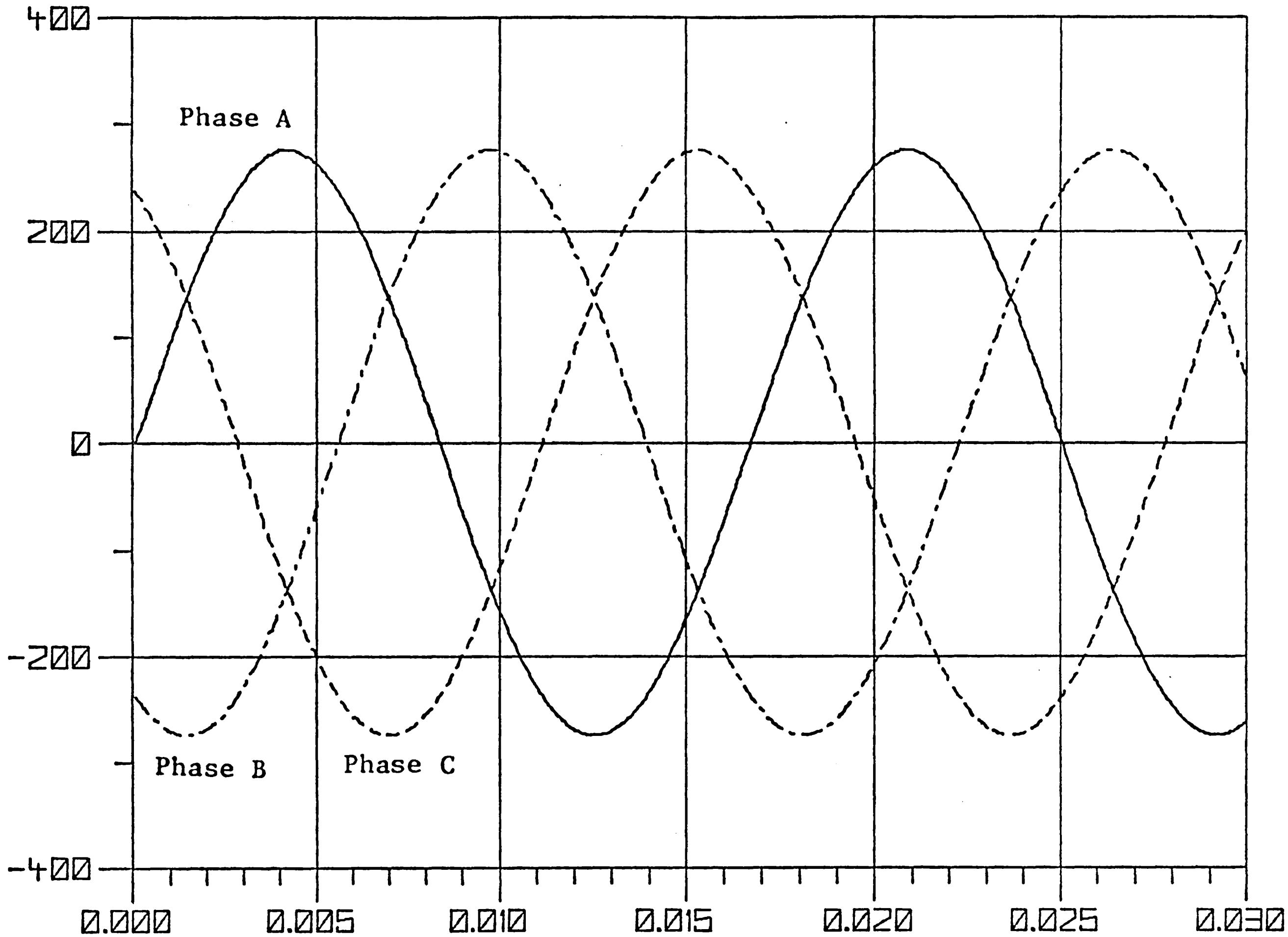


Figure 4.12 Output of Simulated Ideal Supply

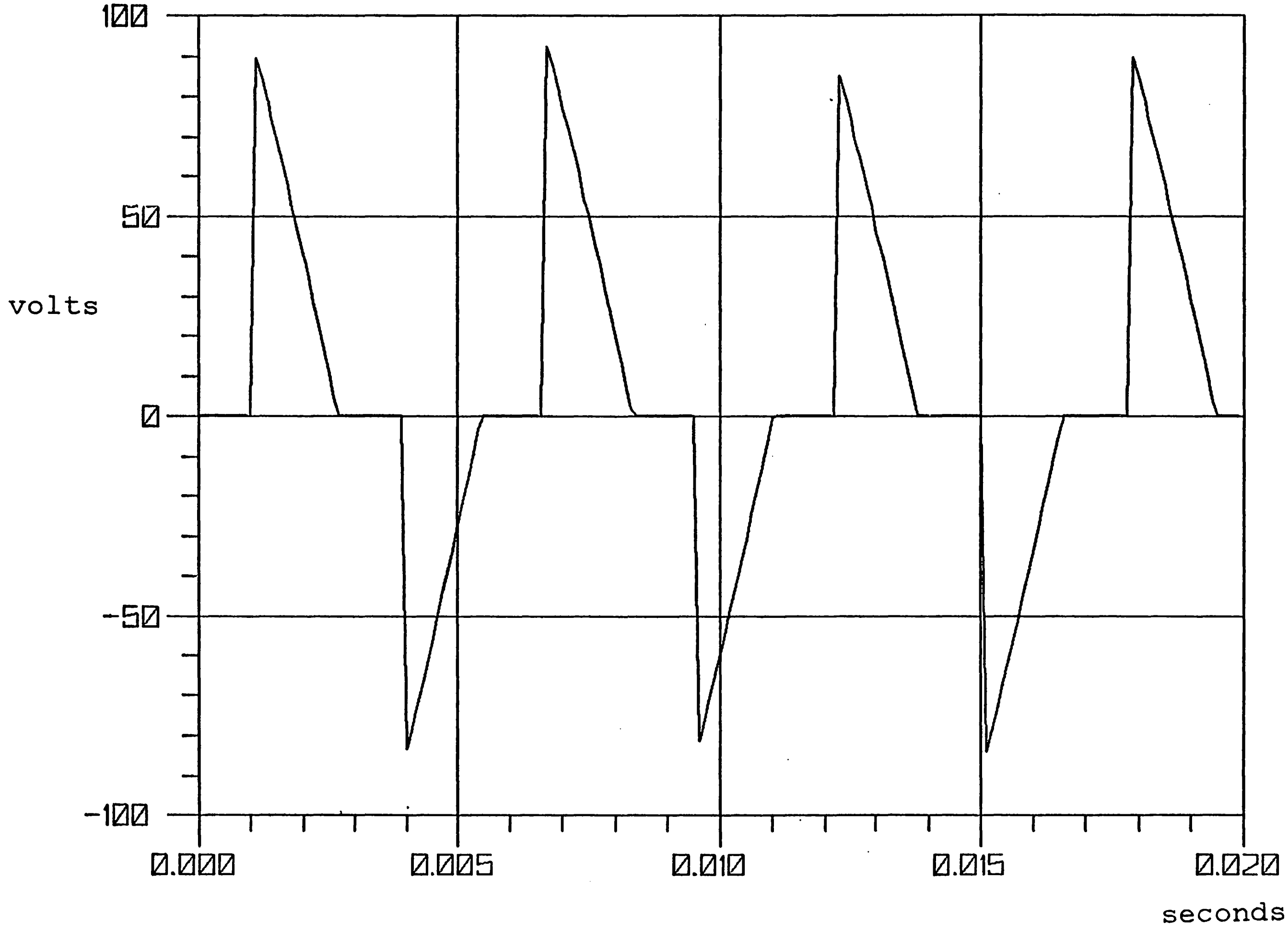


Figure 4.13 Simulated Bridge Output

As noted earlier in this section, the output of the bridge is summed through a smoothing inductor (choke). Since the choke is an inductor, its dynamics must be included in the model. The SCR bridge and choke, figure 4.8a, may be redrawn as in figure 4.14; the fact that the output is taken from the center tap of the choke to ground accounts for the resistance and inductance values being equal to 1/2 R_c (resistance of the choke) and 1/2 L_c (inductance of the choke) respectively. Looking into the circuit at terminals M and G, the Thevenin impedance and Thevenin voltage is:

$$Z_{Th} = \left(\frac{1}{2} r_c + \frac{1}{2} L_c S\right) // \left(\frac{1}{2} r_c + \frac{1}{2} L_c S\right) \qquad (4.2.6)$$

$$Z_{Th} = \frac{1}{4} (r_c + SL_c) \qquad (4.2.7)$$

$$V_{Th} = (V_2 - V_1) \times \frac{Z_1}{Z_1 + Z_2} + V_1 \qquad (4.2.8)$$

but $Z_2 = Z_1$

$$V_{Th} = (V_2 - V_1) \times \frac{Z_1}{2Z_1} + V_1 \qquad (4.2.9)$$

$$V_{Th} = (V_2 - V_1) \frac{1}{2} + V_1 \qquad (4.2.10)$$

$$V_{Th} = \frac{1}{2}(V_1+V_2) \tag{4.2.11}$$

Therefore, the circuit of figure 4.14 may be redrawn with the use of Thevenin's theorem (figure 4.15). Analysis of figure 4.16 shows that the Thevenin impedance looking into the choke may be added in with the series elements of the DC motor model.

The Current Loop output signals RF and FF are input into a RC filter before the ramp voltage is compared. The output of this network is connected to the gate of a unijunction transistor to generate the firing pulses. The RC network is shown in figure 4.16. The current in the filter loop is given by the equation:

$$I(S) = \frac{CS(V_{in}-V_R)}{RCS+1} \tag{4.2.12}$$

The output voltage, V_0, is given by:

$$V_0 = V_C + V_R \tag{4.2.13}$$

$$V_0(S) = \frac{V_{in}}{RCS+1} + \frac{-V_R}{RCS+1} + V_R \tag{4.2.14}$$

Now, by substituting in the values for each component, the following equation is derived:

$$V_0(S) = \frac{1000}{S+1000} V_{in} + \frac{15000}{S(S+1000)} - \frac{15}{S} \qquad (4.2.15)$$

This equation may be simplified to the form of equation 4.2.16:

$$V_0(S) = \frac{1000}{S+1000} V_{in} - \frac{15}{S+1000} \qquad (4.2.16)$$

The second term of equation 4.2.16 is not dependent on the input signal, $V_{in}(s)$, and thus is only a transient term. Since this term has a time constant of 1 ms, it was neglected in the model. This first order device must be placed between the Current Loop output, RF, and the bridge simulation device.

This section has shown the development of a device to simulate the action of the Pulse Generators and the SCR bridge. However, this simulation does not account for the lagging current (due to the inductive load) in the SCRs, this lagging current will change the SCR's turn off time, which means that current will still be flowing in the SCR after the voltage has crossed zero (figure 4.1). The simulation device does supply a reasonable source since the 180 Hz current pulses that are needed to keep other parts of the simulation from saturating are now generated (figure 4.17).

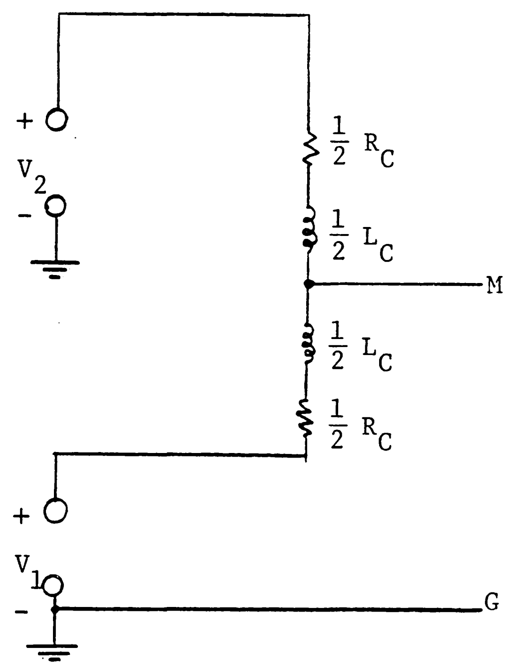


Figure 4.14 Circuit Diagram for SCR Bridge and Choke

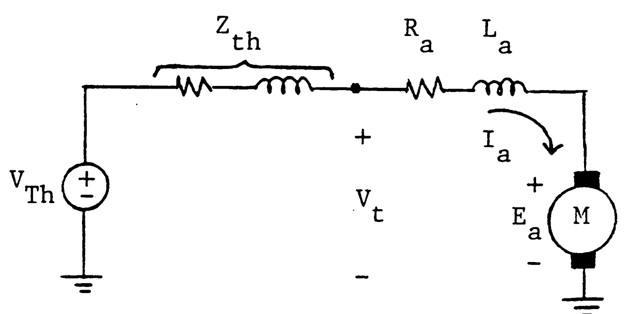
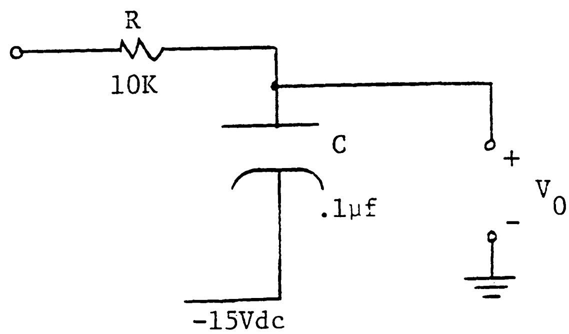


Figure 4.15 Bridge Circuit Redrawn



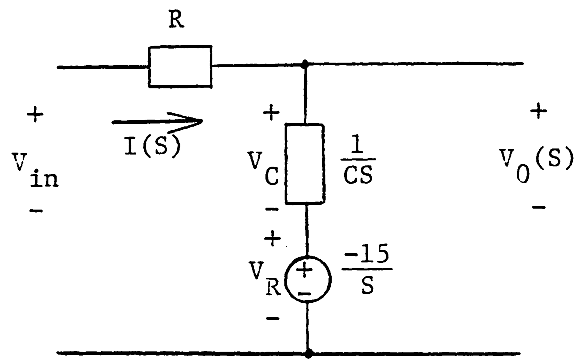


Figure 4.16 Firing Circuit Filter Network

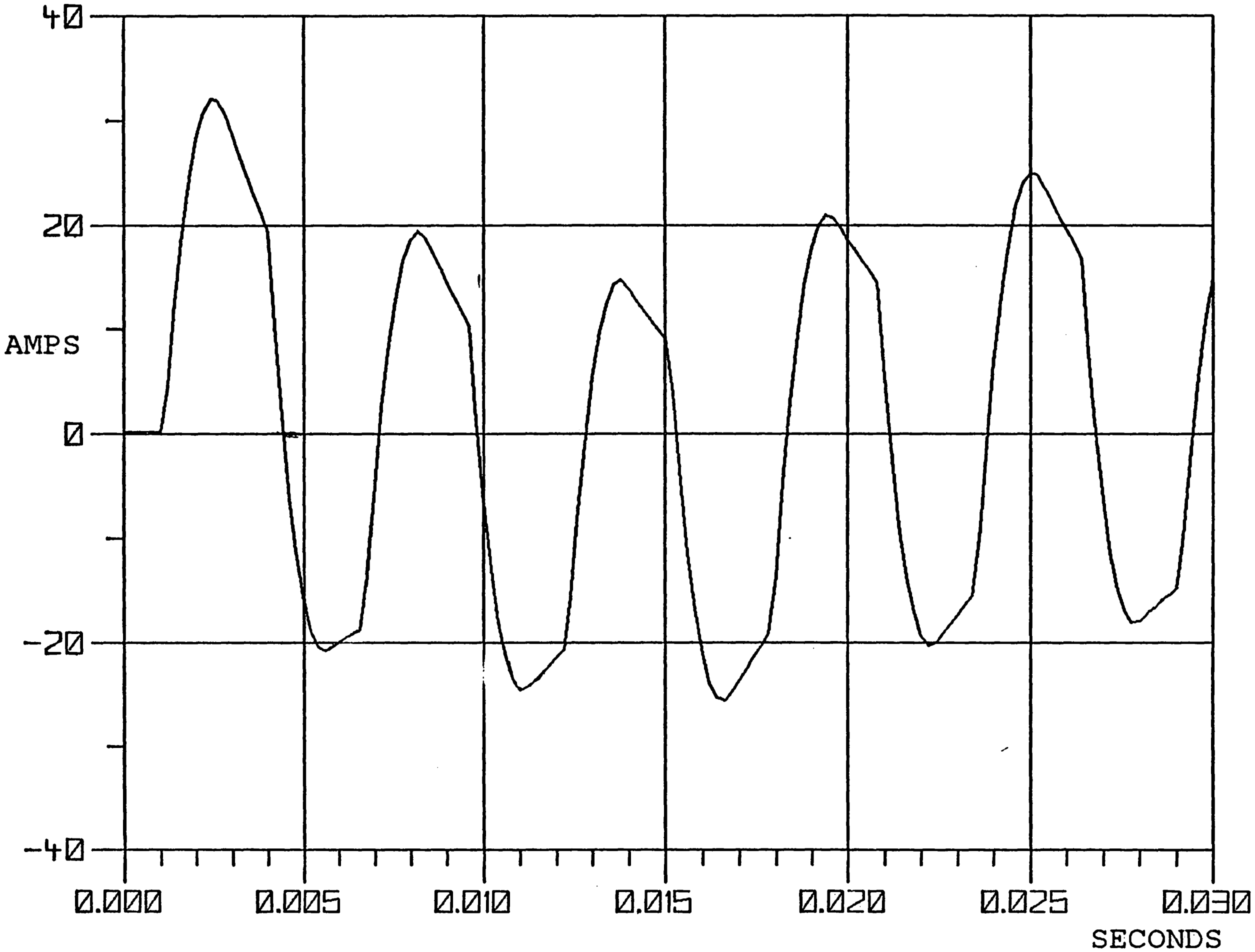


Figure 4.17 Motor Current with Bridge Simulation Input

Chapter V

RESULTS - TOTAL SYSTEM MODEL AND SIMULATION

The two preceding sections showed the development of the simulation model for each functional module of the TPAR-3330 servo amplifier. The total system simulation model can now be formed by cascading each of the four functional module models. The differential amplifier circuit (section 3.2) of the Rate Loop card is treated as a -1 gain times the speed scale factor. The response of this network to step inputs was shown to be very fast and its primary function in the drive is to produce an accurate speed reference voltage. All other modules are cascaded exactly as they were developed in their respective sections (figure 5.1).

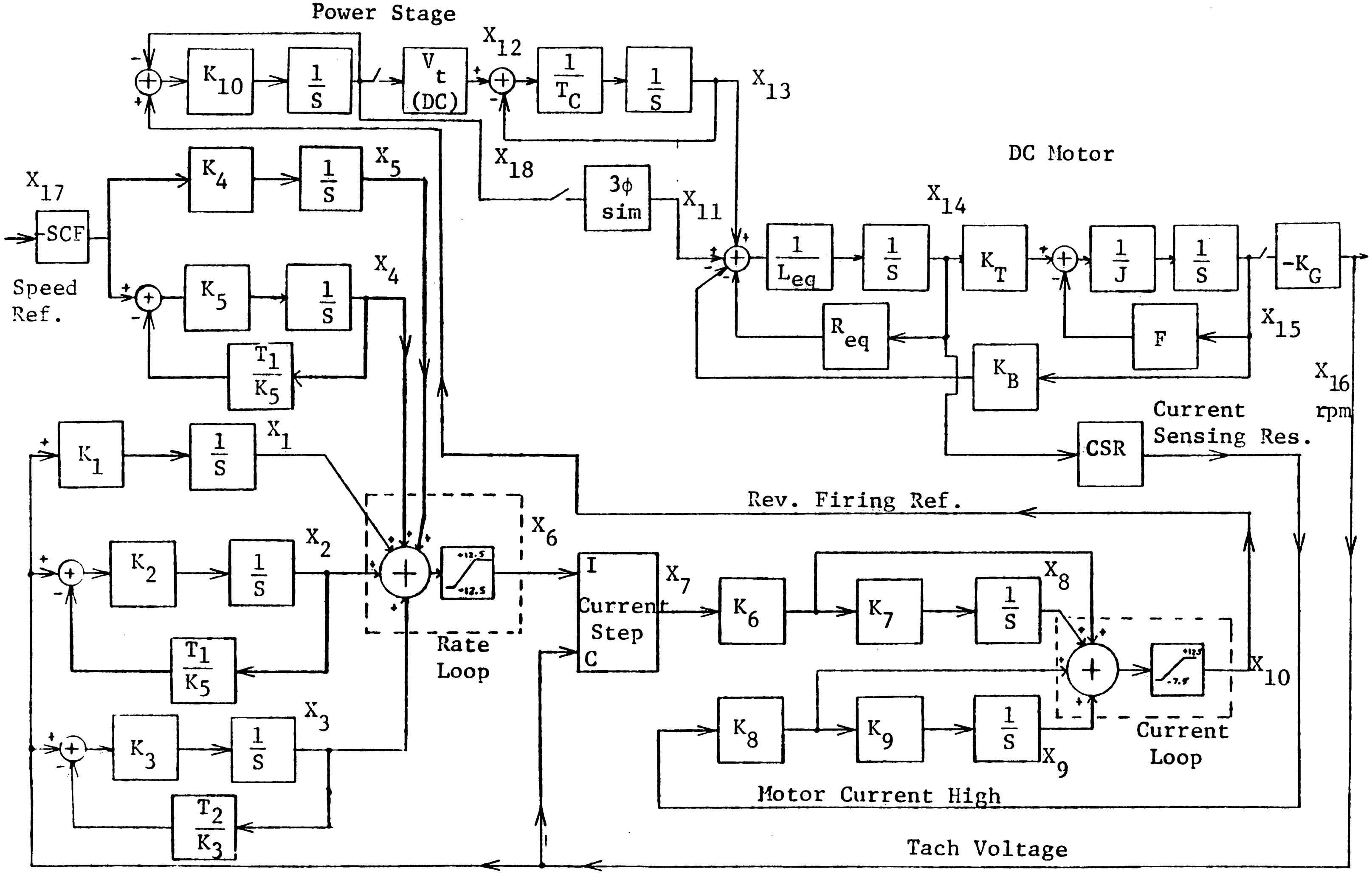


Figure 5.1. Total System Model.

5.1 MATRIX ENTRIES FOR FULL SYSTEM SIMULATION

Now that a complete system model has been established, the matrices necessary for simulation can be completed. As discussed in Chapter 1, the state transistion approach requires four matrices (A,J,N,X) to simulate a system response. The elements of the A matrix are easily determined from inspection of the simulation model. Because the output of each integrator (1/s) has been selected as a state variable, X_i, the input of the integrator will correspond to $\dot{X}_i$. Also, $\dot{X}_i$ for the nonlinear elements is equal to zero. The resulting A matrix is shown in figure 5.2. The second matrix to be defined is the Jump matrix (J). The Jump matrix entries handle both the real and induced sampling in the simulation model by calculating the simulation variables, $X(KT^+)$, as discussed in Chapter 1. Note that the nonlinear elements are the only devices that have diagonal elements that are not equal to 1. The J matrix entries for each nonlinear device is unique and must be determined by examining the description of the device in question. The J matrix for this system is shown in figure 5.3. The third matrix to be defined is the nonlinear description matrix (N). The N matrix describes the type of simulation device being used. The N matrix for the simulation model of the TPAR-3330 drive system is shown in figure 5.4. The N matrix is composed of 6 different types of elements; a literal description of each element is given below:

1 - continuous element
2 - linear device with saturation
9 - summing junction with limits
10 - current step - speed switch
12 - dc terminal voltage
13 - 3 phase converter simulator

The final matrix is the X vector. This vector is the output vector for the system states. The intial conditions for the drive are entered in this vector before the simulation is started. The sample rate, T, is very important in a simulation of a system like this one. The value for T, in this simulation, was chosen to be 0.00015 seconds. A value of this magnitude was needed because of the module that is simulating the power stage. Since the balanced 3 phase source is sampled at each period, a large value of T could mean that a large change in the sinusoids could occur between samples. For this case, T equal to 0.00015 seconds, the power source is sampled every 3.40 degrees.

$$
\begin{array}{cc}
 & \begin{array}{cccccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \end{array} &
\left[\begin{array}{cccccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_1 & 0 & 0 \\
0 & -T_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_2 & 0 & 0 \\
0 & 0 & -T_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_3 & 0 & 0 \\
0 & 0 & 0 & -T_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_{11} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & K_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_{14} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{T_C} & \frac{-1}{T_C} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{L_{eq}} & 0 & \frac{1}{L_{eq}} & \frac{-K_B}{L_{eq}} & \frac{-R_{eq}}{L_{eq}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{K_T}{J} & \frac{-F}{J} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -K_{10}
\end{array}\right]
\end{array}
$$

$$K_{11} = -SCF \cdot K_5$$

$$K_{12} = -SCF \cdot K_4$$

$$K_{13} = K_6 \cdot K_7$$

$$K_{14} = K_9 \cdot K_8 \cdot CSR$$

Figure 5.2 A Matrix

$$\begin{array}{rc}
 & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \end{matrix} \\
\begin{matrix} 1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\\15\\16\\17\\18 \end{matrix} &
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & K_6 & 1 & 1 & 0 & 0 & 0 & 0 & C_1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -K_G & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\end{array}$$

$$C_1 = CSR \cdot K_8$$

Figure 5.3 J Matrix

$$
\begin{array}{rc}
 & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\begin{array}{r} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \end{array} &
\left[\begin{array}{ccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & 5 & 1 & 2 & 3 & 4 & 5 \\
10 & -9.4 & 9.4 & -3 & 3 & 16 & 28 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & -7.5 & 12 & 0 & 0 & 0 & 0 \\
13 & 0 & 18 & 0 & 0 & 0 & 0 \\
12 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array}
$$

Figure 5.4 N Matrix

5.2 SYSTEM SIMULATION (WITH CALCULATED VALUES)

The TT-2953B DC motor used in this drive system has a maximum speed of 3000 rpm. Therefore, three velocities, 0 rpm, 1000 rpm clockwise and 2000 rpm counterclockwise, were chosen to evaluate the simulation model of figure 5.1. The reference to clockwise or counterclockwise refers to the direction the shaft rotates when the user is facing the shaft. Also, three signals in the drive system were chosen for evaluation at each velocity. The following three signals were chosen for comparison:

a) Motor Velocity - Tachometer output

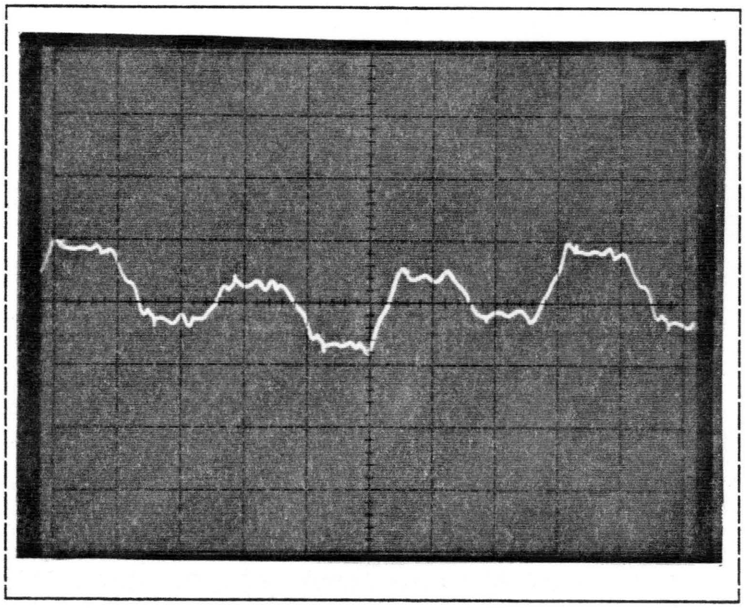
b) Rate Loop Card output

c) DC Motor Current

Figures 5.5,5.6,5.7 show the actual oscillograms of each of these signals in the drive at the shaft velocities desired. The calculated values for each of the variables in the system simulation model, figure 5.1, are shown in table 5.1. For a desired speed of 0 rpm, the speed reference input voltage, R, in figure 2.1, is equal to 0 volts. A simulation was run with the speed reference equal to 0 volts and the output of the simulation is shown in figures 5.8a,b,c. These results show that the average tachometer voltage is oscillating around zero. The magnitude of this oscillation is about 0.5 volts, while the actual tachometer voltage has oscillations of a smaller magnitude (0.1v). Because the tachometer output is one of the inputs to the Rate Loop, the simulated Rate Loop output, figure 5.8b, also shows a magnitude that is larger than the actual Rate Loop output of figure 5.5b. The simulated motor current, figure 5.8c, is over twice the magnitude of the actual motor current that is shown in figure 5.5c. This large value of the motor current is the cause of the other signals having magnitudes that are larger than the actual signals. The simulations for the 1000 rpm case, figures 5.9a,b,c, and the 2000 rpm case, figure 5.10a,b,c, also show the effect of the motor current. The tachometer signals, figures 5.9a and 5.10a, are headed for the correct final value, but the ripple on top of the simulated tachometer signals is more pronounced than the actual ripple that is shown in figures 5.6a and 5.7a. Again, this is due to the larger motor currents (figure 5.6c and figure 5.7c). The Rate Loop signals at these speeds do not compare as well as the tachometer signals, because of the large tachometer ripple.

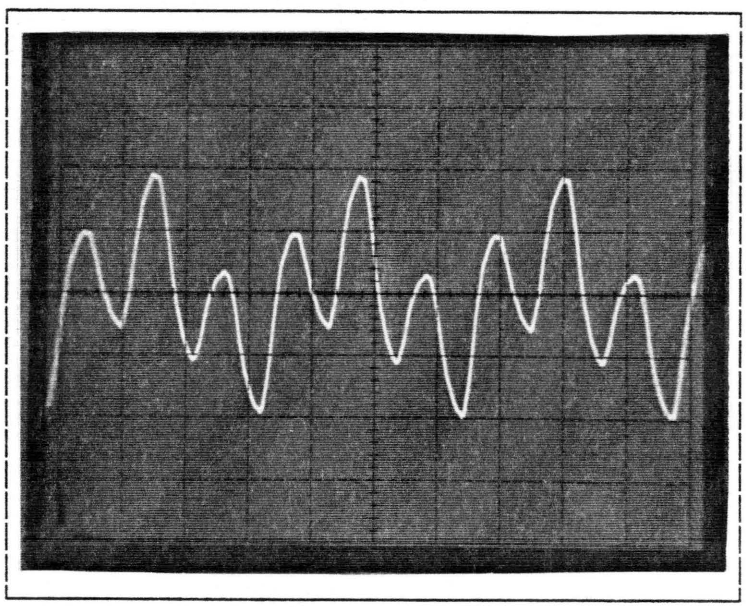
Table 5.1 Calculated Constants for Total System Model

SCF	0.442
K_1	-238.07
K_2	-27995.28
K_3	3399.7
K_4	-1964.21
K_5	-89309.98
K_6	-0.1
K_7	202.02
K_8	-1.83
K_9	202.02
K_{10}	1000
T_1	1408.
T_2	303.03
T_3	30.30
T_c	0.2
CSR	0.01
L_{eq}	1.435 mH
R_{eq}	0.975 ohms
K_T	0.368 Nm/Amp
K_B	0.368 V/rad/sec
J	2.74×10^{-3} kg m^2
F	0.00145 Nm sec/rad
K_G	0.11937 V/rad/sec



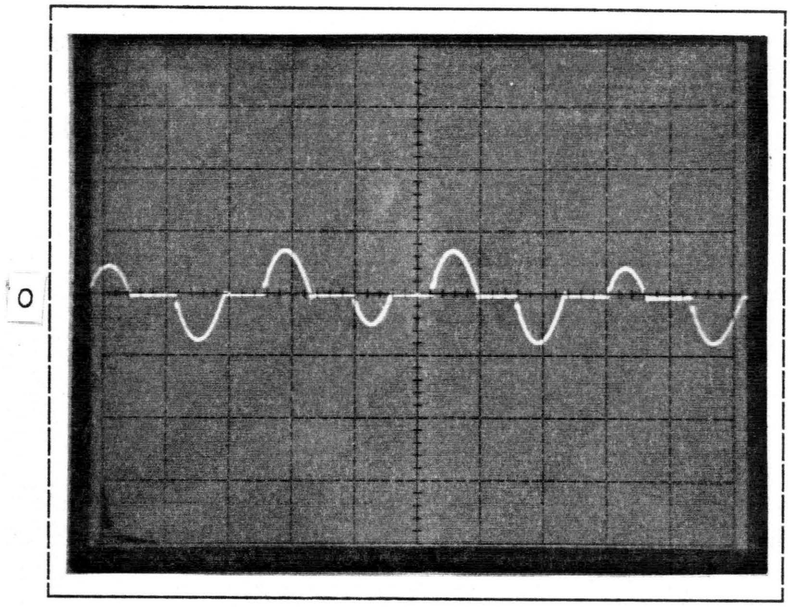
VERT. = 0.1 V/DIV
HORIZ. = 2.0 ms/DIV

Figure 5.5a Tachometer Output at Zero RPM



VERT. = 1.0 V/DIV
HORIZ. = 5.0 ms/DIV

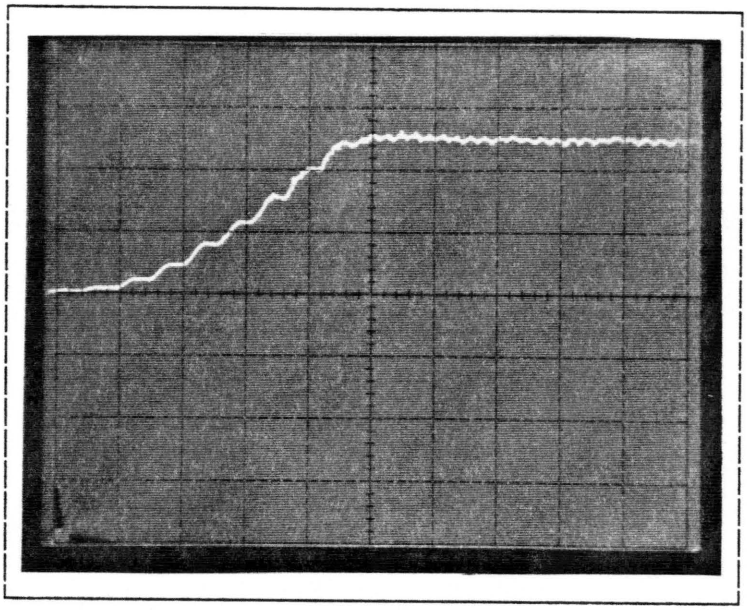
Figure 5.5b Rate Loop Output at Zero RPM



VERT. = 10.0 A/DIV
HORIZ. = 2.0 ms/DIV

Figure 5.5c Motor Current at Zero RPM

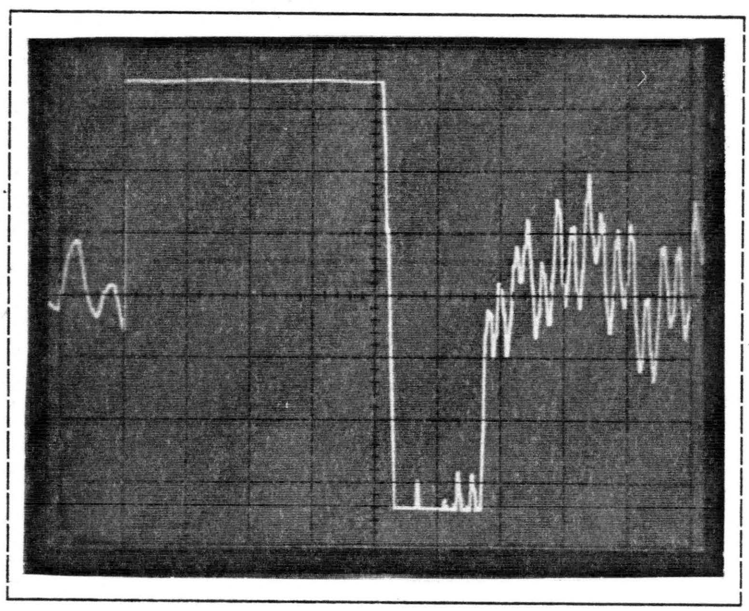
0



VERT. = 5.0 V/DIV
HORIZ. = 10.0 ms/DIV

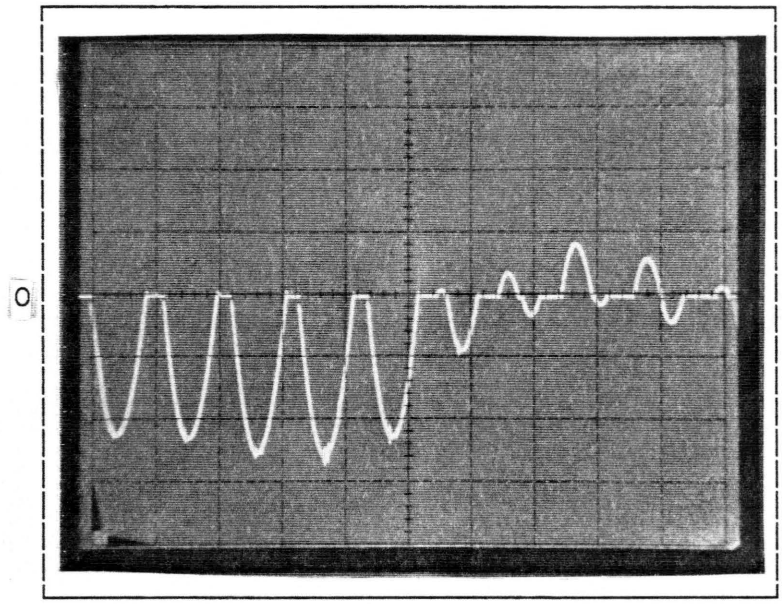
Figure 5.6a Tachometer Output at 1000 RPM (ccw)

0



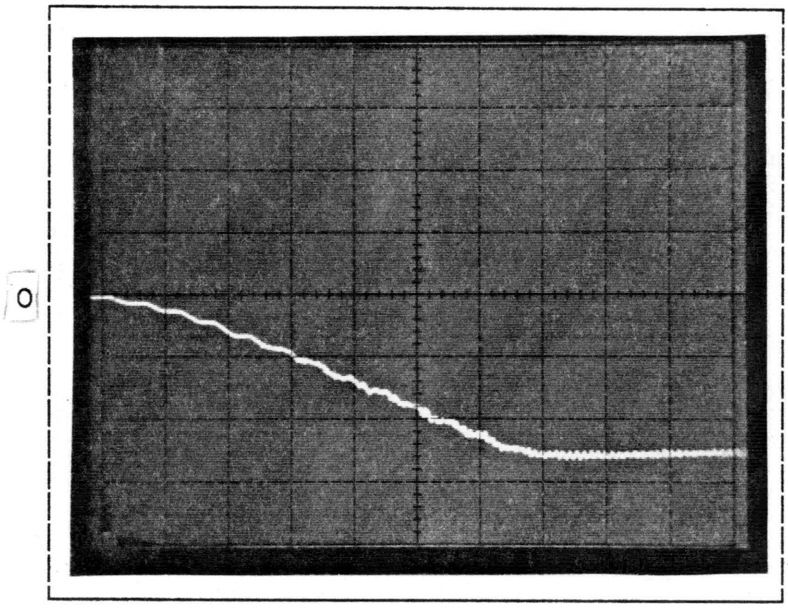
VERT. = 2.0 V/DIV
HORIZ. = 10.0 ms/DIV

Figure 5.6b Rate Loop Output at 1000 RPM (ccw)



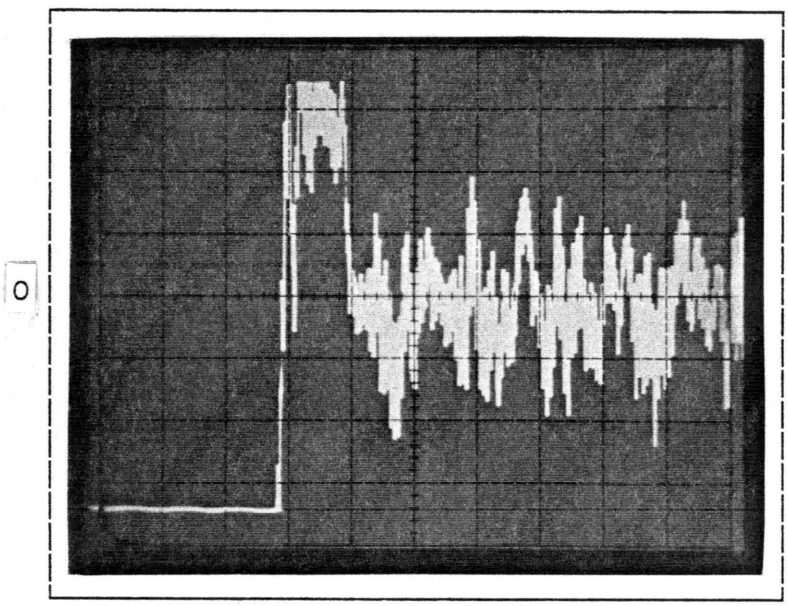
VERT. = 20.0 A/DIV
HORIZ. = 5.0 ms/DIV

Figure 5.6c Motor Current at 1000 RPM (ccw)



VERT. = 10.0 V/DIV
HORIZ. = 10.0 ms/DIV

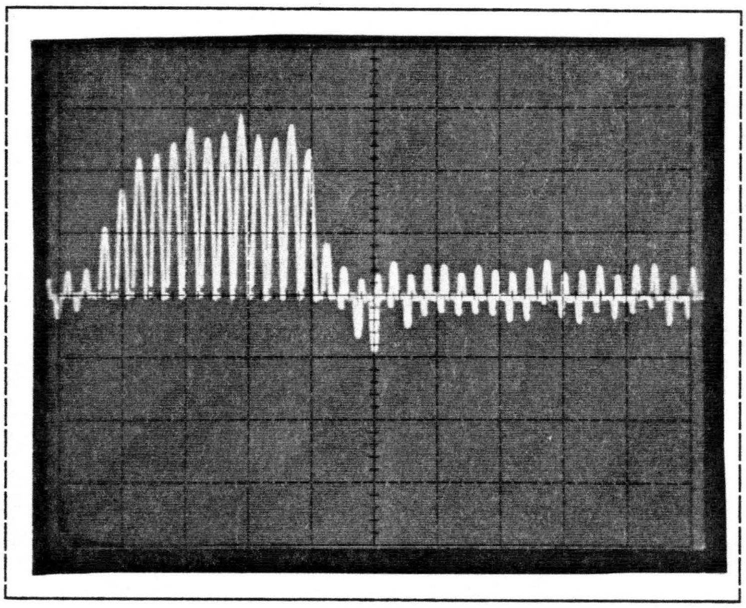
Figure 5.7a Tachometer Output at 2000 RPM (cw)



VERT. = 2.0 V/DIV
HORIZ. = 20.0 ms/DIV

Figure 5.7b Rate Loop Output at 2000 RPM (cw)

0



VERT. = 20.0 A/DIV
HORIZ. = 20.0 ms/DIV

Figure 5.7c Motor Current at 2000 RPM (cw)

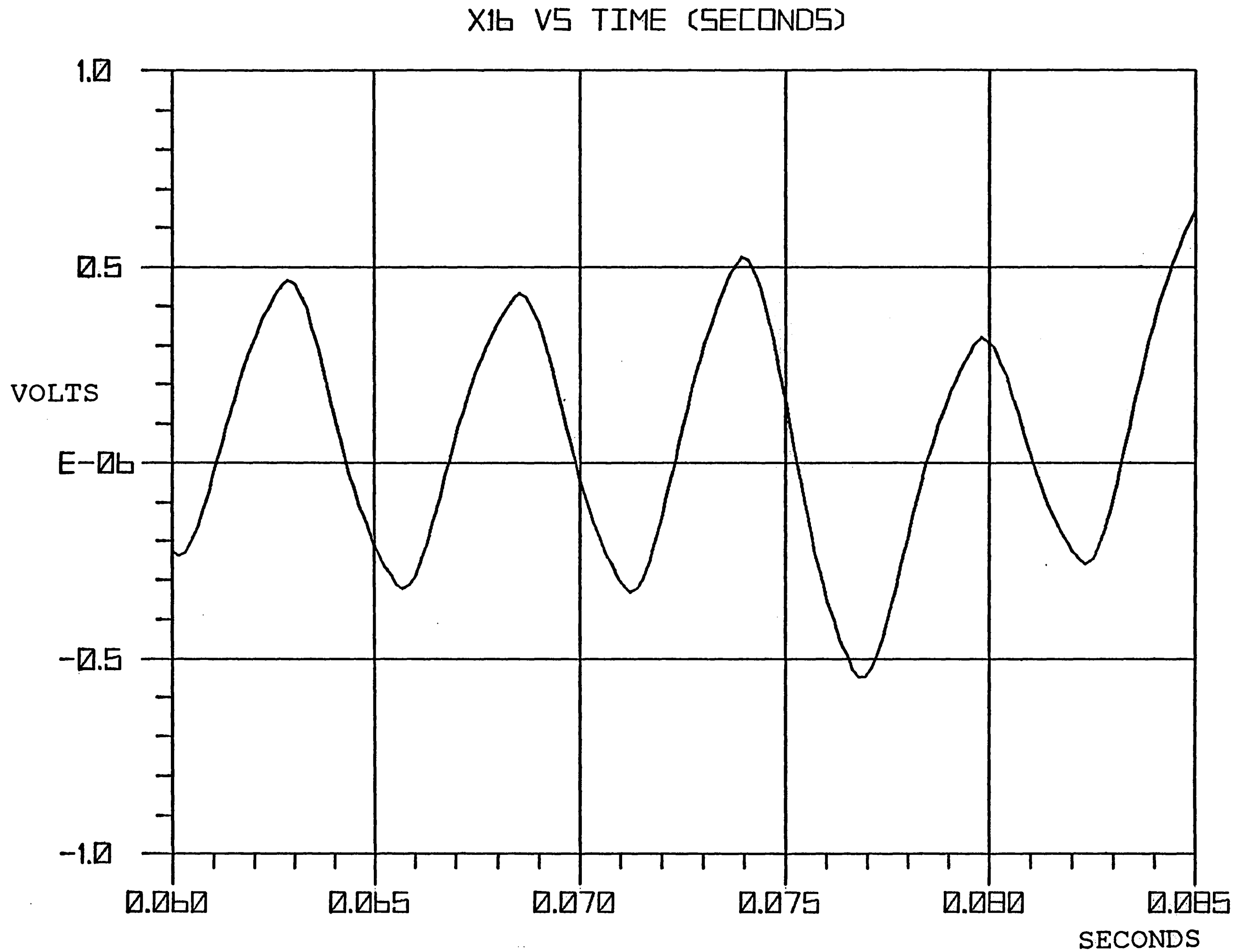


Figure 5.8a Simulated Tachometer Voltage at Zero Speed

(using calculated values)

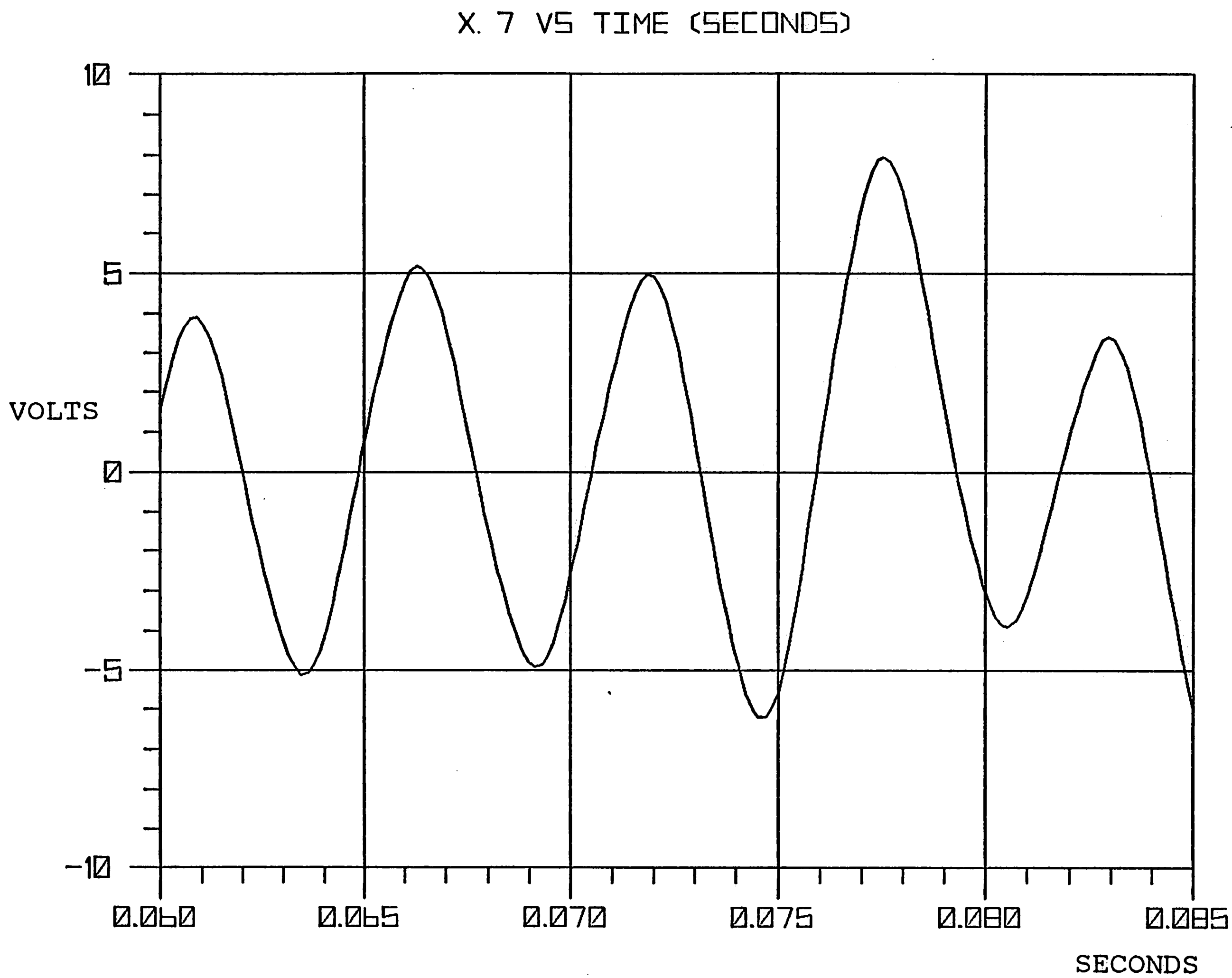


Figure 5.8b Simulated Rate Loop Output at Zero Speed

(using calculated values)

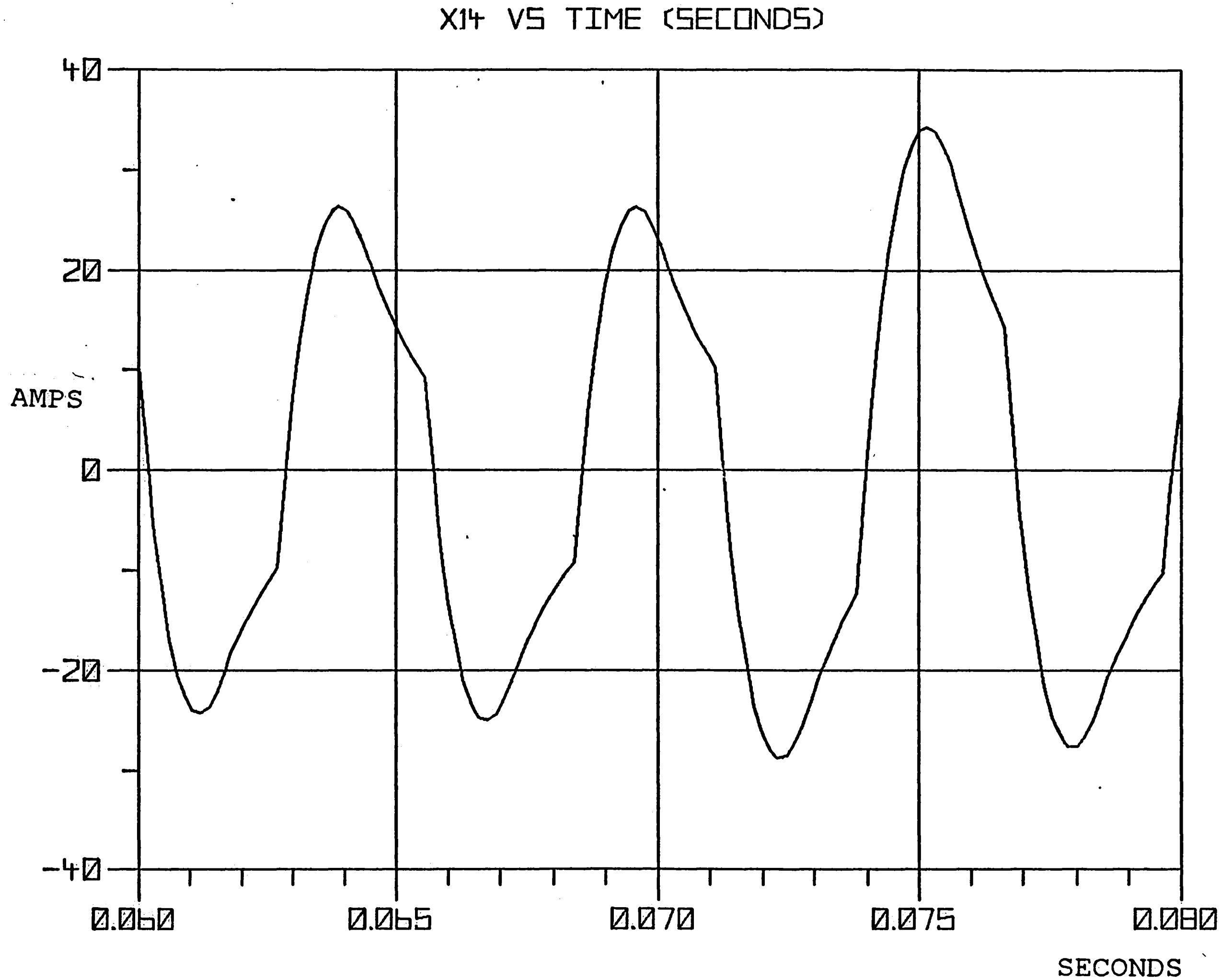


Figure 5.8c Simulated Motor Current at Zero Speed

(using calculated values)

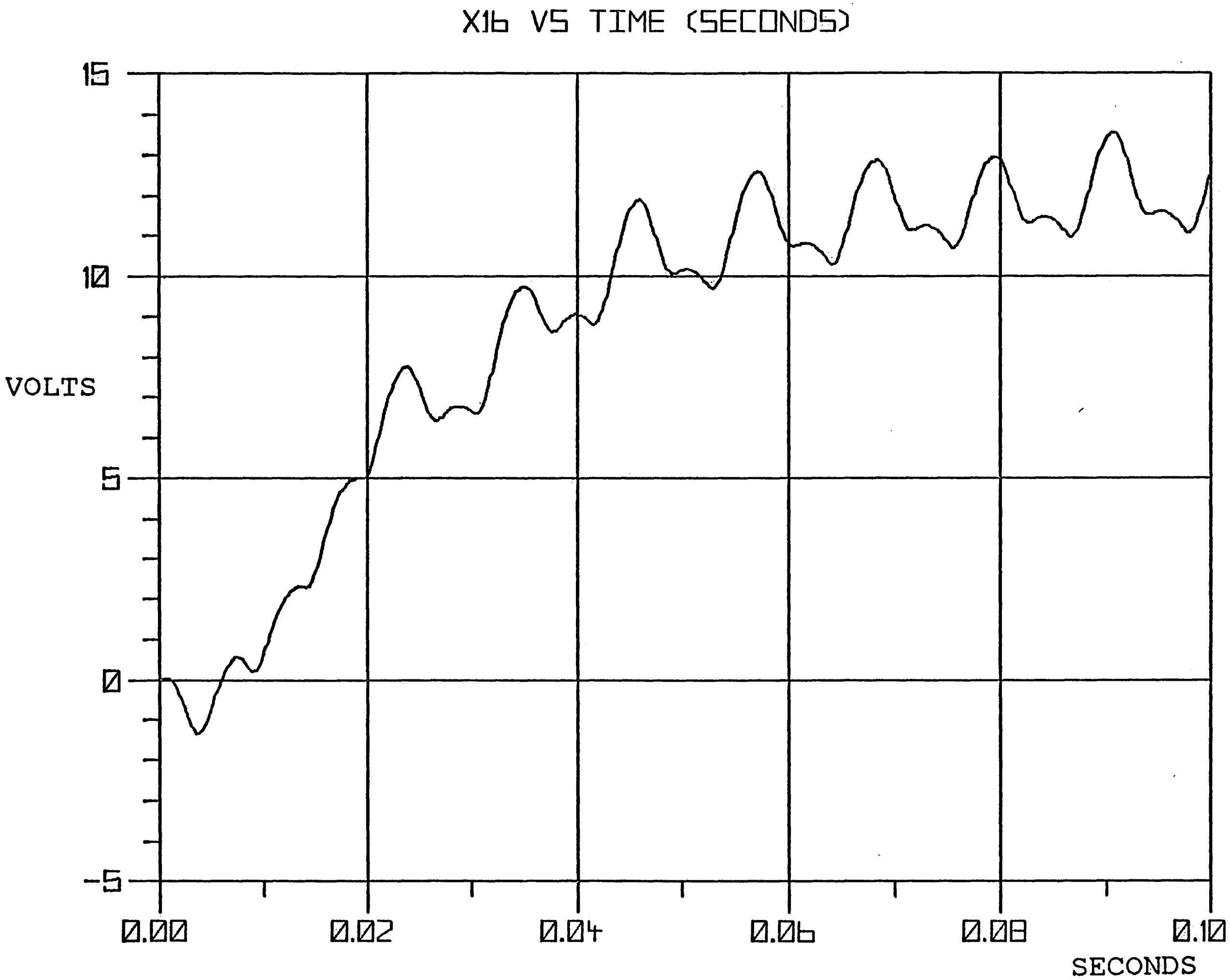


Figure 5.9a Simulated Tachometer Output at 1000 RPM (ccw)

(using calculated values)

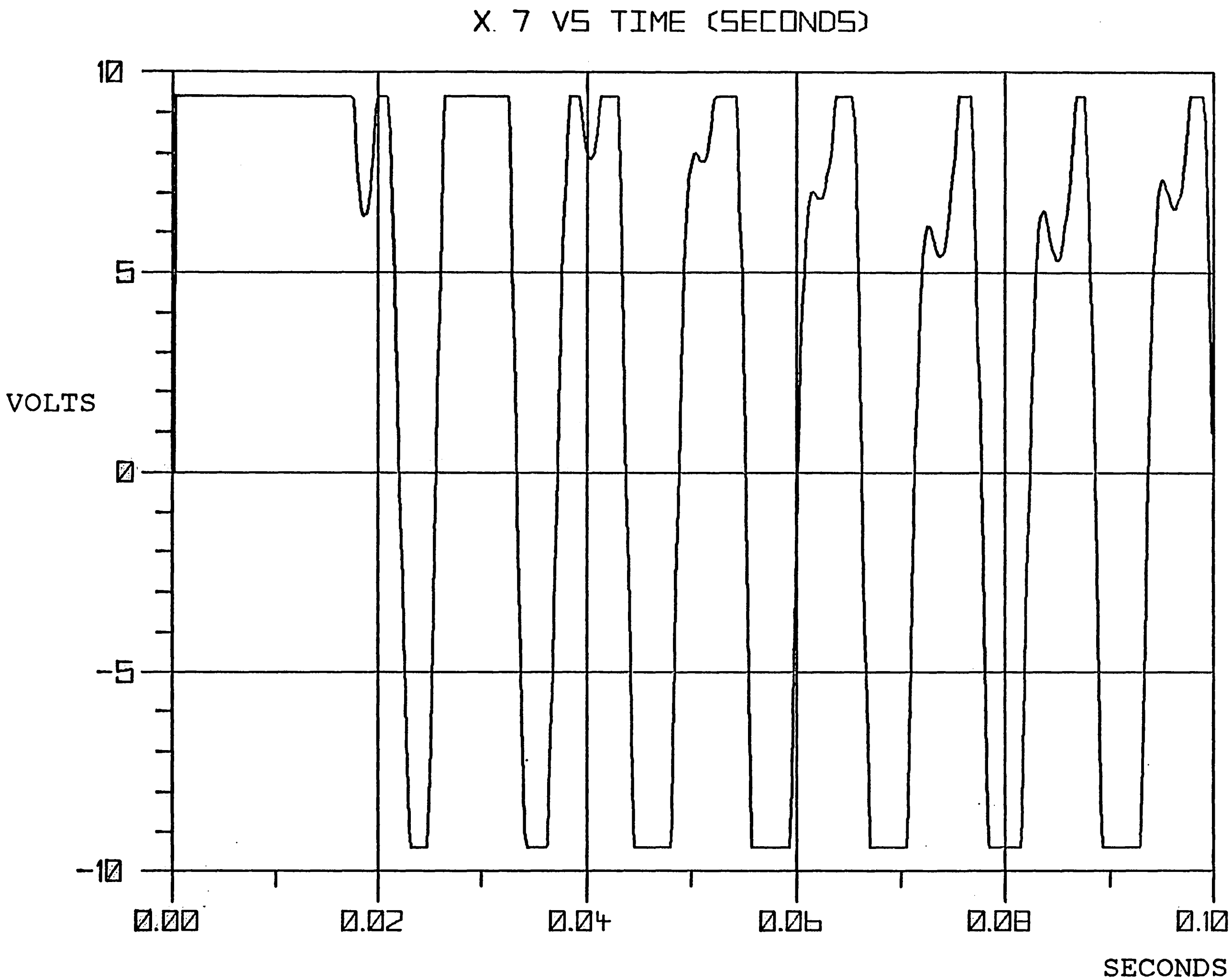


Figure 5.9b Simulated Rate Loop Output at 1000 RPM (ccw)

(using calculated values)

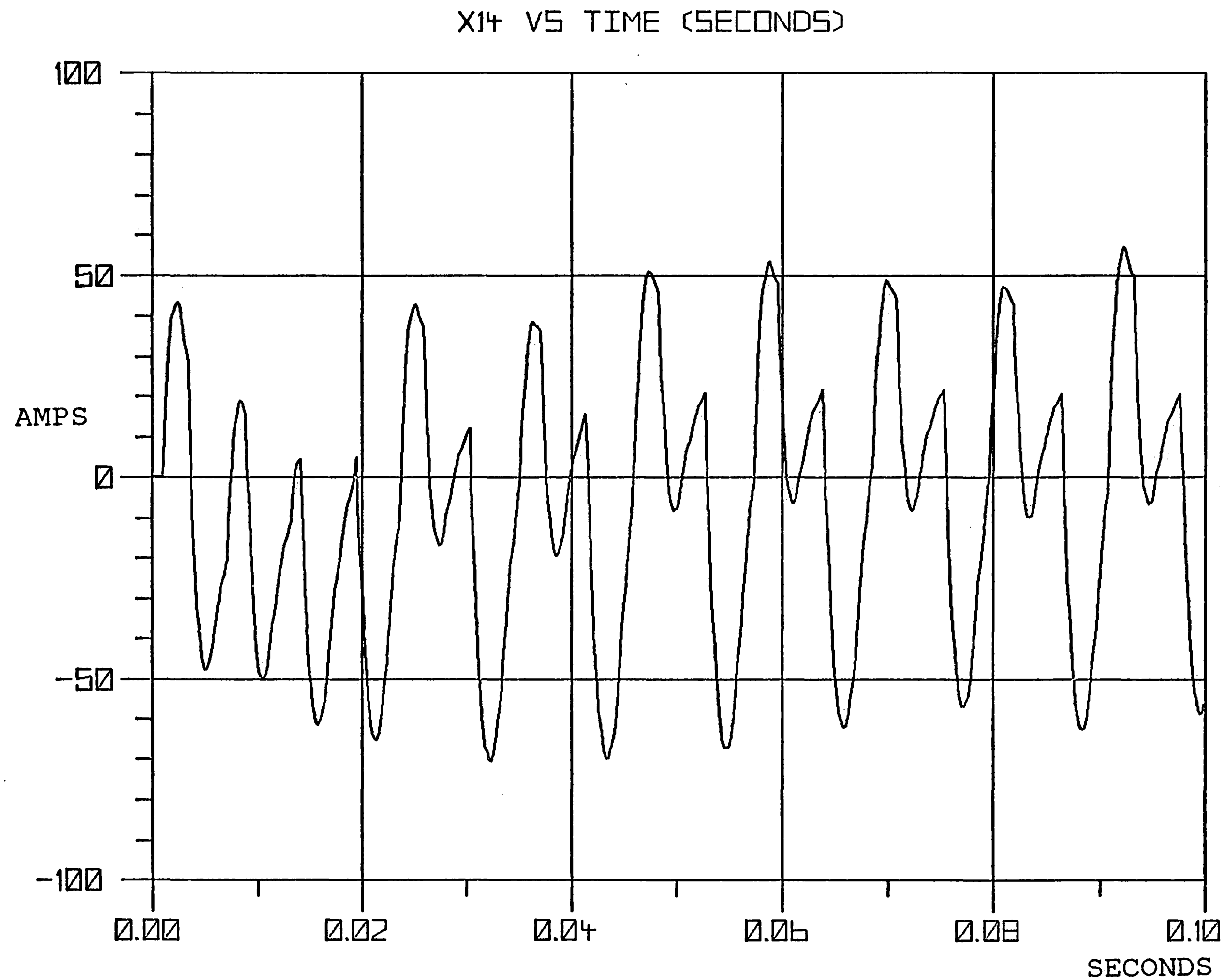


Figure 5.9c Simulated Motor Current at 1000 RPM (ccw)

(using calculated values)

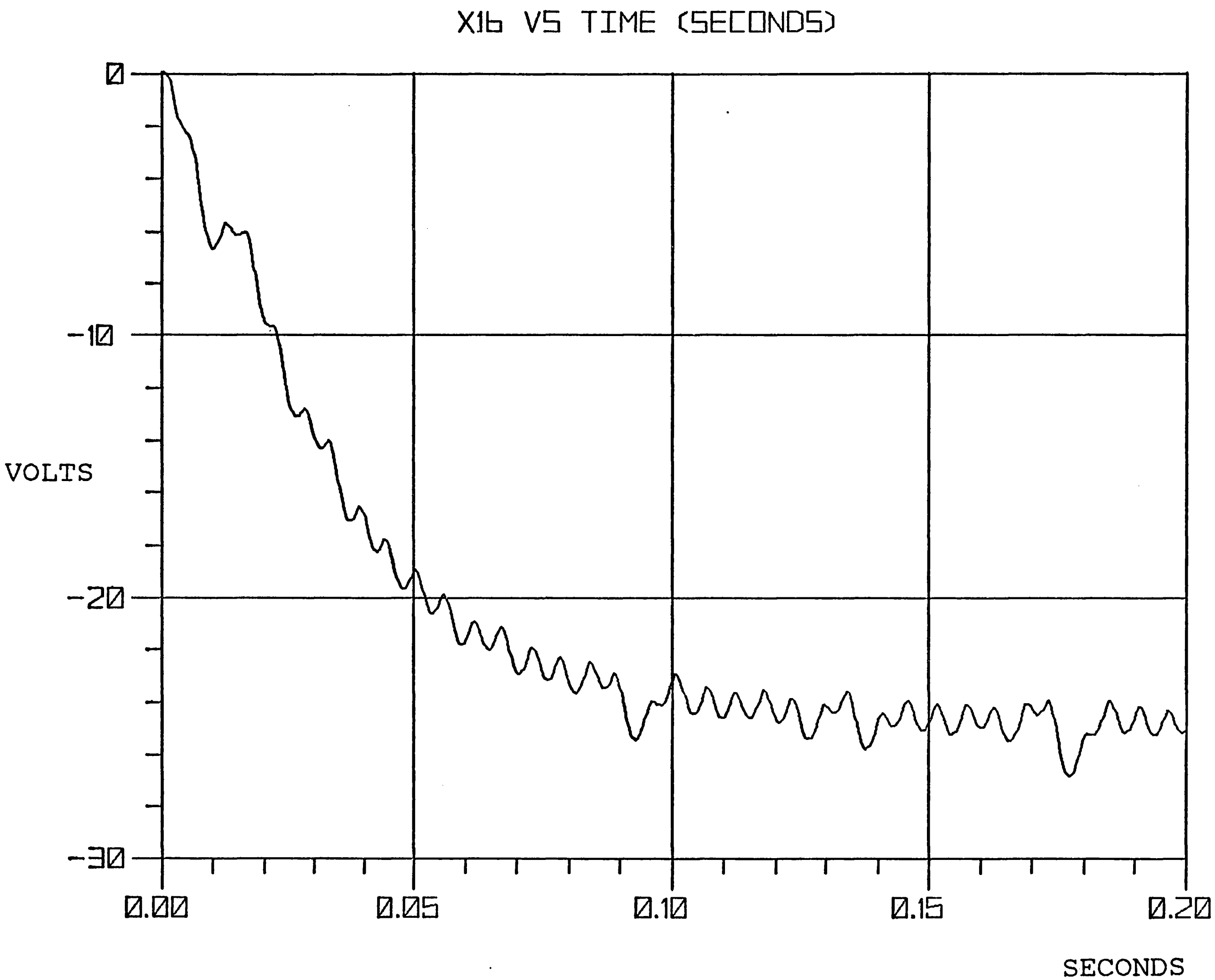


Figure 5.10a Simulated Tachometer Output at 2000 RPM (cw)

(using calculated values)

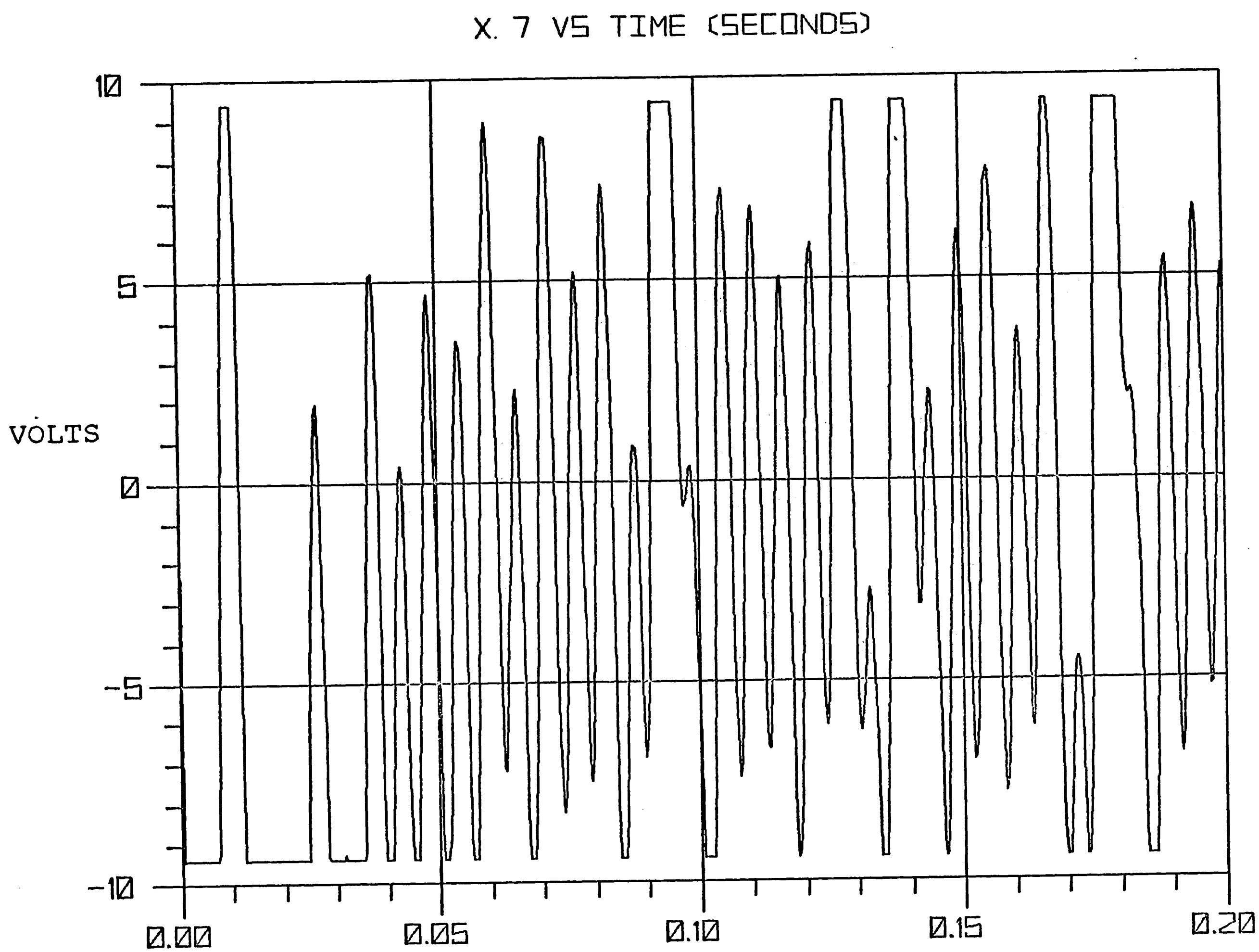


Figure 5.10b Simulated Rate Loop Output at 2000 RPM (cw)

(using calculated values)

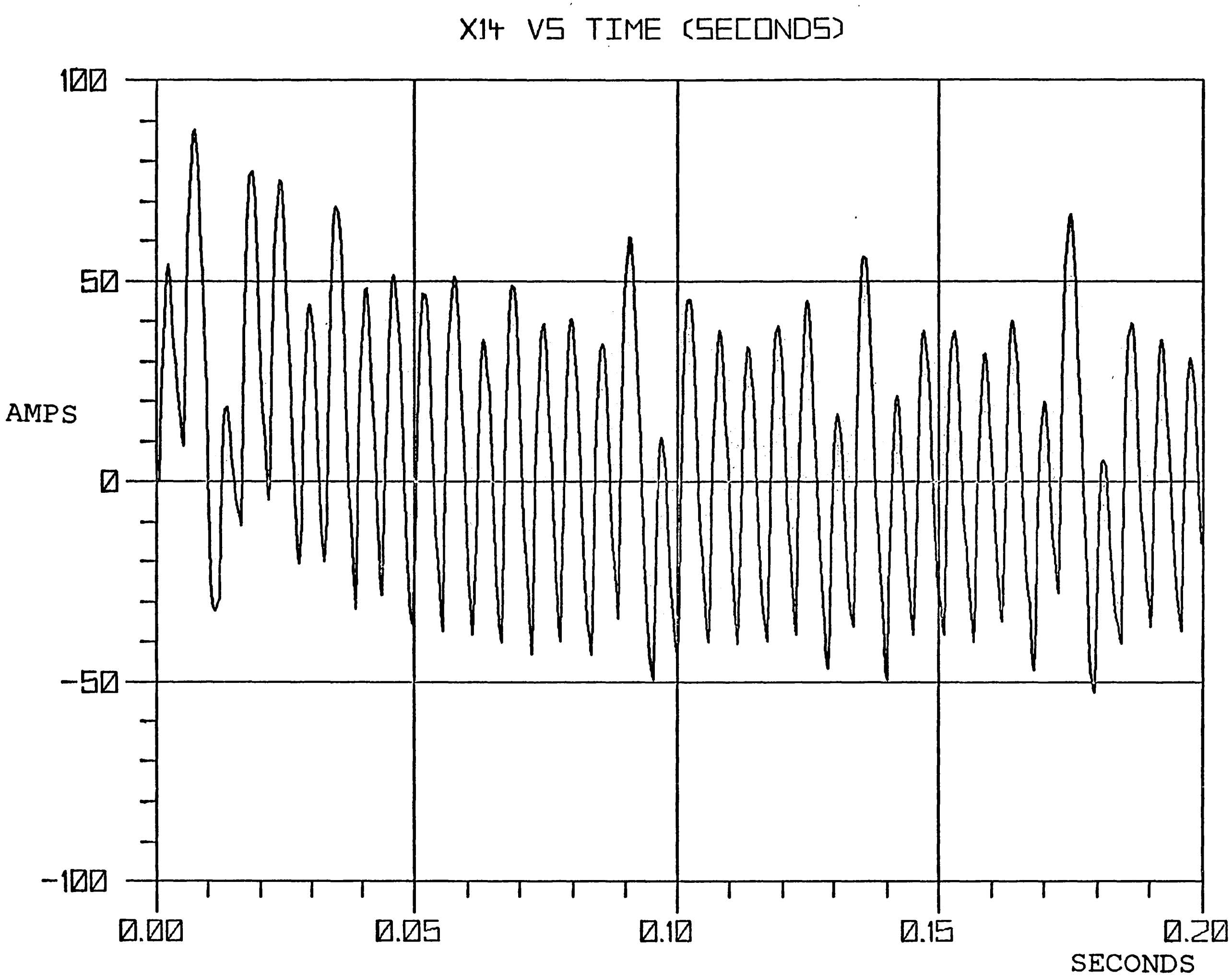


Figure 5.10c Simulated Motor Current at 2000 RPM (cw)

(using calculated values)

5.3 SYSTEM SIMULATION (WITH EMPIRICAL VALUES)

Although the final speed of the DC motor and the Rate Loop error signal agreed somewhat with the actual values, the motor current did not match very well. The simulated tachometer voltage in section 5.2 revealed that the control loops made the motor seek the correct speed, however, the tachometer voltage signal contained a ripple voltage that was higher than the ripple voltage on the actual signal. This ripple voltage caused the simulated Rate Loop output not to match the actual Rate Loop output very well. The large ripple is due to the fact that the simulated motor current is much larger than the actual motor current. By varying the values R_{eq} and L_{eq} in the DC motor model, more realistic values may be obtained for the motor current. Recall that the R_{eq} and the L_{eq} in the DC motor model (figure 5.1) are the sum of the contributions due to the choke and the armature components of the DC motor. Since the values of the components of the DC motor vary over a wide range of operating frequency, these elements make good candidates to be varied. The R_{eq}/L_{eq} ratio controls the response of the armature circuit. The value of Req was varied until a reasonable amount of discontinuity appeared in the motor current simulation at zero speed. A value of R_{eq}/L_{eq} that resulted in a realistic waveform was 2000.

Once this value for the R_{eq} and L_{eq} ratio was established at 0 rpm, the simulations for 1000 rpm and 2000 rpm were run. Figures 5.11, 5.12, 5.13 show the results of the simulation with the empirical values for the DC motor armature components. The tachometer voltage, figure 5.12a, matches the actual tachometer voltage, figure 5.6b, very well. Note that the settling time is approximately 60 ms in both figures. Also, note that the simulation shows the slight overshoot of the motor speed and it also shows the 0.5 volt ripple (180 Hz) on the waveform. The Rate Loop simulation results are also very convincing. The simulated response for 1000 rpm (ccw), figure 5.12b, shows that the Rate Loop output is saturated in the positive direction for about 38 ms, this is very close to the actual value that is shown in figure 5.6b. The difference in the saturation limits is due to the fact that 9.4 volts was calculated to be the breakdown voltage for the zener diodes in figure 3.13. The actual limits seem to be approximately 7.0 volts. Also note the similar values of the Rate Loop output once the system has reached the desired speed (zero error); both the simulation and the oscillogram show an oscillating waveform of about 4 volts maximum amplitude. Finally, the motor current was examined at 1000 rpm. The simulated motor current, figure 5.12c, and the actual motor current, figure

5.6c, also match exceptionally well. The intial transient current in both figures is in excess of -40 amps (maximum). When comparing the two curves, it should be noted that the actual motor current trace is slightly delayed because of the oscilloscope triggering procedure that was used (manual triggering). Thus, zero seconds in figure 5.6c does not exactly correspond to zero seconds in figure 5.12c. The results for the 2000 rpm case, figures 5.13a,b,c, also match the actual responses that are shown in figures 5.7a,b,c. This shows that the simulation model works well in both directions, as well as, high or low speeds. The waveforms and the magnitudes are closer to the actual drive system variables at all velocities than those obtained in section 5.2.

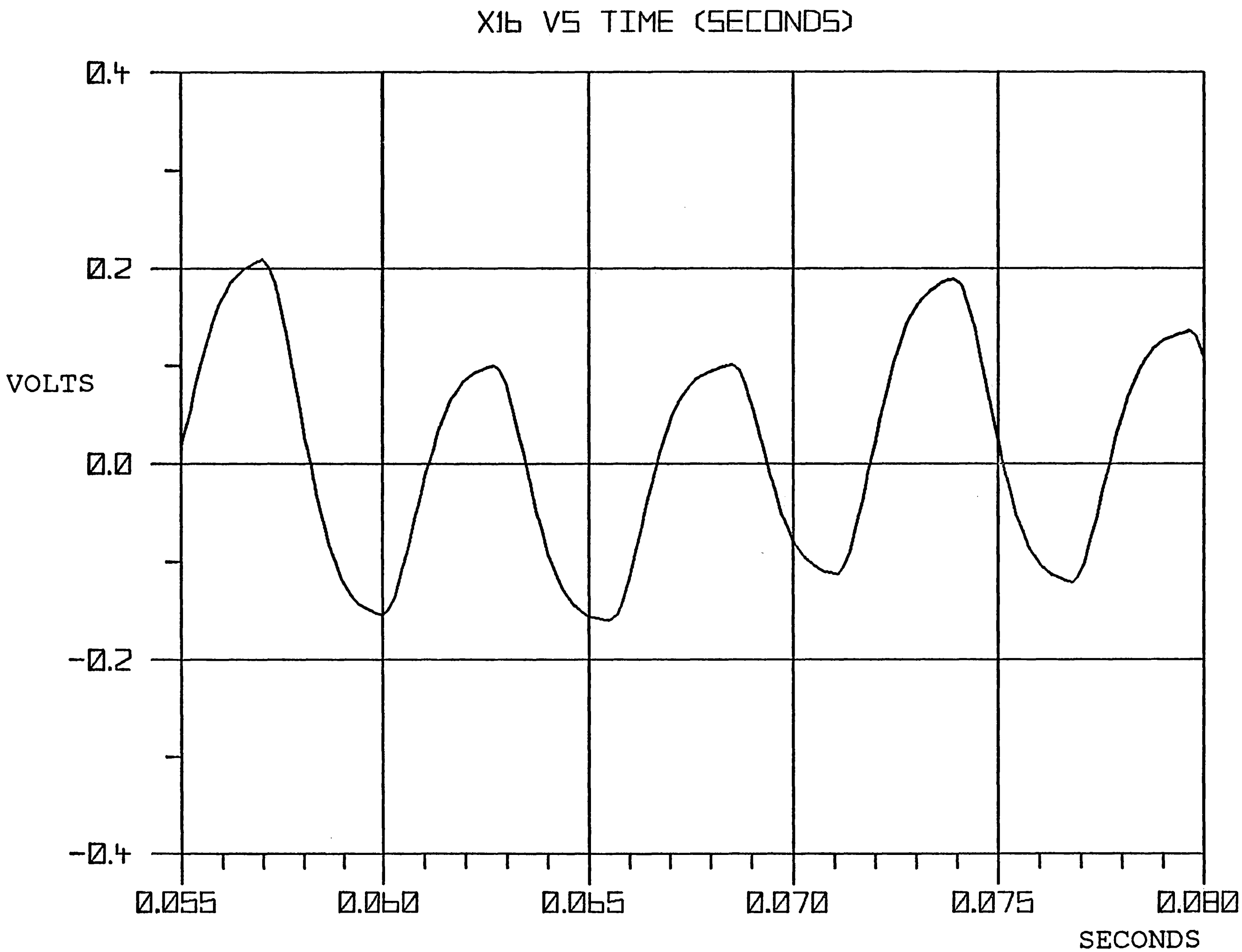


Figure 5.11a Simulated Tachometer Output at Zero Speed

(using empirical values)

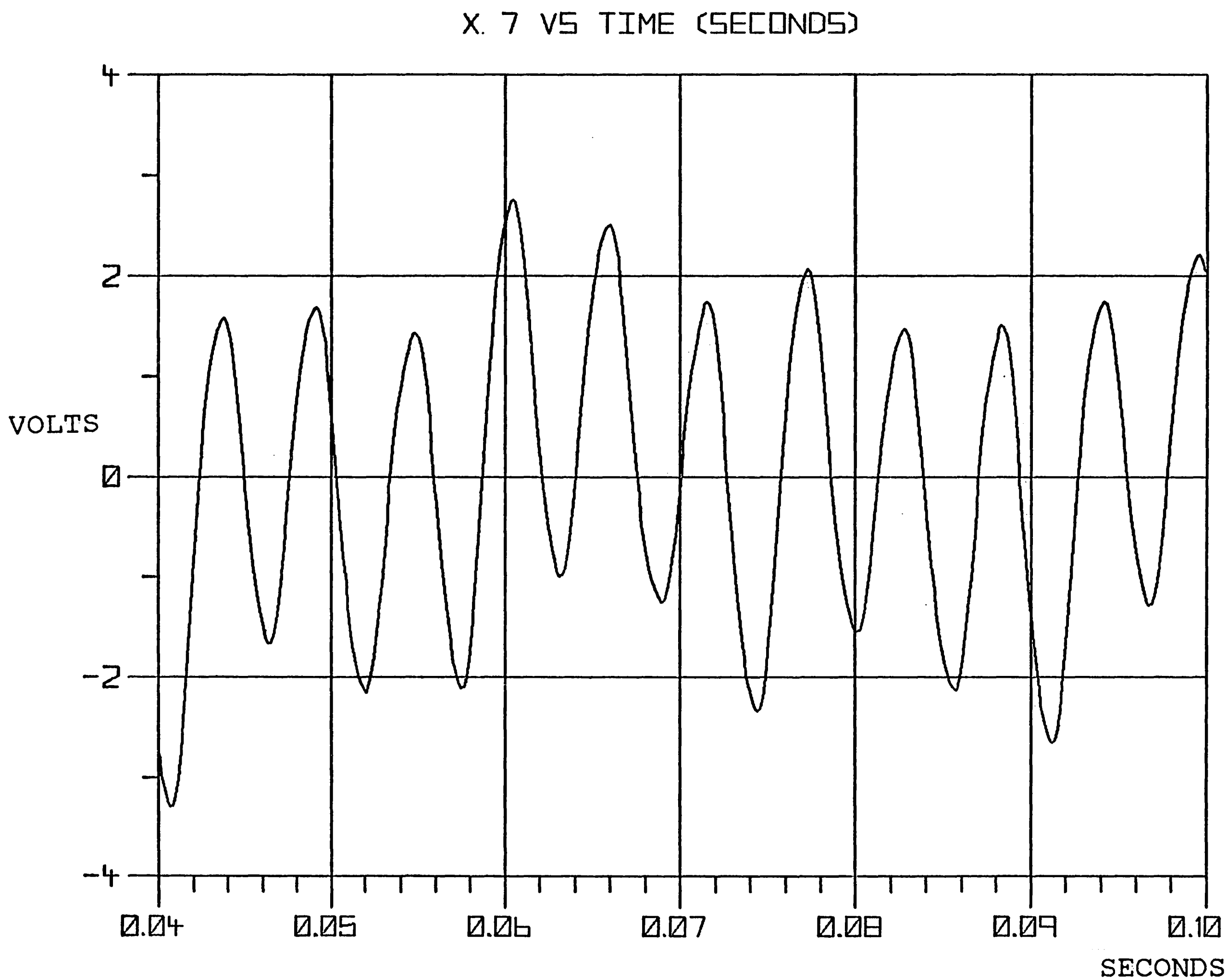


Figure 5.11b Simulated Rate Loop Output at Zero Speed

(using empirical values)

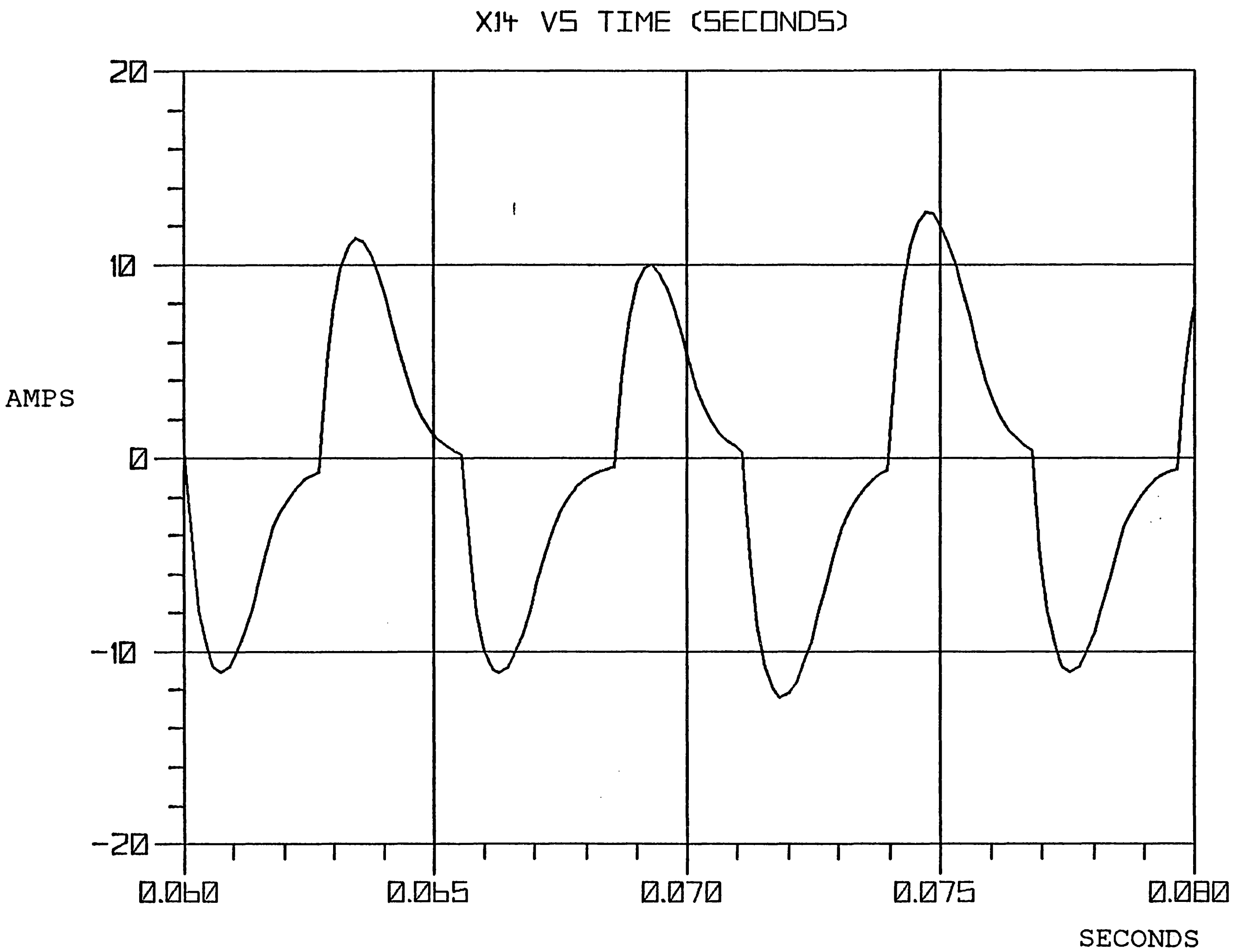


Figure 5.11c Simulated Motor Current at Zero Speed

(using empirical values)

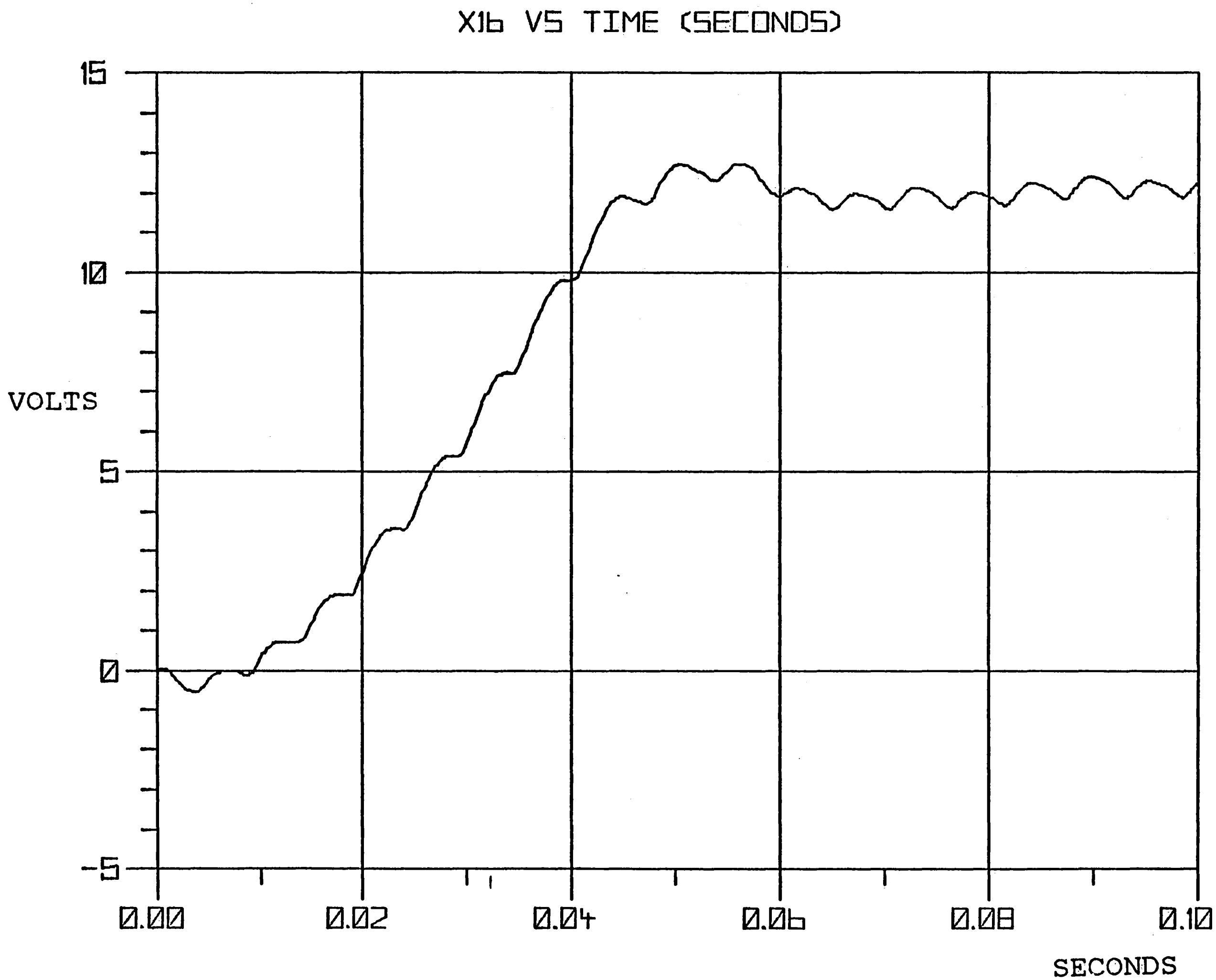


Figure 5.12a Simulated Tachometer Output at 1000 RPM (ccw)

(using empirical values)

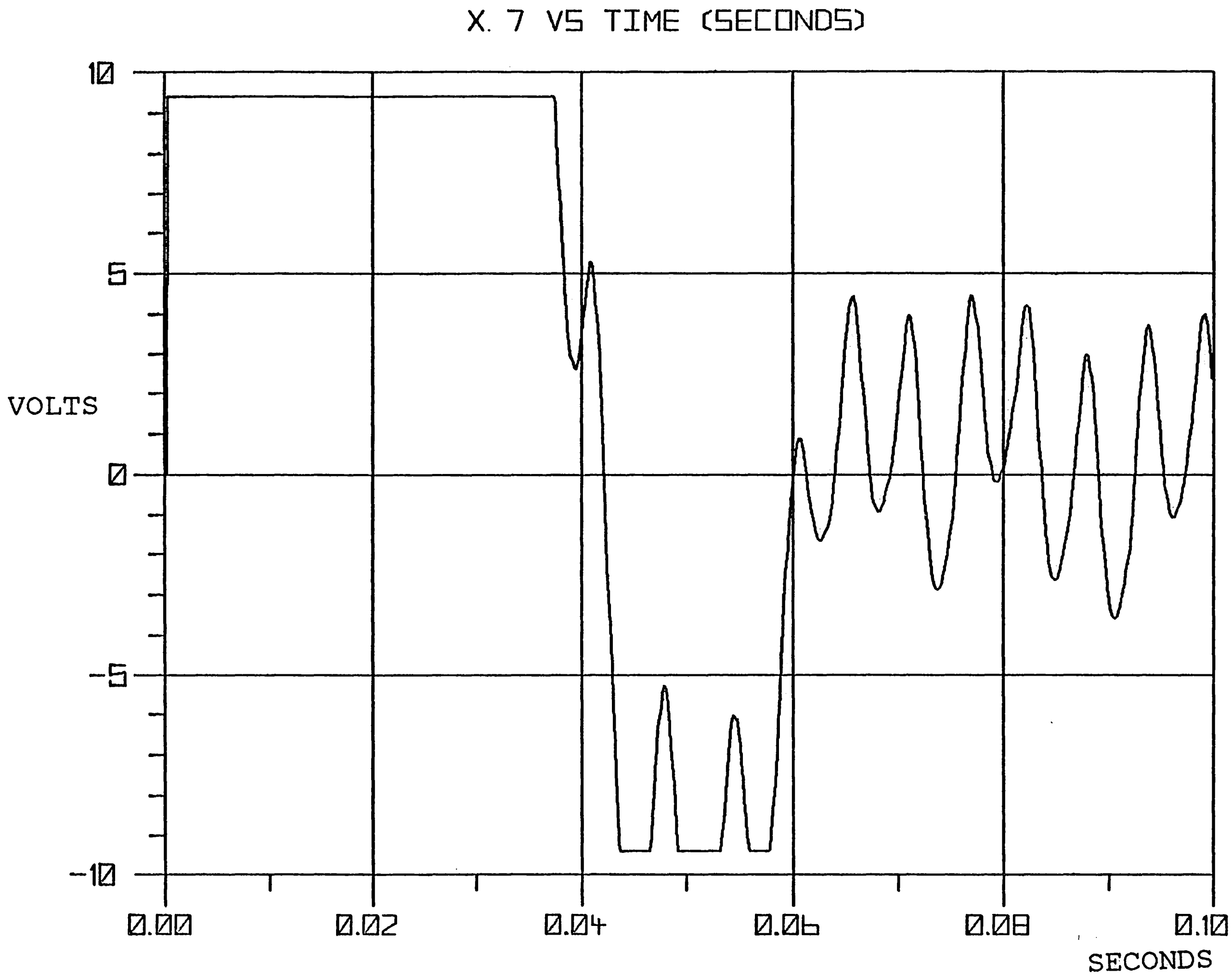


Figure 5.12b Simulated Rate Loop Output at 1000 RPM (ccw)

(using empirical values)

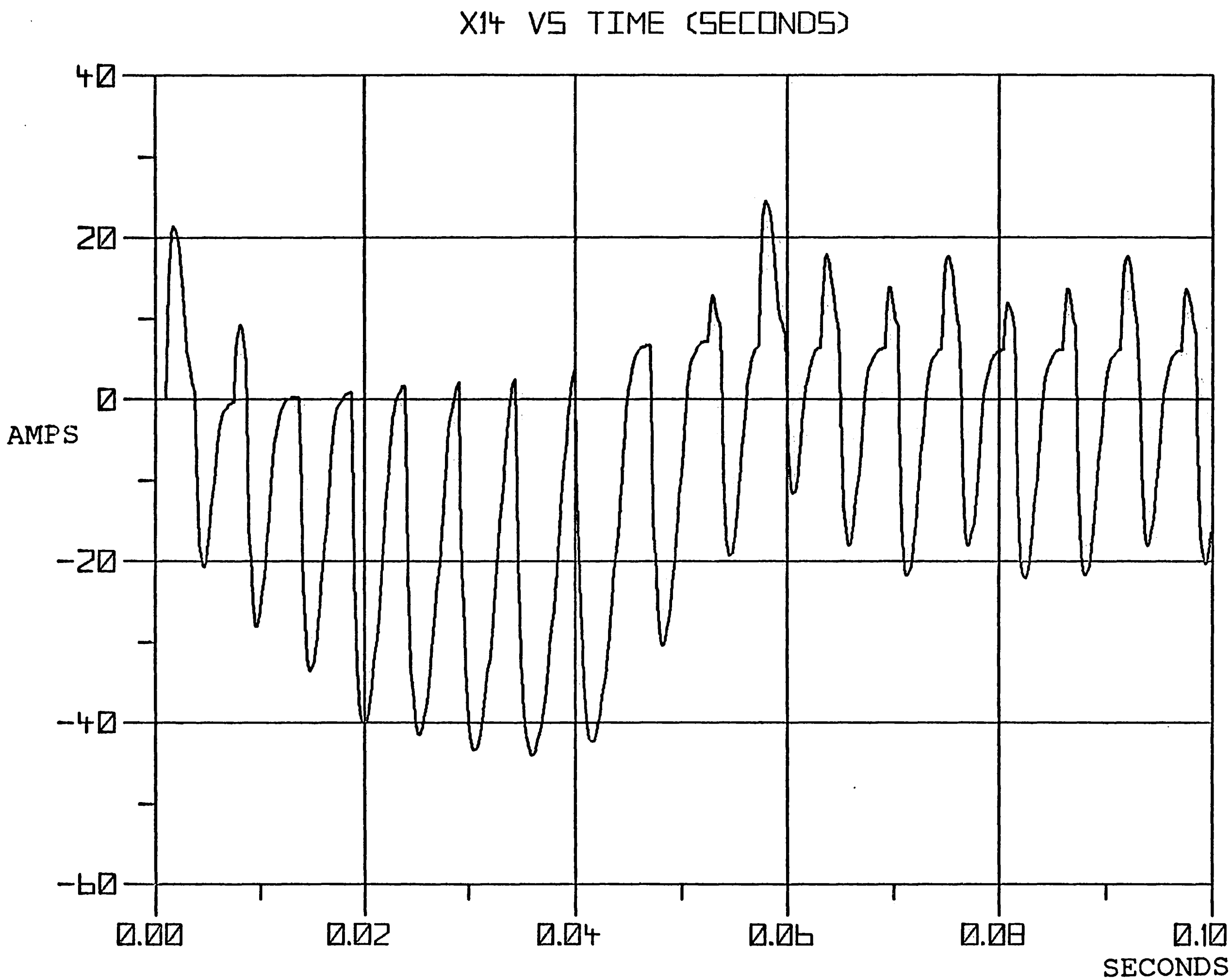


Figure 5.12c Simulated Motor Current at 1000 RPM (ccw)

(using empirical values)

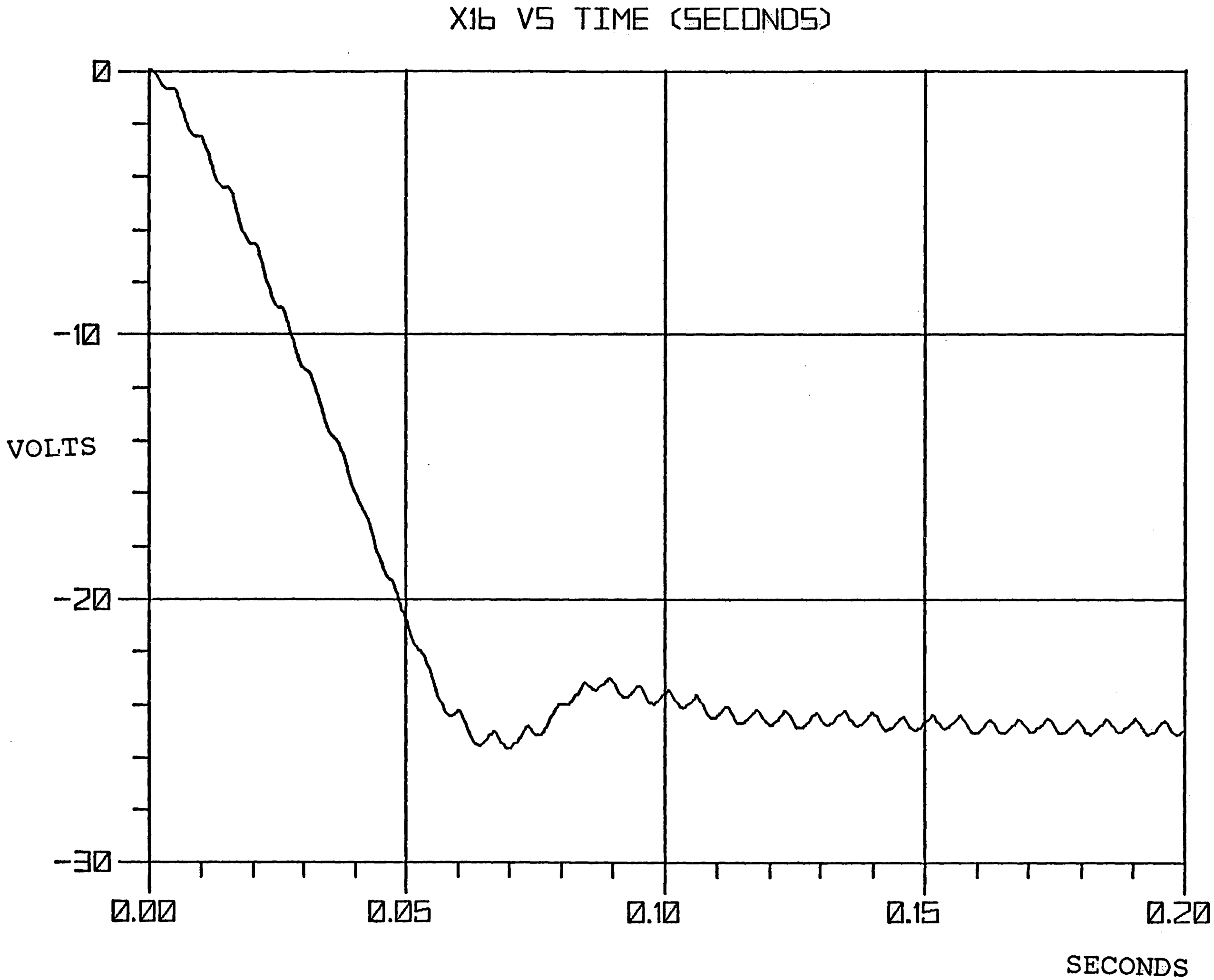


Figure 5.13a Simulated Tachometer Output at 2000 RPM (cw)

(using empirical values)

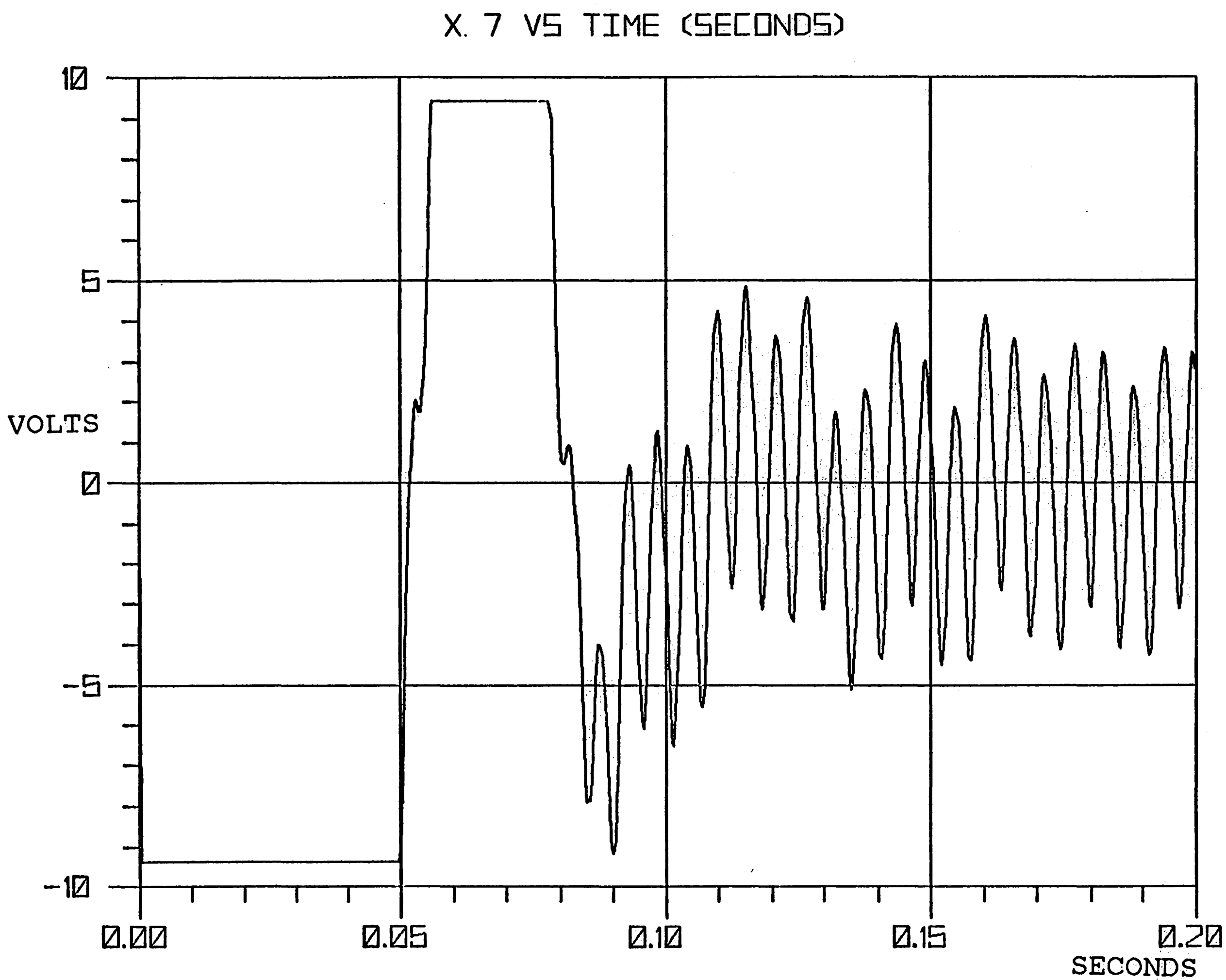


Figure 5.13b Simulated Rate Loop Output at 2000 RPM (cw)

(using empirical values)

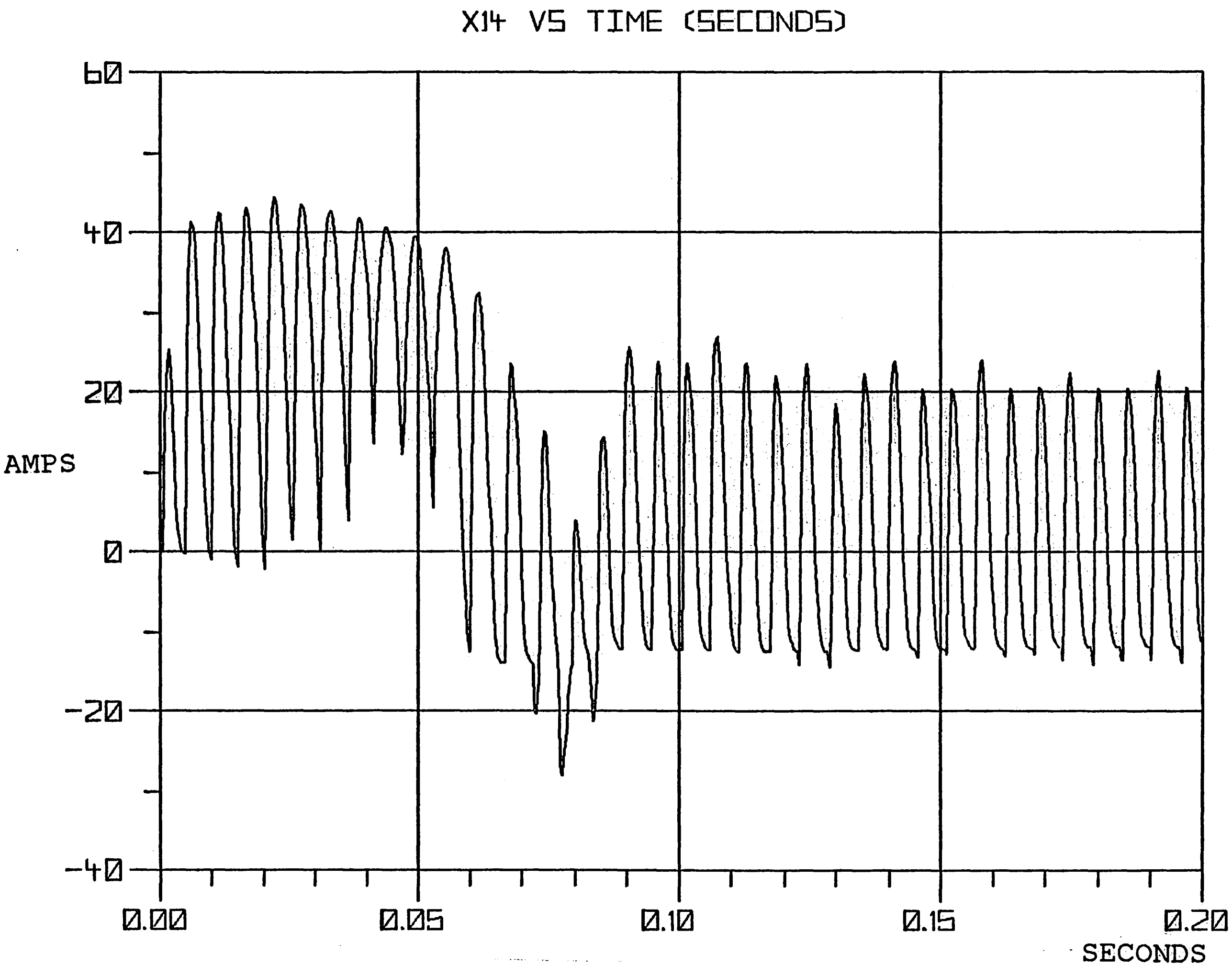


Figure 5.13c Simulated Motor Current at 2000 RPM (cw)

(using empirical values)

Chapter VI

RECOMMENDATIONS

The results of this work offer many areas for further study. First an accurate means for determining the suitable DC motor parameters must be developed if this simulation package is to be used with any other motor. From a modeling aspect, the system's power stage model is an area where more work could be done. As noted in section 4, the power stage output is now being simulated as the sum of a DC component and a set of voltage pulses (to simulate the SCRs switching on and off). Perhaps, this stage could be modeled as one simulation device. Also, the fact that the load on the power stage is a resistive and inductive load causes the current in the SCRs to lag behind the voltage in each device. Because of this lagging current, the SCRs do not switch off exactly as the phase voltage crosses zero, but at some later time. The actual tachometer signal ripple should also be examined to determine how much of the signal is due to the magnetic pickup of the tachometer from the pulsating motor current. Another area related to the power stage where further study could be done is the power source itself. At the present, the drive system's power is supplied from a 3 phase delta to wye transformer. The harmonics from this type of connection are small, but not transparent, therefore, the actual power supply voltages are not smooth sinusoids as assumed. Perhaps this could be added in when more work is done on the power stage model. The modeling of integrators that have leakage currents would aid in analyzing the system in cases where this leakage current cannot be neglected. An op amp with an input open would be such a case. Perhaps this topic should be studied further.

Another subject to be addressed in the future is the application of microprocessors to this type of drive system. As discussed in the introduction, microprocessors have been placed at various places in the control loop of drive systems. The results obtained in the modeling process have revealed the time constants in the control loops for this drive. This is very important information to have when applying microprocessor control because, as previously mentioned, the microprocessor computation time cannot be neglected when placing the processor in a drive system control loop [5]. For this system, it should be interesting to examine the performance of the system with a microprocessor Current Loop or a microprocessor Pulse Generator module. Also, some type of microprocessor - adaptive control should provide a topic for further study.

The work done by Mounfield [30], in 1980, showed that analog adaptive control can be applied to a DC drive.

From a modeling and simulation standpoint, it should be very rewarding to study how well this model of the DC drive system simulates the drive system's perfomance under loaded conditions. In this case, the lagging current produced by the DC motor may not be able to be neglected, if realistic simulation results are desired. Finally, another interesting problem would be to simulate the microprocessor in the control loop and to compare the actual drive system performance to the simulated results.

Appendix A

EQUIPMENT

Servo Amplifier	Inland Motors Model TPAR-3330
Motor-Tachometer	Inland Motors Model TT-2953B
Function Generator	Hewlett-Packard Model 3310A VPI&SU No. 3514
DC Power Supply	Hewlett-Packard Model 6226B VPI&SU No. 3503
Oscilloscope	Tektronix Type 561A Serial No. 015654
Digital Meters (2)	Fluke Model 8000A VPI&SU No. 58363 VPI&SU No. 58377
Oscilloscope Camera	Tektronix Model C-5 VPI&SU No. 3240
Storage Scope	Tektronix Model 5103N VPI&SU No. 3262

Appendix B

SIMULATION ROUTINE

Appendix B contains the Fortran Code for the Digital Simulation Program (DSIM). This is the routine that was used to simulate the drive system response.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                            C
C         PROGRAM NAME      DSIM FORTRAN A1                  C
C                                                            C
C      THIS IS A STATE TRANSITION PROGRAM DEVELOPED          C
C      TO SIMULATE SYSTEM RESPONSES                          C
C                                                            C
C      DEVELOPED FOR CONTROL SYSTEMS RESEARCH                C
C      VIRGINIA  TECH                                        C
C                                                            C
C      BY G. MANDEL DUDLEY                                   C
C                                                            C
C         REVISION 8      APRIL 25,1983                      C
C              JUMP MATRIX                                   C
C              NONLINEAR SIMULATION                          C
C                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C MAIN PROGRAM
C
C TYPE STATEMENTS
C
      REAL ID(25,25),A(25,25),TEMP(25,25),THETA(25,25)
     *,X(25),C(25),OLD(25,25),TIME(1500),PHI(25,25)
     *,N(25,7),LGSTAK(100),XSVE(1500,25),JUMP(25,25)
     *,RAMP(6),SCR(6),FLAG(6),TP(6)
C
      INTEGER OUT,SIZE,CODE(25),BUFFER,TERM,STORE,ANS,
      Integer Itick(6)
C
C
C DATA STATEMENTS
C
C BUILD IDENTITY MATRIX
      DATA  MAX/25/,MDIM/25/,OLD/625*0./,ID/625*0.0/
C     INTIALIZE STATE TRANSITION MATRIX
      DATA PHI/625*0.0/,TEMP/625*0./,IFLAG/0/,JUMP/625*0.0/
      DATA CODE/25*1/
CC
C DEFINE I/O DEVICE NUMBERS
      BUFFER = 20
      IN = 31
      OUT = 32
      STORE = 33
      TERM = 9
C
C
      WRITE(TERM,678)
 678  FORMAT(10X,'BEGIN SIMULATION')
C
C  READ PRINT INTERVAL
```

```
C
      READ(IN,*) IPRT
C
C READ MATRIX SIZE
C
      READ(IN,1000) SIZE
C
C  BUILD IDENTITY MATRIX AND STANDARD JUMP MATRIX
C
      DO 19 K=1,SIZE
      JUMP(K,K) = 1.0
  19  ID(K,K)=1.0
C     READ THE A MATRIX
C
      READ(IN,1010)((A(I,J),J=1,SIZE),I=1,SIZE)
C
C READ THE JUMP MATRIX
C
      READ(IN,1010) ((JUMP(I,J),J=1,SIZE),I=1,SIZE)
C
C COMPARE JUMP MATRIX TO IDENTITY MATRIX
SET UP MULTIPLICATION CODE
C
      DO 36 I =1,SIZE
         DO 36 J = 1,SIZE
  36  IF(JUMP(I,J) .NE. ID(I,J)) CODE(I) = 2
C
C     READ INTIAL CONDITION VECTOR
C
      READ(IN,1010) (X(I),I=1,SIZE)
C
C     READ SAMPLE RATE T
      READ(IN,1010) T
CC
C     READ THE NUMBER OF SAMPLES DESIRED
      READ(IN,*) NSAMP
C
C READ NONLINEAR DEFINITION MATRIX N
C
      READ(IN,1010) ((N(I,J),J=1,7),I=1,SIZE)
C
C SET INTIAL CONDITIONS FOR DUAL CONVERTER
C
      IP4MS = 0.0004/T
      I6DEG = 1.0/(3600.*T)
      I240 = 1.0/(90.*T) + 0.5
      BIAS = -2.5
      DO 1310 I=1,6
      RAMP(I) = -15.0
      SCR(I) = 0.0
```

```
 1310 FLAG(I) = 5.0
      ITICK(1) = 0.0
      ITICK(4) = -1.0/(120.*T) -.5
      ITICK(2) =-(1.0/(180.*T)) - 0.5
      ITICK(5) = 1.0/(360.*T) + .5
      ITICK(3) = 1.0/(180.*T)+ .5
      ITICK(6) =-(1.0/(360.*T)) -.5
      BIAS = -2.5
      IPRIOD = 1.0/(60.*T) + .5
C
C CHECK FOR TRANSPORT LAG
C
      DO 23 I = 1,SIZE
      ITYPE = N(I,1)
C
C SAVE INTIAL VALUES IN TRANSPORT LAG STACK.
C
         LIST = N(I,2)
         DO 21 J = 1, LIST
  21     LGSTAK(J) = X(I)
  23  CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                           C
C THIS ROUTINE FORMS THE STATE TRANSITION MATRIX PHI        C
C                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C CHECK FLAG TO DETERMINE WHETHER THE PHI MATRIX NEEDS TO BE
C RECALCULATED.    0 = NO     1 = YES
C
      READ(IN,1041) ANS
      WRITE(TERM,1401)
 1401 FORMAT(2X,'PAST DATA')
C
C IF ANS = 1, CALCULATE PHI
C
      IF(ANS.EQ.1) GO TO 220
C
C NO NEED TO RECALCULATE PHI, JUST READ IT IN.
C
      DO 219 I=1,SIZE
 219  READ(STORE,117) (PHI(I,J),J=1,SIZE)
C
C PHI HAS BEEN READ, CONTINUE TO SIMULATION.
C PRINT OUT PHI MATRIX
C
      WRITE(OUT,500)
```

```
      DO 218 I=1,SIZE
 218  WRITE(OUT,117) (A(I,J),J=1,SIZE)
C
      GO TO 50
C
C
C
 220  DO 22 I=1,SIZE
      DO 24 J=1,SIZE
 24   TEMP(I,J) = A(I,J)
  22  CONTINUE
      DO 10 I=1,SIZE
         DO 20 J=1,SIZE
  20     PHI(I,J) = ID(I,J)
  10  CONTINUE
C
C CALL ROUTINE TO FORM T*A
         WRITE(OUT,500)
 500  FORMAT(10X,'A MATRIX')
      DO 88 I=1,SIZE
  88  WRITE(OUT,117) (A(I,J),J=1,SIZE)
C
      CALL MULC(T,TEMP,SIZE,MDIM)
C
C ADD I + TA
C
      CALL ADDM(PHI,TEMP,SIZE,MDIM)
C
C     BEGIN SUMMATION LOOP
C
      NTM=0
      DO 30 K=2,MAX
      CALL  MULA (A,TEMP,SIZE,MDIM)
C
C      INCREMENT LOOP COUNTER
      NTM = NTM + 1
C
C     FORM NEXT TERM
      CONS = T/K
C
C     CALL ROUTINE TO MULTIPLY BY A CONSTANT
C
      CALL MULC(CONS,TEMP,SIZE,MDIM)
C
C     ADD NEXT TERM
  117 FORMAT(1X,8F13.4)
C
      CALL ADDM (PHI,TEMP,SIZE,MDIM)
C
C CHECK FOR TERMINATION OF SUMMATION
```

```
C
      CALL CHECK (PHI,IFLAG,NTM,SIZE,OLD)
C
      IF(IFLAG.EQ.1) GO TO 50
C
C
C      SERIES HAS NOT CONVERGED ADD ANOTHER TERM
C
  30  CONTINUE
C
C
C     CLEAR OUTPUT PAGE
C
 50   WRITE(OUT,66)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                            C
C     OUTPUT HEADINGS                                        C
C                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      WRITE(OUT,71)
  71  FORMAT(/10X,'STATE TRANSITION MATRIX - PHI IS ')
      DO 76 I=1,SIZE
C
C SAVE PHI MATRIX
C
      IF(ANS.EQ.1) WRITE(STORE,117) (PHI(I,J),J=1,SIZE)
  76  WRITE(OUT,117) (PHI(I,J),J=1,SIZE)
      WRITE(OUT,73)
 73   FORMAT(/10X,'STATE VARIABLE DEFINITIONS:')
      WRITE(OUT,176)
      DO 177 I = 1,SIZE
 177  WRITE(OUT,178)  I,X(I)
 178  FORMAT(10X,'X',I2,'=',F15.8)
 176  FORMAT(/10X,'THE INTIAL CONDITION VECTOR IS:',/)
      WRITE(OUT,175)
 175  FORMAT(/10X,' NONLINEAR DESCRIPTION MATRIX    N  ')
      DO 167 I =1,SIZE
 167  WRITE(OUT,717) (N(I,J),J=1,7)
 717  FORMAT(4X,7F15.6)
      WRITE(OUT,718)
 718  FORMAT(/10X,'THE JUMP MATRIX  J')
      DO 719 I = 1,SIZE
 719  WRITE(OUT,117) (JUMP(I,J),J=1,SIZE)
      WRITE(OUT,66)
      TM = 0.00
      ICNT=0
      WRITE(OUT,27) T
```

```
 27   FORMAT(/10X,'SOLUTION USING A STEP SIZE =',F14.7,/
     *10X,'  THE RESULTS ARE PRINTED BELOW (OPTIONAL)  .',/)
C
C
C
C*************************************************************
C                                                            *
C                      START SIMULATION                      *
C                                                            *
C****************************************/*******************
C
      IT = 1
C  WRITE INTIAL CONDITIONS
C***  WRITE(OUT,1040) TM
C  THIS SECTION MAY BE PUT IN IF ACTUAL DATA IS NEEDED
C
C
C
C
C**** WRITE(OUT,1042) (X(J),J=1,SIZE)
      TIME(IT) = 0.0
      DO 601 IC=1,SIZE
 601  XSVE(1,IC) = X(IC)
C
      DO 60 K=1,NSAMP
C CALL MATRIX MULTIPLY
C
C MULTIPLY BY STATE TRANSITION MATRIX (PHI)
C
      CALL MULAV(PHI,X,SIZE,MDIM,C)
C
C MULTIPLY BY JUMP MATRIX
C
C
      CALL JMULAV (JUMP,X,SIZE,MDIM,CODE)
C
C SIMULATE SAMPLING PROCESS FOR NONLINEAR ELEMENTS
C
      IPT = 0
 699  IPT = IPT + 1
      IF(IPT.GT.SIZE) GO TO 698
      DO 700 LOOP=IPT,SIZE
      NLNEAR = N(LOOP,1)
         GO TO (701,702,703,704,705,706,707,708,709,710,711
     * ,712,713),NLNEAR
  700 CONTINUE
 698  CONTINUE
C
C
C     INCREMENT TIME
```

```
      TM = TM + T
C
      ICNT=ICNT + 1
C
C     SAVE POINTS FOR OUTPUT
C
      IF(ICNT.LT.IPRT) GO TO 59
         ICNT=0
      IT = IT + 1
      TIME(IT) = TM
C**** WRITE(OUT,1040) TM
C**** WRITE(OUT,1042) (X(J),J=1,SIZE)
C
C  SAVE DATA FOR PLOT ROUTINE
C
      DO 591 IX=1,SIZE
 591  XSVE(IT,IX) = X(IX)
C
 59   CONTINUE
         GO TO 60
C
C*************************************************************
C                                                            *
C           NONLINEAR DEFINTION SECTION                      *
C                                                            *
C*************************************************************
C
C THIS SECTION DEFINES THE FUNCTIONS OF THE
C NONLINEAR ELEMENT OF THE SYSTEM
C
C
C************   TYPE 1   CONTINUOS ELEMENT   **************
C
  701 GO TO 699
C
C
C*********   TYPE 2   LINEAR AMPLIFIER WITH SATURATION ***
C
 702  IF(X(LOOP).LE.N(LOOP,2)) X(LOOP) = N(LOOP,2)
      IF(X(LOOP).GE.N(LOOP,3)) X(LOOP) = N(LOOP,3)
      GO TO 699
C
C********   TYPE 3   PIECEWISE LINEAR DEVICE   **********
C
 703  VALUE = ABS(X(LOOP))
      IF(VALUE .LE. N(LOOP,2)) GO TO 699
      IF(VALUE .GE. N(LOOP,4)) GO TO 7031
      RESULT = (VALUE-N(LOOP,2))*N(LOOP,3) + N(LOOP,2)
      X(LOOP) = SIGN(RESULT,X(LOOP))
      GO TO 699
```

```
 7031 IF(VALUE.GE.N(LOOP,6)) GO TO 7032
      RESULT =(VALUE-N(LOOP,4))*N(LOOP,5) + N(LOOP,2)
     ; (N(LOOP,4)-N(LOOP,2))*N(LOOP,3)
      X(LOOP) = SIGN(RESULT,X(LOOP))
      GO TO 699
 7032 RESULT = (N(LOOP,6)-N(LOOP,4))*N(LOOP,5)
     ; + (N(LOOP,4)-N(LOOP,2))*N(LOOP,3) +N(LOOP,2)
 7033 X(LOOP) = SIGN(RESULT,X(LOOP))
      GO TO 699
C
C**    TYPE 4    POLYNOMIAL  MAXIMIUM FIFTH ORDER    ********
C
 704  X(LOOP)=N(LOOP,2) + N(LOOP,3)*X(LOOP)
     ; + N(LOOP,4) * X(LOOP)**2
     ; + N(LOOP,5) * X(LOOP)**3
     ; + N(LOOP,6) * X(LOOP)**4
     ; + N(LOOP,7) * X(LOOP)**5
      GO TO 699
C
C
C**********    TYPE 5     RELAY DEADBAND       **************
C
  705 IF(X(LOOP).LT.0)  GO TO 7051
C CHECK FOR POSITIVE DEADBAND
      IF(X(LOOP).LT.N(LOOP,3))  X(LOOP) = 0.0
      IF(X(LOOP).GE.N(LOOP,3))  X(LOOP) = N(LOOP,4)
C RETURN
      GO TO 699
C CHECK NEGATIVE DEADBAND
 7051 IF((ABS(X(LOOP)).LT.ABS(N(LOOP,5)))) X(LOOP)=0.0
      IF((ABS(X(LOOP)).GE.ABS(N(LOOP,5)))) X(LOOP)=N(LOOP,6)
C RETURN
      GO TO 699
C
C***************     TYPE 6   MULTIPLE SAMPLER    ***********
C
C CHECK THE DOWN COUNTER, SAMPLE EVERY TM/T SECONDS.
  706 IF(N(LOOP,6).GT.0.0) GO TO 7061
C COUNTER EQUAL TO ZERO ?   (TIME TO TAKE A SAMPLE)
C NEGATIVE SATURATION ?
C
      IF(X(LOOP).LE.N(LOOP,4)) X(LOOP) = N(LOOP,4)
C POSITIVE SATURATION ?
      IF(X(LOOP).GE.N(LOOP,3)) X(LOOP) = N(LOOP,3)
C RESET COUNTERS
      N(LOOP,6) = N(LOOP,5) - 1.0
C RETURN
      GO TO 699
C DECREMENT COUNTER
 7061 N(LOOP,6) = N(LOOP,6) - 1.0
```

```
C  KEEP SAME VALUE FOR STATE VARIABLE
      X(LOOP) = C(LOOP)
C RETURN
      GO TO 699
C
C****************      TYPE 7    FINITE SAMPLER    **********
C
C IS THE SAMPLER CLOSED ?
C
  707 IF(N(LOOP,2).EQ.0.0)  GO TO 7072
C NO, IS THE COUNTER = 0 ?
      IF(N(LOOP,7).EQ.0.0)   GO TO 7071
C NO, DECREMENT COUNTER
      N(LOOP,7) = N(LOOP,7) - 1.0
C SAMPLER IS OPEN.
      X(LOOP ) = 0.0
C RETURN
      GO TO 699
C
C SET SAMPLER CLOSED
 7071 N(LOOP,2) = 0.0
      IF(X(LOOP) .LE. N(LOOP,4))  X(LOOP) = N(LOOP,4)
      IF(X(LOOP) .GE. N(LOOP,3))  X(LOOP) = N(LOOP,3)
C RESET TIMER, KEEP SAMPLE.
      N(LOOP,7) = N(LOOP,5) - 1.0
      GO TO 699
C
C
C****************** SAMPLER CLOSED ************************
C
C IS COUNTER = 0 ?
 7072 IF(N(LOOP,7).EQ.0.0)  GO TO 7073
C NO, TAKE SAMPLE WITH SATURATION LIMITS
      IF(X(LOOP) .LE. N(LOOP,4))  X(LOOP) = N(LOOP,4)
      IF(X(LOOP) .GE. N(LOOP,3))  X(LOOP) = N(LOOP,3)
      N(LOOP,7) = N(LOOP,7)-1.0
      GO TO 699
C
C SET SAMPLER OPEN.
C
 7073 N(LOOP,2) = 1.0
      N(LOOP,7) = N(LOOP,6) - 1.0
C SAMPLE = 0.0
      X(LOOP) = 0.0
C RETURN
      GO TO 699
C
C
C**************    TYPE 8    TRANSPORT LAG      ************
C
```

```
C SAVE PRESENT SAMPLE
C
  708 TDATA = X(LOOP)
C
C TAKE DATA FROM TOP OF STACK
C
      X(LOOP) = LGSTAK(1)
C
C MOVE ALL DATA UP BY 1
C
      ICNTR = N(LOOP,2) - 1.0
      DO 7081 L = 1, ICNTR
C
 7081 LGSTAK(L) = LGSTAK(L+1)
C
C BRING IN NEW DATA
C
      LGSTAK(ICNTR+1) = TDATA
C
C RETURN
C
      GO TO 699
C
C
C***************     TYPE 9   SUMMING JUNCTION   ***********
C
C
 709  CONTINUE
C
C THIS ROUTINE SIMULATES A SUMMING JUNCTION WITH FIVE INPUTS
C   *** OUTPUT LIMIT = +12.5,-12.5 ***
C
      X(LOOP) = 0.0
      INDEX = N(LOOP,2)
      IF(INDEX.GT.5) WRITE(TERM,7092) LOOP
 7092 FORMAT(10X,'TOO MANY INPUTS TOP SUMMING JUNCTION ',I2)
      IF (INDEX .GT. 5) STOP
      DO 7091 I=1,INDEX
 7091    X(LOOP) = X(LOOP) + X(I)

C
C TEST FOR SATURATION
C
      TEST = X(LOOP)
      IF(ABS(TEST).GT.12.5) X(LOOP) = SIGN(12.5,X(LOOP))
      GO TO 699
C
C
C
C***********   TYPE 10 CONTROLLED LIMIT SATURATING DEVICE **
```

```
C
C IF CONTROL INPUT EQUALS ZERO, LIMITS BASED ON L1,L2
C
 710  SWITCH = ABS(X(N(LOOP,6)))
      IF ( SWITCH .GE. N(LOOP,7)) GO TO 7011
C
C...CONTROL INPUT EQUAL ZERO, ... USE L1,L2
C
      IF(X(LOOP) .LE. N(LOOP,2))  X(LOOP) = N(LOOP,2)
      IF(X(LOOP) .GE. N(LOOP,3))  X(LOOP) = N(LOOP,3)
C
C RETURN
C
      GO TO 699
C
C... CONTROL INPUT NONZERO ... LIMITS BASED ON L3,L4
C
 7011 IF(X(LOOP) .LE. N(LOOP,4))  X(LOOP) = N(LOOP,4)
      IF(X(LOOP) .GE. N(LOOP,5))  X(LOOP) = N(LOOP,5)
C
C RETURN
C
      GO TO 699
C
C
C*************        TYPE 11  RECTIFIER            ********
C
C IF INPUT LESS THAN VFD, THEN VOUT = 0.
C
  711 IF(X(LOOP) .LT. (N(LOOP,2)))  X(LOOP) = 0.0
C
C IF VIN GREATER THAN OR EQUAL TO VFD, VOUT=VIN-VFD
C
      IF(X(LOOP) .GE. (N(LOOP,2))) X(LOOP) = X(LOOP)-N(LOOP,2)
C
C RETRUN
C
      GO TO 699
C
C***************   TYPE 12 STRAIGHT LINE APPROXIMATION *****
C
C THIS DEVICE GENERATES A NONLINEAR GAIN MADE FROM LINES
C OF THE FORM Y = MX + B
C
C FIND  OPERATING REGION
C
 712  IF(X(LOOP) .LT.-1.13) GO TO 7121
C
C LINE 1
C
```

```
      X(LOOP) =  62.583* X(LOOP) + 69.031
C     IF(X(LOOP) .GT. 132.) X(LOOP) = 132.
C RETURN
      X(LOOP) = X(LOOP) /2.0
      GO TO 699
C
C
C LINE 2
C
 7121 X(LOOP) = 63.05 * X(LOOP) + 78.365
C     IF(X(LOOP).LT. -132.) X(LOOP) = -132.
C RETURN
C
      X(LOOP) = X(LOOP)/2.0
      GO TO 699
C
C****************** TYPE 13  DUAL CONVERTER ****************
C
C
C THIS IS A ROUTINE TO SIMULATE A 3 PHASE DUAL CONVERTER
C
C
C SET INTIAL CONDITIONS
C
C
C ERROR INPUT NUMBER
C
 713  IRF = N(LOOP,3)
C
C
         DO 800 I=1,6
         IF(FLAG(I) .LT. 0.0) GO TO 820
C
C CHECK FOR SIX DEGREE PHASE SHIFT
C
         IF(ITICK(I) .GT. I6DEG) GO TO 810
C
C DO NOT START RAMP
C
         RAMP(I) = -15.0
C
C INCREMENT AND RETURN
C
         GO TO 822
C
C PASSED 6 DEGREE DELAY, NOW CHECK FOR RAMP1 OR 2
C
 810  IF(ITICK(I) .GT.IP4MS) GO TO 820
C
C USE RAMP
```

```
C
         RAMP(I)=15000.0*(T*(ITICK(I)-0.0))-15.0
         GO TO 822
 820     IF(ITICK(I).GE. I240) GO TO 830
         RAMP(I) = 1250.0*(T*(ITICK(I)-0.0+0.0))-9.0
C
C INCREMENT AND RETURN
C
         IF(FLAG(I) .LT. 0.0) GO TO 822
         GO TO 822
 830     RAMP(I) = -15.0
C
C GONE THROUGH 1 CYCLE YET ? IF YES RESET COUNTER.
C
 822  IF(ITICK(I) .GE. IPRIOD) ITICK(I) = -1
         IF(ITICK(I) .EQ. -1) RAMP(I) =-15.
         IF(ITICK(I) .EQ. -1) FLAG(I) = 5.0
         IF(ITICK(I) .EQ. -1) GO TO 800
C
C TEST FOR FIRING PULSES
C
 7000  SCR(I)  = 0.0
         IF(ITICK(I) .GE. I240) GO TO 800
         IF(FLAG(I).LT. 0.0) GO TO 1801
         IF(I.LE.3) TP(I)=BIAS -X(IRF)+ .5
         IF(I.GE.4) TP(I) = X(IRF) + 0.5
         IF(RAMP(I) .GT. TP(I)) FLAG(I) = -5.0
         IF(FLAG(I) .GT. 0.0) GO TO 800
C
C SEND FIRING PULSES
C
 1801    RAMP(I) = TP(I) -1.0
C
C FIRE THE CORRECT SCR
C
         GO TO (801,802,803,804,805,806),I
 801  SCR(1) = 275.*(SIN(377.*ITICK(I)*T))
      IF(SCR(1).LT. 0.0) SCR(1) =0.0
      GO TO 800
 802  SCR(2) = 275.*(SIN(377.*ITICK(1)*T - 2.0944))
      IF(SCR(2) .LT. 0.0 ) SCR(2) = 0.0
      GO TO 800
 803  SCR(3) = 275.*(SIN(377.*ITICK(1)*T + 2.0944))
      IF(SCR(3) .LT. 0.0) SCR(3) = 0.0
      GO TO 800
 804  SCR(4) = 275.*(SIN(377.*ITICK(1)*T))
      IF(SCR(4) .GT. 0.0) SCR(4) = 0.0
      GO TO 800
 805  SCR(5) = 275.*(SIN(377.*ITICK(1)*T -2.0944))
      IF(SCR(5) .GT. 0.0) SCR(5)= 0.0
```

```
      GO TO 800
 806  SCR(6) = 275.*(SIN(377.*ITICK(1)*T + 2.0944))
      IF(SCR(6) .GT. 0.0 ) SCR(6) = 0.0
      GO TO 800
 800  ITICK(I) = ITICK(I) + 1
      X(LOOP) = 0.0
      DO 1702 KC=1,6
 1702 X(LOOP) = X(LOOP) + SCR(KC)
      X(LOOP) = X(LOOP)/2.0
      GO TO 699
C***************   END OF NONLINEAR DEFINITIONS ***********
C
C
  60  CONTINUE
C
C  WRITE DATA TO PLOT BUFFER
C
      WRITE(BUFFER,*) SIZE
      WRITE(BUFFER,*) IT
      WRITE(BUFFER,*) (TIME(I),I=1,IT)
      DO 600 IP =1,SIZE
      DO 600 JIP=1,IT
 600  WRITE(BUFFER,1010) XSVE(JIP,IP)
C
      WRITE(TERM,679)
 679  FORMAT(10X,'END SIMULATION')
C
C FORMAT STATEMENTS
C
  66  FORMAT(//)
 1000 FORMAT(I2)
 1001 FORMAT(I3)
 1010 FORMAT(F15.6)
C1020 FORMAT(21X,'X',I1,8(12X,'X',I1),/)
 1040 FORMAT(1X,'T=',F09.5)
 1042 FORMAT(8(2X,F11.5))
 1041 FORMAT(I1)
      STOP
      END
C
C END MAIN PROGRAM
C
      SUBROUTINE MULC (T,TEMP,SIZE,MDIM)
C
C THIS IS A ROUTINE TO MULTIPLY A SQUARE MATRIX BY A CONSTANT
C
C T = CONSTANT   TEMP = MATRIX NAME    SIZE  = MATRIX SIZE
C
      INTEGER SIZE
      REAL T,TEMP(MDIM,MDIM)
```

```
C
C     PERFORM MULTIPLICATION
C
      DO 10 K = 1,SIZE
         DO 20 L = 1,SIZE
  20     TEMP(K,L) = T*TEMP(K,L)
  10  CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE ADDM(PHI,TEMP,SIZE,MDIM)
C
C     THIS IS A ROUTINE TO ADD TWO SQUARE MATRICES.
C     THE RESULT IS STORED IN PHI.  THE ORIGINAL CONTENTS
C     OF PHI ARE LOST.
C
      INTEGER SIZE
      REAL PHI(MDIM,MDIM),TEMP(MDIM,MDIM)
      DO 20 K = 1,SIZE
         DO 20 L=1,SIZE
  20  PHI(K,L) = PHI(K,L) + TEMP(K,L)
      RETURN
      END
C
C
C
      SUBROUTINE MULA(A,TEMP,SIZE,MDIM)
C
C     THIS IS A ROUTINE TO MULTIPLY TWO SIZE X SIZE
C     SQUARE MATRICES.  THE RESULT IS STORED IN
C     TEMP (ORIGNAL DATA LOST)
C
      INTEGER SIZE
      REAL A(MDIM,MDIM),TEMP(MDIM,MDIM),C(25,25)
      IOUT = 32
C
C CALL IMSL ROUTINE TO DO MULTIPLICATON
C
      CALL VMULFF(A,TEMP,SIZE,SIZE,SIZE,MDIM,MDIM,C,MDIM,IER)
C
C     CHECK THE RETURN CODE
C
      IF(IER.NE.0) WRITE(IOUT,60) IER
  60  FORMAT(/20X,'ERRROR IN VMULFF = ',I5,/)
      IF(IER.NE.0) STOP
      DO 10 K=1,SIZE
         DO 20 L=1,SIZE
  20     TEMP(K,L) = C(K,L)
```

```
   10 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE CHECK (NEW,IFLAG,NTM,SIZE,OLD)
C
C     THIS IS A ROUTINE TO TEST WHETHER THE TRANSITION
C     MATRIX HAS CONVERGED TO THE TOLERANCE SPECIFIED
C     IN THE EQUATE STATEMENT.  THE FLAG WORD IFLAG = 0
C     IF THE MATRIX HAS NOT CONVERGED, IFLAG=1 IF IT HAS
C     IF IT HAS CONVERGED
C
C
      INTEGER SIZE,NTM,IFLAG
      REAL NEW(25,25),OLD(25,25)
C
C SET STATUS FLAG TO NOT READY
      IFLAG = 0
      IF(NTM.EQ.1) GO TO 99
C
C IF THIS IS THE FIRST TIME-STORE DATA IN BUFFER
C
C     SET UP TOLERRANCE
      TOL = 0.0001
C
C     THIS IS A LOOP TO DETERMINE IF EVERY VALUE
C     IN THE PHI MATRIX HAS CONVERGED TO WITHIN THE
C     TOLLERANCE.  IF ANY VALUE HAS NOT CONVERGED THE LOOP
      WILL RETURN
C
      DO 20 I =1,SIZE
         DO 30 J=1,SIZE
         IF(NEW(I,J).LT.TOL) GO TO 30
         TERM=ABS((NEW(I,J)-OLD(I,J))/NEW(I,J))
         IF(TERM.GT.TOL) GO TO 99
   30    CONTINUE
   20 CONTINUE
C
C     ALL VALUES HAVE CONVERGED, SET DATA READY FLAG
C
         IFLAG = 1
C
C     RETURN
C
C     DATA NOT READY - MOVE NEW DATA TO OLD.
   99 DO 60 I=1,SIZE
      DO 60 J=1,SIZE
   60 OLD(I,J)=NEW(I,J)
```

```
      RETURN
      END
C
C
C
      SUBROUTINE MULAV(PHI,X,SIZE,MDIM,C)
C
C THIS IS A ROUTINE TO MULTPLY AN ARRAY(PHI)
C TIMES A VECTOR (X).  THE RESULT IS STORED IN X.
      INTEGER SIZE
      REAL PHI(25,25),X(25),C(25)
      IOUT = 32
      CALL VMULFF (PHI,X,SIZE,SIZE,1,MDIM,MDIM,C,MDIM,IER)
C
C CHECK RETURN VALUE
      IF(IER.NE.0) WRITE(IOUT,60) IER
 60   FORMAT(/20X,'THE RETURN CODE FROM VMULFF= ',I5,/)
      IF(IER.NE.0) STOP
C
C EVERYTHING OK CONTINUE
      DO 58 IJ=1,SIZE
 58   X(IJ) = C(IJ)
C STORE VALUE IN X
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                           C
C                  THIS IS ROUTINE JMULAV                   C
C        THIS ROUTINE HANDLES THE JUMP MATRIX CALCULATIONS C
CC  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE JMULAV(JUMP,X,SIZE,MDIM,CODE)
C
C
      INTEGER SIZE,CODE(MDIM)
      REAL JUMP(MDIM,MDIM),X(MDIM),C(25)
C
C INTIALIZE THE VECTOR
C
      DO 30 I=1,SIZE
 30   C(I) = 0.0
C
C CHECK CODE, DO MULTIPLICATION IF NOT EQUAL TO 1
C
      DO 20 I=1,SIZE
      IF(CODE(I).EQ.1) GO TO 20
         DO 10 K=1,SIZE
      C(I) = C(I) + JUMP(I,K)*X(K)
 10   CONTINUE
```

```
   20  CONTINUE
C
C UPDATE STATE VARIABLE
C
       DO 40 I=1,SIZE
       IF(CODE(I).NE.1) X(I) = C(I)
   40  CONTINUE
       RETURN
       END
```

Appendix C

GRAPHICS ROUTINE

Appendix C contains the Fortran code for the graphic output of the DSIM program (appendix b). The graphics calls can be found in the PLOT 10 AG II manual [31].

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                           C
C                        GRAPH FORTRAN                      C
C                                                           C
C THIS IS A ROUTINE TO GRAPH THE OUTPUT OF DSIM FORTRAN     C
C                   ON THE VERSATEC PLOTTER                 C
C                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C                   BY G M DUDLEY  APRIL 26,1983
C
      REAL TIME(999),XP(999),YP(999),X(999,20)
      INTEGER BUFFER,SIZE,NSAMP,ANS,YES,IPLOT(20),
     ;LABEL(25),TERM
      DATA YES/'Y'/
C
C SET UP LABELS
C
      DATA LABEL/88,32,49,32,86,83,32,84,73,77,69,32,40,
     ;83,69,67,79,78,68,83,41/
C
C SET UP DEVICE NUMBERS
C
      BUFFER = 20
      TERM = 9
      WRITE(TERM,1112)
 1112 FORMAT(/10X,'SIMULATOR OUTPUT DATA BEING READ IN ...')
C
C READ INFO FROM BUFFER
      READ(BUFFER,*) SIZE
C
C READ NUMBER OF SAMPLES TO BE PLOTTED
      READ(BUFFER,*) NSAMP
C
C READ IN TIME MATRIX
      READ(BUFFER,*) (TIME(I),I=1,NSAMP)
C
C SET UP PLOT ARRAY
C
      DO 13 I = 1,SIZE
 13   IPLOT(I) = I
C
C BRING IN STATE VARIABLES
C
      DO 10 I = 1, SIZE
         DO 10 J = 1,NSAMP
  10  READ(BUFFER,*) X(J,I)
C WRITE MESSAGE TO THE TERMINAL
C
      WRITE(TERM,2000)
```

```
 2000 FORMAT(2X,'ALL STATE VARIABLES ? (Y/N)')
      READ(TERM,1003) ANS
      IDESIR = SIZE
      IF (ANS.EQ.YES) GO TO 100
C
C
C
 1999 WRITE(TERM,2001)
 2001 FORMAT(/2X,'ENTER NUMBER OF VARIABLES DESIRED. (<20)')
      READ(TERM,*) IDESIR
      WRITE(TERM,2002)
 2002 FORMAT(/2X,'ENTER THE DESIRED STATE VARIABLE NUMBERS')
      DO 11 I = 1,IDESIR
         WRITE(TERM,2003) I
 2003 FORMAT(2X,'GRAPH NUMBER ',I1,' = ?')
      READ(TERM,*) IPLOT(I)
 11   CONTINUE
C
C DO PLOTS
C
 100  CONTINUE
      CALL INITT(240)
      XP(1) = FLOAT(NSAMP)
      YP(1) =FLOAT(NSAMP)
      DO 12 I = 1,IDESIR
         ICNT = NSAMP +1
         DO 21 JJ = 2,ICNT
         IJ = JJ-1
         XP(JJ)=TIME(IJ)
         YP(JJ)= X(IJ,IPLOT(I))
  21  CONTINUE
C
C  SET UP LABELS
C
         IF(IPLOT(I) .LT. 20) LABEL(2) = 49
         IF(IPLOT(I) .LT. 20) LABEL(3) = 48 + IPLOT(I) -10
         IF(IPLOT(I) .LT. 10) LABEL(2) = 32
         IF(IPLOT(I) .LT. 10) LABEL(3) = 48 + IPLOT(I)
         CALL BINITT
         CALL CHECK(XP,YP)
         CALL DSPLAY(XP,YP)
         CALL MOVABS(370,730)
         CALL HLABEL(21,LABEL)
         CALL TINPUT(IO)
85       CALL MOVABS(0,700)
         CALL ANMODE
         WRITE(TERM,60)
 60      FORMAT(5X,'WANT A HARD COPY OF THIS GRAPH? <Y/N>')
         READ(TERM,70) IRPLY
 70   FORMAT(A1)
```

```
            IF(IRPLY .EQ. YES) CALL PLSAVE
         CALL MOVABS(0,45)
         WRITE(TERM,75)
75       FORMAT('WANT TO ZOOM ON A PORTION OF THIS DRAWING?')
         READ(TERM,70) IRPLY
         IF(IRPLY.NE.YES) GO TO 80
         CALL VCURSR(ICHAR,XMIN,YMIN)
         CALL BELL
         CALL TSEND
         CALL VCURSR(ICHAR,XMAX,YMAX)
         CALL BELL
         CALL BELL
         CALL TSEND
         CALL BINITT
         CALL DLIMX(XMIN,XMAX)
         CALL DLIMY(YMIN,YMAX)
         CALL ERASE
         CALL CHECK(XP,YP)
         CALL DSPLAY(XP,YP)
         CALL MOVABS(370,730)
         CALL HLABEL(21,LABEL)
         CALL TINPUT(IO)
         GO TO 85
80       CONTINUE
            CALL ERASE
            CALL ANMODE
C
   12    CONTINUE
         WRITE(TERM,1004)
 1004    FORMAT(5X,'MORE GRAPHS ? (Y/N)')
         READ(TERM,1003) ANS
         IF(ANS .EQ. YES ) GO TO 1999
         CALL FINITT(0,0)
 1003    FORMAT(A1)
         STOP
         END
```

REFERENCES

1. Zimmerman, P. , and Gmbh, R.B., 'Electronically Commutated DC Feed Drives for Machine Tools', Drives and Controls International October/November 1982, PP. 13-19.

2. Brogan, William, L., Modern Control Theory New York, N.Y.: Quantum Publishers, 1974.

3. Safiuddin, M., and Bello, L., 'Performance Comparison Between the Voltage and the Current Controlled Converters for DC Motor Servo Drives', Power Conversion International, September 1982, PP. 34-41.

4. Konishi, T., and Kamiyama, K., 'A Performance Analysis of Microprocessor - Based Control Systems Applied to Adjustable Speed Motor Drives', IEEE Transactions on Industry Applications, Vol.IA-16, No.3, May/June 1980, PP. 378-387.

5. Tsuchiya, T., 'Suboptimal Control of a Static Scherbius Induction Motor System Using a Microprocessor', IEEE Transactions on Industry Applications, Vol.IA-16, No.5, September/October 1980, PP. 686-689.

6. Jing-Ping, J., and Marleau, R.S., 'Digitally Controlled DC Drive Motors', IEEE Transactions on Industry Applications, Vol.IA-18, No.6, November/December 1982, PP.728-735.

7. Ohmae, T., and others, 'Microprocessor - Controlled High - Accuracy Wide - Range Speed Regulator for Motor Drives', IEEE Transactions on Industrial Electronics, Vol.IE-29, No.3, August 1982, PP. 207 - 211.

8. Lin, A.K., and Koepsel, W.W., 'A Microprocessor Speed Control System', IEEE Transactions on Industrial Electronic and Control Instrumentation, Vol. IECI-24, No.3, August 1977, PP. 241 - 247.

9. Reed, M., and Mergler, H.W., 'A Microprocessor - Based Control System', IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol. IECI-24, No.3, August 1977, PP. 253 - 257.

10. Kuo, B.C., Digital Control Systems, New York, N.Y.: Holt, Rinehart and Winston, Inc., 1980.

11. Kuelz, E., Jr., State Analysis Modeling Program, M.S. Thesis, VPI&SU, Blacksburg,VA., 1970.

12. Kuelz, E., Jr., and Grigsby, L.L., 'A State - Space Approach to System Simulation Using a Time - Shared Terminal', Proceedings of The 2nd Southeastern Symposium on System Theory, Univ. of Florida, 1970.

13. VanLandingham, H.F., and Grigsby, L.L., 'A Sampled - Data Approach to the Simulation of Nonlinear Networks', Proceedings of The 6th Southeastern Symposium on System Theory, Louisana State University, 1974.

14. Godiwala, P.M., and Moose, R.L., 'Digital Simulation of Nonlinear Systems with Deterministic or Stochastic Inputs', Proceedings of The 14th Southeastern Symposium on System Theory, Virginia Polytechnic Institute and State University, Blacksburg, Va., 1982.

15. Grigsby, L.L., 'Users Guide for Program SIM' unpublished lecture notes for Industrial Drives Group of Kollmorgen Corporation, July 1982.

16. Kollmorgen Corporation, Service and Installation Manual TPAR Series - M-037201 - Issue 5, Radford, Va.: Industrial Drives Group of Kollmorgen Corporation.

17. Kollmorgen Corporation, User's Guide for TPAR Amplifiers - M-037201, Radford, Va.: Industrial Drives Group of Kollmorgen Corporation, PP. 26 - 59.

18. D'azzo, J.J., and Houpis, C.H., Linear Control System Analysis and Design, New York, N.Y.: McGraw Hill, 1975.

19. Kollmorgen Corporation, 'DC Direct Drive Servo Motor - Tachs; Series TT-295X Specification Sheet' Radford, Va.: Industrial Drives Group of Kollmorgen Corporation.

20. Smith, R.J., Electronics - Circuits and Devices 2nd Edition, New York, N.Y.: John Wiley and Sons, 1980.

21. Kollmorgren Corporation, 'Phase Angle (SCR) Amplifier Series: SPA/TPA, SPAR/TPAR Specification Sheet', Radford, Va.: Industrial Drives Group of Kollmorgen Corporation.

22. Bonnert, R.R., 'Automatic Speed Control of One Quadrant DC Drives', IEEE Transactions on Industry Applicatons, Vol.IA-18, No.5, September/October 1982, PP. 491 - 499.

23. Sen, P.C., Thyristor DC Drives, New York, N.Y.: John Wiley and Sons, 1981.

24. Nunnally, C. E., Stability Considerations for a Class of Phase Modulated Switched Systems, Ph.D Dissertation, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia, 1968.

25. Pelly, B.R., Thyristor Phase - Controlled Converters and Cycloconverters, New York, N.Y.: Wiley, 1971.

26. Grotsollen, H., 'Comparison of Speed Controlled DC Drives With and Without Subordinate Current Loop', Proceedings of The 2nd IFAC Symposium, Dusseldorf, F.R.G., 3-5 October 1977, New York N.Y.: Pergamon Press, 1978.

27. McMurray, W., 'The Closed - Loop Stability of Power Converters with an Integrating Controller', IEEE Transactions on Industry Applications, Vol.IA-18, No.5, September/October 1982, PP. 521 - 531.

28. Parrish, E.A., and McVey, E.S., 'A Theoretical Model for Single - Phase Silicon Controlled Rectifier Systems', IEEE Transactions on Automatic Control, Vol. AC-12, No.5, October 1967, PP. 577 - 579.

29. Texas Instruments, Texas Instruments SR-51A Calculator Owner's Manual, Dallas, Texas: Texas Instruments Inc, 1975.

30. Mounfield, W.P., Adaptive Control By Signal Synthesis, M.S. Thesis, Mechanical Engineering Dept, Virginia Polytechnic Institute and State University, Blacksburg,Va, 1980.

31. Tektronix, Plot 10 Advanced Graphing II, Users's Manual, Tektronix,Inc: Beaverton, Oregon, August 1979, VPI&SU User Services No. M-AGII-2.

The vita has been removed from the scanned document

A NO LOAD SIMULATION MODEL OF A DC DRIVE SYSTEM

by

Guy Mandel Dudley

(ABSTRACT)

This research investigates the modeling and simulation, using the state transistion approach to nonlinear system simulation, of a DC drive system. The drive system that was modeled is a closed loop system composed of a velocity loop with an inner current loop. The power stage is composed of a pulse generator module and a scr bridge, while the motor is a permanent magnet DC motor.

A detailed development of each module model is included to reveal the open loop characteristics of the system. A total system state model was developed from each of the modules and closed loop simulations were run. The results show that the system may be modeled and simulated using this technique. However, the exactness of the simulation is heavily dependent on the DC motor parameters selected for the model.