To the Graduate Council:

I am submitting herewith a thesis written by John Murray Hickey III entitled "Computer Control of an Elementary Hardware Power System Simulator." I recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Thomas W. Reddoch

Thomas W. Reddoch, Major Professor

We have read this thesis and recommend its acceptance:

W. L. Green

C. C. Travis

Accepted for the Council:

Hilton A. Smith

Vice Chancellor
Graduate Studies and Research

COMPUTER CONTROL OF AN ELEMENTARY HARDWARE POWER SYSTEM SIMULATOR

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee

John Murray Hickey III

March 1975

ACKNOWLEDGMENTS

The author is indebted to Dr. Thomas W. Reddoch, Assistant Professor of Electrical Engineering, for his help, encouragement, and constructive criticism in the preparation of this thesis.

A sincere thanks to my parents, Dr. and Mrs. John M. Hickey Jr., for instilling in me the desire for a higher education and their encouragement and support in reaching this goal.

Appreciation is expressed to Jon Baker for his help on the electronics design, Laurence Phillips for his help in the construction of the facilities for the simulator, and to the Department of Electrical Engineering for granting me a graduate assistantship.

1197628

ABSTRACT

This thesis presents a realization of a closed loop control scheme for an elementary hardware power system simulator using a computer as the control element of the scheme. A general power system model is derived and is related to the model for the hardware power system simulator. Identification techniques of the hardware power system simulator parameters are presented. Sampled-data systems design techniques are applied to the model of the simulator and a digital control algorithm is derived. Several test cases are presented in order to demonstrate the feasibility of the system simulator.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
II.	SYSTEM MODELING	4
	Control Area Concept	4
	Derivation of Control Area Model	5
	Derivation of the Model of the Hardware Control Area Simulator	10
III.	SYSTEM IDENTIFICATION	17
	Implementation of the Simulator	17
	System Identification of the D.C. Motor	20
	Measurement of the electrical time constant	20
	Measurement of the torque constant (K_t)	20
	Measurement of the moment of inertia (J)	24
	Measurement of the voltage constant (K_e)	26
	Determination of K_f	26
	Tachometer Constant K_A	26
IV.	DESIGN OF THE CONTROLLER	30
	Sampled-Data Systems Representation	30
	Design of the Controller	32
	Design philosophy	32
	System specifications	33
	Application of Bode design techniques to design the controller	33

CHAPTER PAGE

The use of the computer as a controller 40

V. SIMULATOR PERFORMANCE 47

System Performance 47

Uncontrolled performance 48

Controlled performance 48

Hardware Performance 50

Tachometer 50

PWM controller 51

PDP-11/20 Computer 54

VI. CONCLUSIONS 56

LIST OF REFERENCES 57

APPENDIXES 59

Appendix A 60

Appendix B 63

Appendix C 68

VITA 80

LIST OF FIGURES

FIGURE		PAGE
1.	Hybrid Computer Simulator	2
2.	Block Diagram for $\Delta P_{tie}(s)$	8
3.	Block Diagram of $\Delta F(s)$	9
4.	Block Diagram of Hardware Power System Simulator	11
5.	Equivalent Circuit for the D.C. Motor	12
6.	Block Diagram of an Isolated Control Area	15
7.	Arrangement of the Hardware Power System Simulator	18
8.	Block Diagram of the D.C. Motor With K_f	21
9.	Test Set Up for R_a Measurement	22
10.	Torque vs. Armature Current	25
11.	Generated Voltage vs. Speed	27
12.	Tachometer Voltage vs. Speed	29
13.	Block Diagram of the Uncompensated System	34
14.	Final Block Diagram of the Uncompensated System	36
15.	System Bode Plots	38
16.	Block Diagram of the Compensated System	41
17.	Block Diagram of the Compensated System Showing the Position of the Computer	43
18.	Flow Diagram of Clock Interrupt Service Routine	44
19.	General Flow Diagram of Overall Controller Program	46
20.	Motor Speed Variations of the Controlled System	49

FIGURE PAGE

21. PWM Controller Waveforms 53

22. Output Configuration of the PWM Controller 54

CHAPTER I

INTRODUCTION

Digital computer simulation of power system dynamics through the use of mathematical models has been well explored in the literature. Various controls for power systems have been derived and have been applied to these system models in order to check the effects on stability and the ability of the system to respond under changing load conditions. A requirement of this method is an accurate mathematical model of a power system. These models are lengthy due to the fact that the power system is modeled in its entirety. This method, while feasible, is sometimes prohibitive in terms of the costs for computer use time due to lengthy execution times of the program and memory size limitations of the available computer. This method does not explore the use of the computer as a central element in a closed loop control scheme.

Recently, hybrid computers have been used in an effort to circumvent the factors mentioned above [1]. This scheme, illustrated in Figure 1, uses an analog computer to simulate the power system dynamics and a small digital computer is used to investigate different control or data monitoring schemes. Various control schemes can be implemented by writing a separate computer program for each scheme. Several advantages are realized from this type of simulator. The speed that an A.C. load flow study can be obtained in reduced since the digital computer does not have to solve a large set of simultaneous algebraic equations. Solution

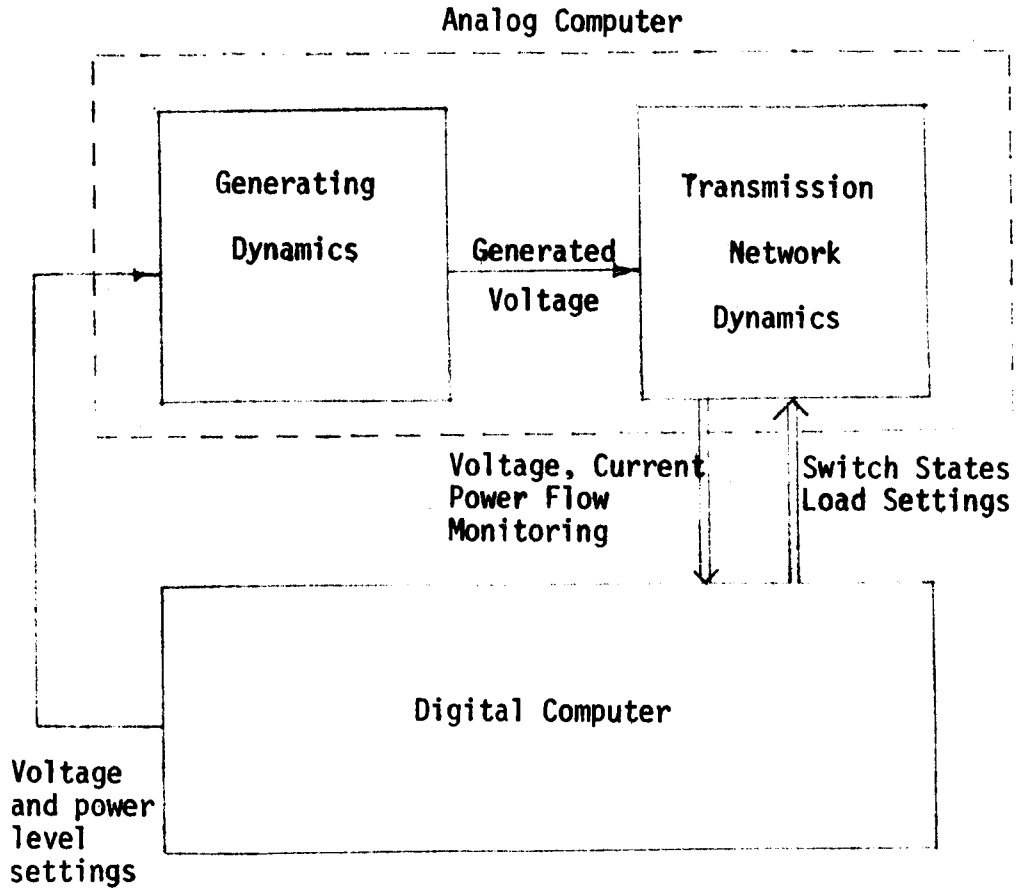


Figure 1. Hybrid Computer Simulator

times for transient stability studies are enhanced greatly by the fact that the digital computer does not have to solve large sets of simultaneous differential equations. The hybrid simulator does all of its computing in parallel instead of sequentially, thus eliminating the need for complex numerical techniques which exist for pure digital simulators. The fast solution times with which A.C. load flow and transient stability studies can be obtained can also be applied to steady-state contingency evaluations, dynamic contingency evaluations, and to state estimation techniques. The hybrid simulator can be interfaced with control panels to allow real-time operator training. This familiarizes an operator with the system response characteristics under different operating conditions and potential operating conditions which could lead to trouble.

An extension of the hybrid simulation concept is a hardware power system simulator to produce the power system dynamics. This scheme has some added advantages that a hybrid computer does not have. It allows an investigator of power system dynamics to experience first hand the limitations of the hardware components and how to identify potential operating hazards. It also gives the investigator working experience with interfacing and input/output difficulties with the digital computer. This thesis presents a practical realization of a hardware power system simulator coupled with a minicomputer to implement a closed loop control.

CHAPTER II

SYSTEM MODELING

I. CONTROL AREA CONCEPT

Present day power systems are interconnected to form power pools. These power pools consist of several power companies or control areas interconnected to form a composite system. The power pool concept is used in order to maintain continuity of service to customers (better security) and to provide for the most economical power production.

However, each control area is expected to absorb its own load changes and maintain its share of the scheduled net tie-line interchange to other areas within the pool.

In order for a multiple-area power system to be able to meet demands placed upon it by the power pool, each control area must be well regulated in order to provide for maximum efficiency and economical power production. Before a policy is established for a control area which is connected to others, various control concepts of a single control area must be investigated. An isolated control area is one that consists of a generation source and load with no connections to other control areas. This thesis is devoted to the investigation of an isolated control area.

II. DERIVATION OF CONTROL AREA MODEL

Any investigation of power system performance requires a dynamic model for a control area. A model of a typical control area of an interconnected system is developed and then simplified to that of an isolated area.

It is assumed that the area experiences a real power load change of ΔP_D. Due to natural governing action, the area increases its output by ΔP_G. The surplus power thus generated will be absorbed by the area in three ways:

1. The area's kinetic energy due to the rotating of a generating source will increase.
2. Load consumption will increase.
3. Power can be absorbed by other control areas.

This situation can be expressed as

$$\Delta P_G - \Delta P_D = \frac{d}{dt} W_K + D\Delta f + \Delta P_{tie} \tag{2.1}$$

where

$\frac{d}{dt} W_K$ is the rate of change of kinetic energy

$D\Delta f = \frac{\partial P_D}{\partial f} \Delta f$ is the rate of change of the dissipated power as the frequency varies

ΔP_{tie} is the power flow to other control areas. ΔP_{tie} is positive when the power flows out of the control area.

It can be shown for a linear system that [2]

$$\frac{d}{dt} W_K = \frac{2W_K^\circ}{f^\circ} \frac{d}{dt} \Delta f$$

where

W_K° is the control area's initial kinetic energy

f° is the nominal frequency throughout the control area.

If $M \equiv \frac{2W_K^\circ}{f^\circ}$ and is combined with equation (2.1)

$$\Delta P_G - \Delta P_D = M \frac{d}{dt} \Delta f + D\Delta f + \Delta P_{tie} \qquad (2.2)$$

Furthermore, it can be shown that the power interchange is approximated by [2]

$$\Delta P_{tie} = \sum_{\nu} T_\nu^\circ \left(\int \Delta f dt - \int \Delta f_\nu dt \right) \qquad (2.3)$$

where

T_ν° is a constant that represents the synchronization between control areas

Δf_ν is the incremental frequency variable due to phase angle deviation.

Laplace transforming equation (2.3)

$$\Delta P_{tie} = \frac{1}{s} \sum_{\nu} T_{\nu}^{\circ} \, [\Delta F(s) - \Delta F_{\nu}(s)] \tag{2.4}$$

From equation (2.4), a block diagram is derived for $\Delta P_{tie}(s)$ and is shown in Figure 2. Frequency deviations in adjoining control areas are represented by the $\Delta F_{\nu}(s)$'s. Now, Laplace transforming and rearranging equation (2.2)

$$\Delta P_G(s) - \Delta P_D(s) - \Delta P_{tie}(s) = \Delta F(s) \, (\frac{M}{D} s + 1)D \tag{2.5}$$

Now, let

$$\tau_p = \frac{M}{D}$$

$$K_p = \frac{1}{D}$$

$$G_p(s) = \frac{K_p}{1 + s\tau_p}$$

then equation (2.5) becomes

$$[\Delta P_G(s) - \Delta P_D(s) - \Delta P_{tie}(s)]G_p(s) = \Delta F(s) \tag{2.6}$$

A block diagram for $\Delta F(s)$ can be constructed from equation (2.6) as is shown in Figure 3. $\Delta P_G(s)$ is the power generated by the prime mover of

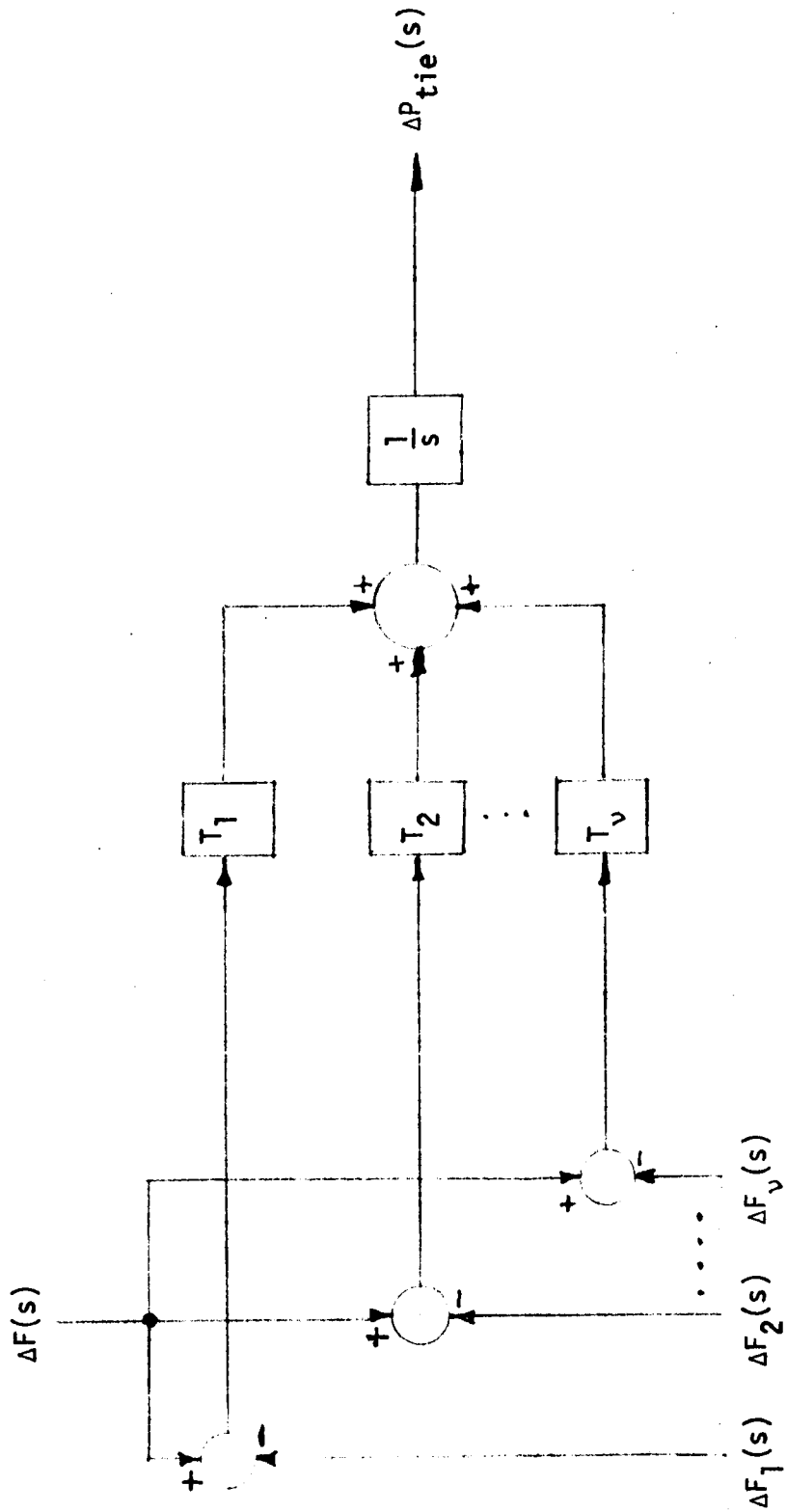


Figure 2. Block Diagram for $\Delta P_{tie}(s)$

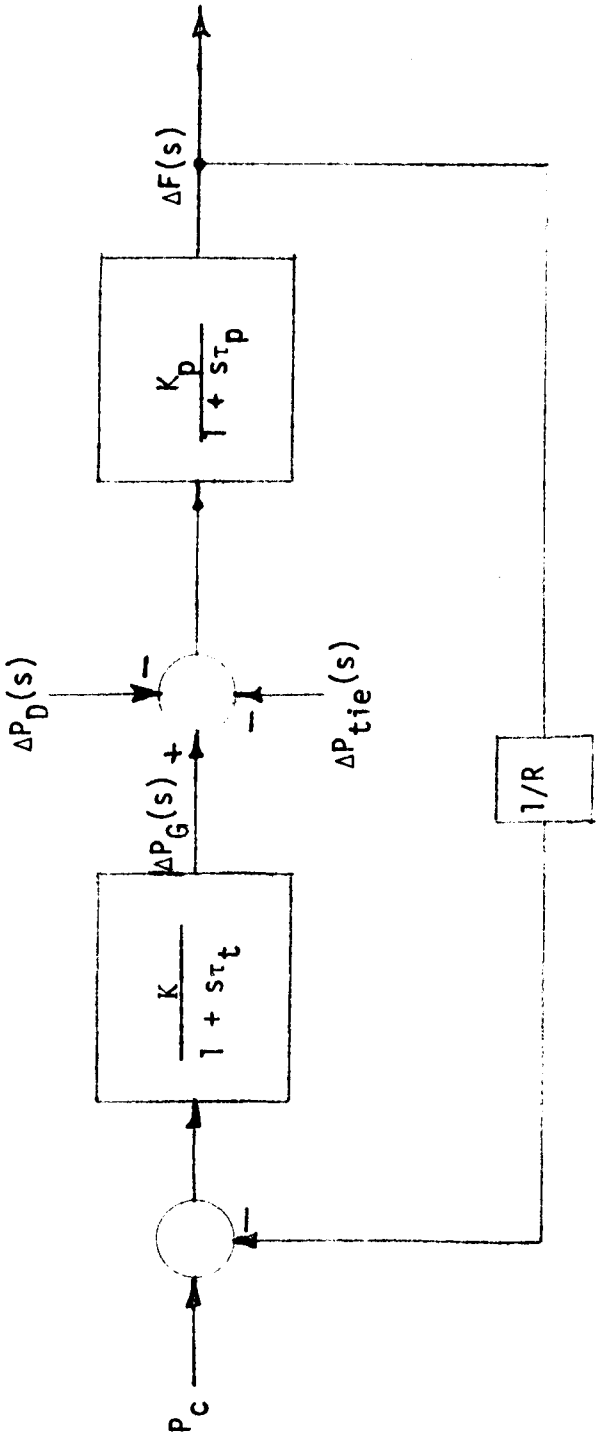


Figure 3. Block Diagram of ΔF(s)

the system. The prime mover, represented in time constant form, is controlled by natural governing action provided by the proportional control loop which regulates the speed of the system. Supplementary control for $\Delta P_G(s)$ can be entered in the control area at P_c. This part of the system is also shown in Figure 3.

The simplification of this general system representation to an isolated control area can be easily shown. If there is no connection to other control areas then the term $\Delta P_{tie}(s)$ is zero. This leaves only $\Delta P_D(s)$ and $\Delta P_G(s)$ as inputs to the system.

III. DERIVATION OF THE MODEL OF THE HARDWARE CONTROL AREA SIMULATOR

The single control area hardware simulator consists of a d.c. motor as the prime mover for the system, a d.c. generator, a power amp, and a load resistance. The system configuration is shown in Figure 4.

The equivalent circuit of the separately excited d.c. motor is shown in Figure 5. Since the d.c. motor is separately excited, the dynamic effects of the field circuit are constant and are absorbed in the other constants of the motor. Writing the loop voltage equation for the d.c. motor

$$V_t = R_a\, i_a(t) + L_a \frac{di_a(t)}{dt} + e_b \tag{2.7}$$

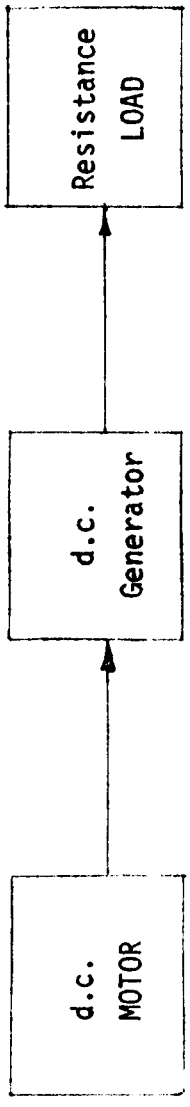


Figure 4. Block Diagram of Hardware Power System Simulator

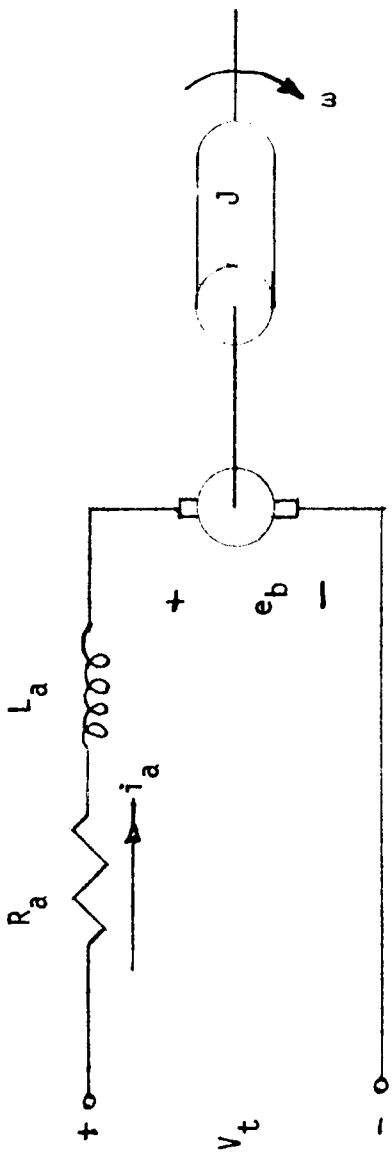


Figure 5. Equivalent Circuit for the D.C. Motor

where

R_a is the d.c. resistance of the armature windings

L_a is the inductance of the armature windings

e_b is the back e.m.f. produced by the armature.

$i_a(t)$ is the time varying armature current

V_t is the d.c. driving voltage

The electrical torque produced by the d.c. motor is related to the armature current by

$$T_e = K_t i_a \tag{2.8}$$

where

K_t is a constant.

Writing the torque balance equation

$$T_e = J \frac{d\omega}{dt} + T_L \tag{2.9}$$

where

J is the mass moment of inertia of the shaft of the motor

ω is the rotational speed of the shaft in radians/second

$\frac{d\omega}{dt}$ is the instantaneous acceleration of the shaft.

T_L is the opposing load torque presented to the motor by the demands of the rest of the simulator.

The speed is related to the back e.m.f. by

$$e_b = K_e \omega \tag{2.10}$$

where

K_e is a constant.

Laplace transforming equations 2.7, 2.8, and 2.9

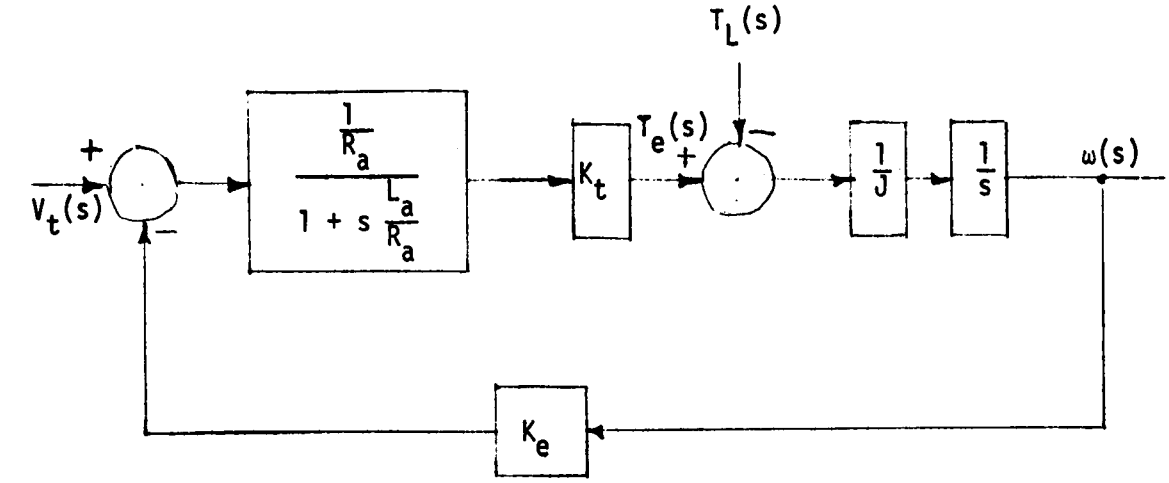
$$V_t(s) - K_e \omega(s) = R_a I_a(s) + S\, L_a I(s) \tag{2.11}$$

$$T_e(s) = K_t I_a(s) \tag{2.12}$$

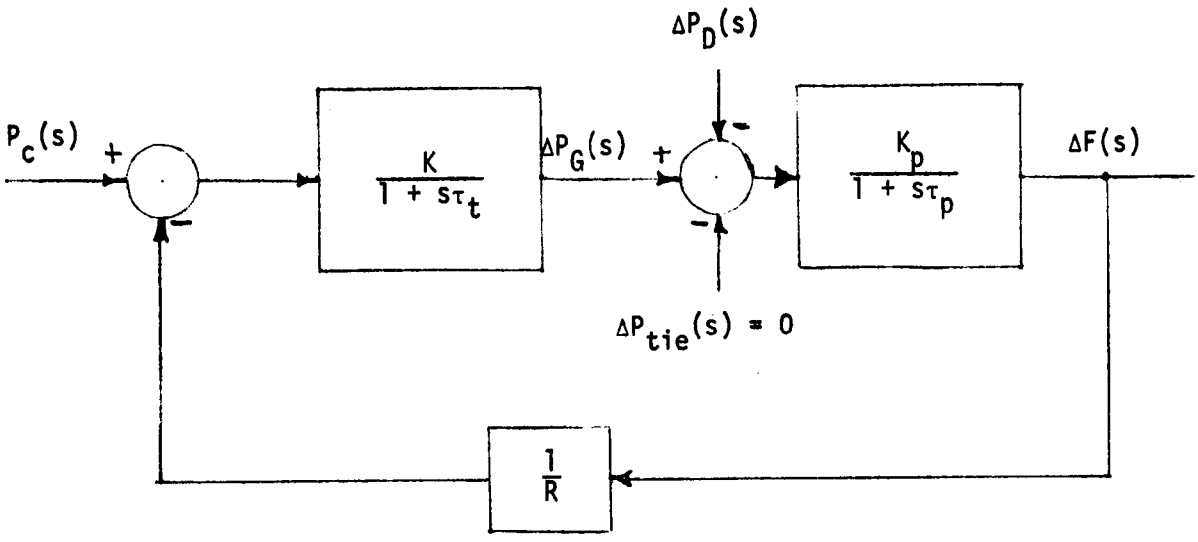
$$T_e(s) - T_L(s) = s\, J\omega(s) \tag{2.13}$$

A block diagram constructed from equations 2.11, 2.12, and 2.13 is shown in Figure 6a. Relating this to the single control area model derived in the previous section, $\Delta F(s)$ is $\Delta\omega(s)$, $\Delta T_e(s)$ is $\Delta P_G(s)$ and $\Delta T_L(s)$ is $\Delta P_D(s)$. It is assumed that there are no disturbances due to other control areas, thus $\Delta P_{tie}(s)$ is zero. The natural control loop is shown as the back e.m.f. loop. Figure 6b shows the block diagram of the general control area in order to show the similarities of the two models.

The equations for the d.c. generator, and the load resistance are not derived because their effects can be represented to the d.c. motor



(a)



(b)

Figure 6. Block Diagram of an Isolated Control Area

as a change in load torque. The general control philosophy of this isolated control area can be summed up as one of regulating the d.c. motor to ignore $\Delta T_L(s)$.

CHAPTER III

SYSTEM IDENTIFICATION

I. IMPLEMENTATION OF THE SIMULATOR

The hardware power system simulator consists of the following components: a d.c. motor, d.c. generator, a resistance load, a solid state pulse-width modulated power amp, electronic instrumentation, a d.c. tachometer, and a PDP-11/20 computer. A general block diagram for this arrangement is shown in Figure 7. The d.c. motor, the d.c. generator, and the load resistance are standard equipment of Lab Volt Educational Systems of Farmingdale, New Jersey. Table I specifies the rating of each of these components.

The solid state pulse-width modulated power amp electronics and the control circuits have been developed in an undergraduate special projects course at the University of Tennessee. The basic idea consists of producing a square pulse train of constant frequency but varying width. When this signal is applied over a period of time, an average value of voltage is presented to the d.c. motor as a driving force. Thus, by controlling the width of the pulses, the speed of the d.c. motor can also be controlled. This circuit is represented by a gain of 12.

The PDP-11/20 computer is a general purpose 16-bit machine. It is a modular system with a wide range of software capability and peripherals

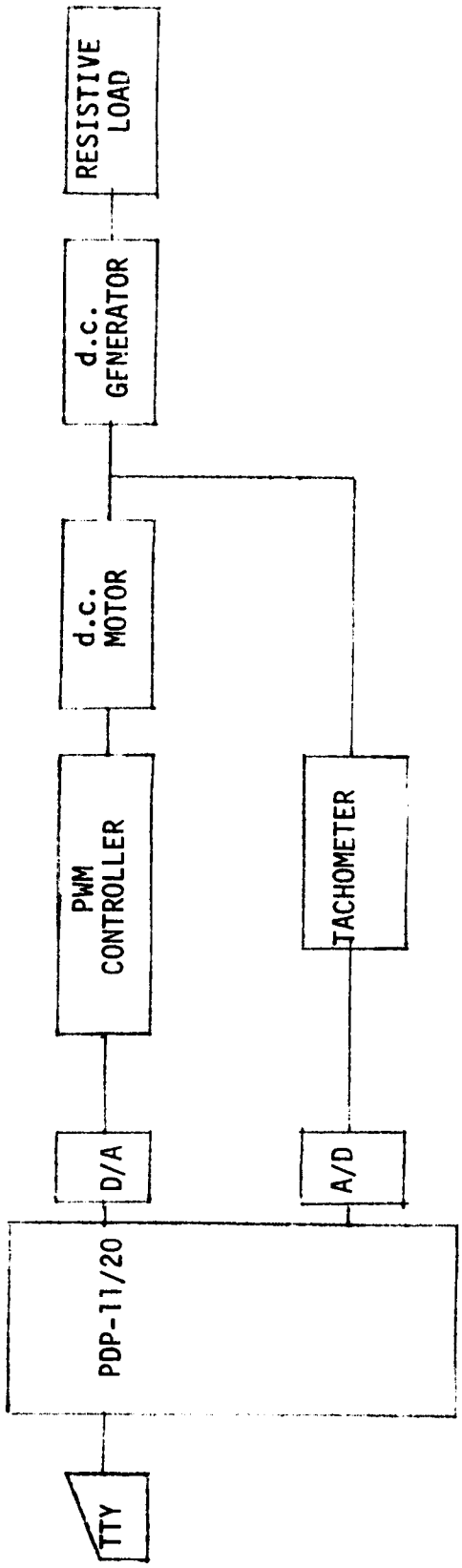


Figure 7. Arrangement of the Hardware Power System Simulator

TABLE I

HARDWARE COMPONENT RATINGS

d.c. motor	1/4 hp, 1800 $\frac{rev}{min}$, 120vdc
d.c. generator	120w, 1800 $\frac{rev}{min}$, 120vdc, 1A
resistance load	Loading capacity is 0 - 252w in - 12w steps, 5% accuracy 120 volts

which are not normally found in comparable 16-bit machines. A detailed description of the PDP-11/20 and its peripherals can be found in Appendix A and in References 3 and 4.

II. SYSTEM IDENTIFICATION OF THE D.C. MOTOR

The block diagram derived in Chapter II for the d.c. motor is again presented in Figure 8 but with one modification. A constant (K_f) has been added to the output of the integrator block. Its significance will become evident.

Measurement of the Electrical Time Constant

The electrical time constant of the motor is $\frac{L_a}{R_a}$. L_a has been determined through the use of an impednace bridge to be 44.25 millihenrys.

R_a has been determined through the use of the test set-up shown in Figure 9. The measured values of armature current, armature voltage, and the calculated values of R_a are tabulated in Table II. As can be seen, the value of R_a was not constant and a nominal value of 8.0 ohms was chosen. Thus, the electrical time constant is

$$\frac{L_a}{R_a} = \frac{44.25 \text{ mh}}{8\Omega} = 5.53 \times 10^{-3} \text{sec}$$

Measurement of the Torque Constant (K_t)

K_t has been found by measuring the torque produced by the motor and plotting this against the armature current. K_t is the slope of this curve.

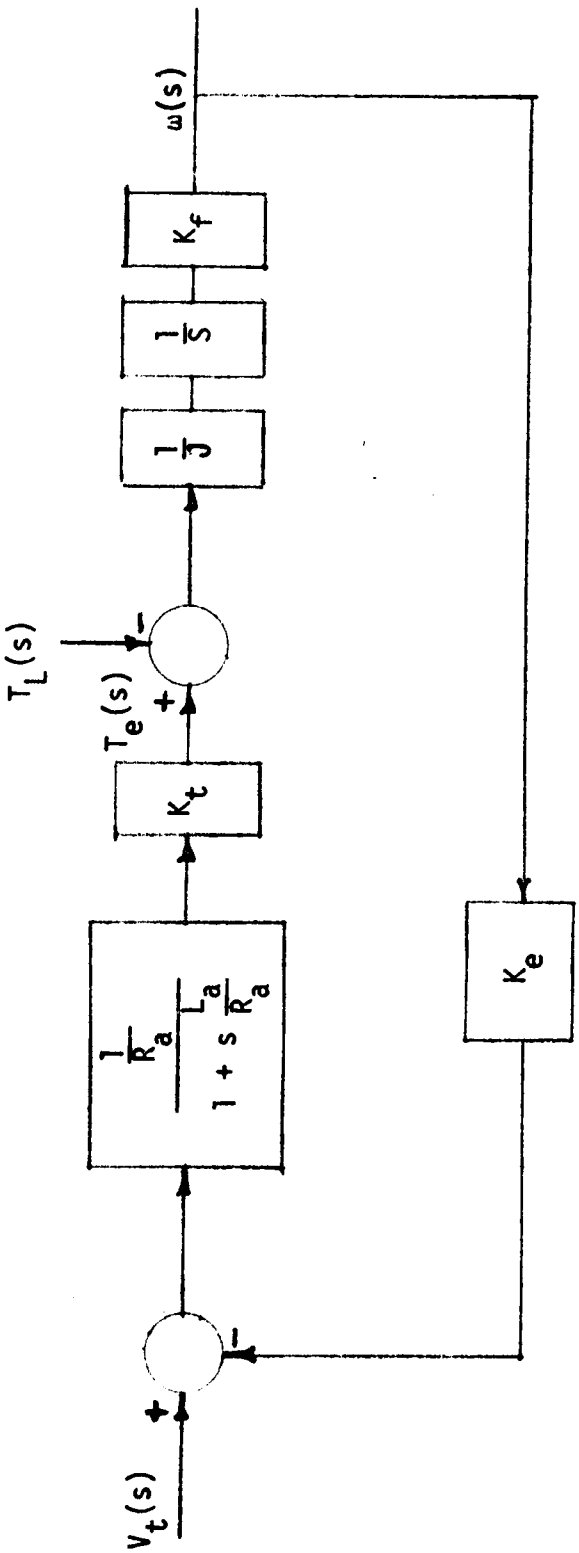


Figure 8. Block Diagram of the D.C. Motor With K_f

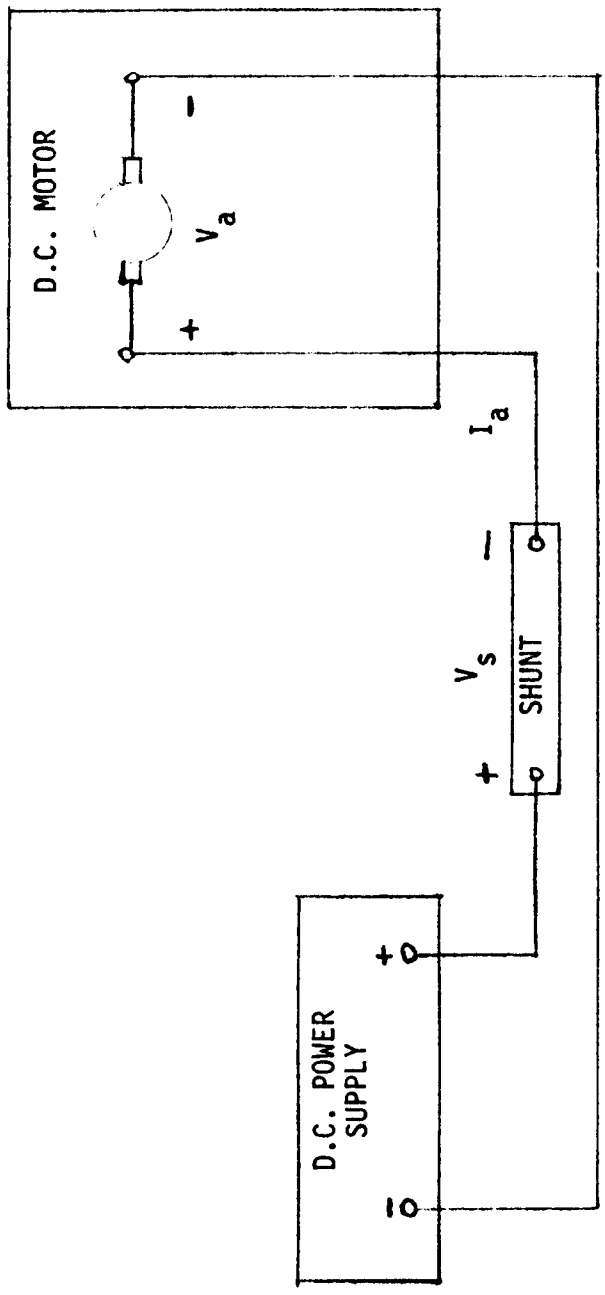


Figure 9. Test Set Up for R_a Measurement

TABLE II

R_a MEASUREMENT DATA

V_a (Volts)	I_a (Amps)	R_a (Ohms)
7.63	1	7.63
9.65	1.25	7.72
11.59	1.50	7.73
13.83	1.75	7.90
16.11	2.0	8.05
18.51	2.25	8.23
20.9	2.5	8.36
23.4	2.75	8.51

The torque is measured by attaching a 12 inch moment arm to the shaft of the d.c. motor. As voltage is applied to the d.c. motor, torque is developed thus pressuring the moment arm against a scale. The armature current is monitored for each increment of force. Thus, a graph of torque versus armature current is found. This graph is shown in Figure 10 with

$$K_t = \frac{\Delta T}{\Delta I_a} = .111 \frac{\text{ft - lb.}}{\text{Amp}}$$

Measurement of the Moment of Inertia (J)

The moment of inertia of the shaft was measured by using the theory of the torsion pendulum. The shaft was attached to the end of a three foot length of .029 inch diameter piano wire and twisted 180 degrees. The period of rotation (T) was measured to be 5.51 seconds. It has been shown that [6]

$$J = \frac{T^2 n\, r^4}{8\pi\, \ell} \tag{3.1}$$

where n is the modulus of rigidity of the wire

r is the radius of the wire

ℓ is the length of the wire

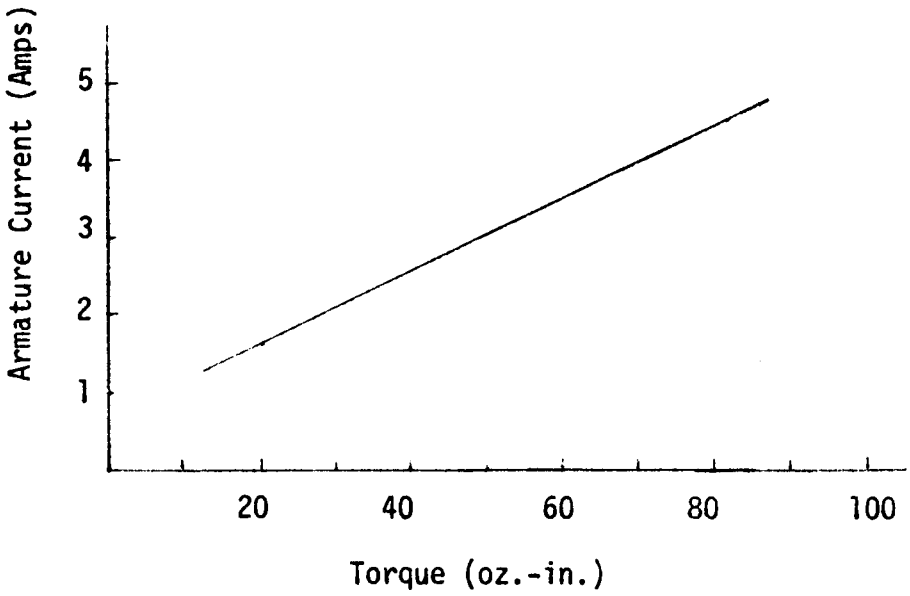


Figure 10. Torque vs. Armature Current

Therefore, from equation 3.1

$$J = 1.14 \times 10^{-3} \text{ ft - lb - sec}^2$$

Measurement of the Voltage Constant (K_e)

K_e was determined by driving the d.c. motor as a generator and measuring the generated voltage (V_g) as speed (n) varied. The speed was measured through the use of a tachometer whose calibration will be discussed in a later section. A graph of generated voltage versus speed is shown in Figure 11. K_e is the slope of this curve. Thus,

$$K_e = \frac{\Delta V_g}{\Delta n} = .048 \frac{\text{Volts}}{\text{rpm}}$$

Determination of K_f

K_f is added to convert the speed from radians per second to revolutions per minute. Thus,

$$K_f = \left(\frac{1 \text{ revolution}}{2\pi \text{ radians}}\right)\left(\frac{60 \text{ sec.}}{1 \text{ min.}}\right)$$

$$K_f = 9.55$$

Tachometer Constant K_A

A permanent magnet d.c. generator has been used as a tachometer. Its output voltage has a significant noise component present, however,

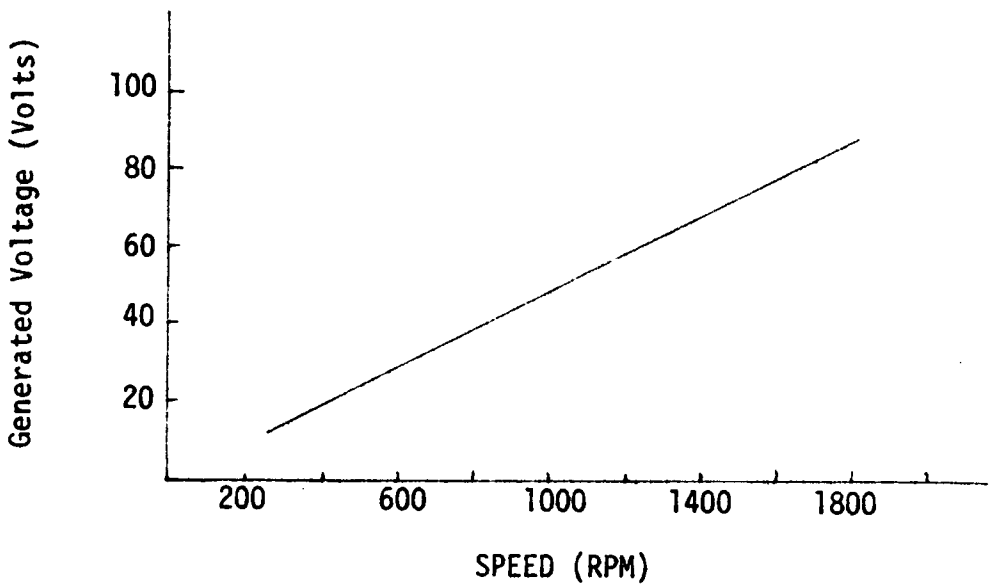


Figure 11. Generated Voltage vs. Speed

it can be filtered. An investigation showed that the lowest frequency of the noise was approximately 100 Hz. An active low-pass filter with a gain of unity has been designed by an accepted procedure [7] to remove the noise. The low-pass filter reduced the magnitude of the noise voltage to an acceptable level. The d.c. level generated by the tachometer is preserved. The tachometer has been calibrated using a strobe light which produces the graph of tachometer voltage (V) versus speed (n) as shown in Figure 12. The tachometer constant (K_A) is the inverse slope of this graph.

$$K_A = \frac{\Delta n}{\Delta V} = 511.23 \frac{\text{rpm}}{\text{volt}}.$$

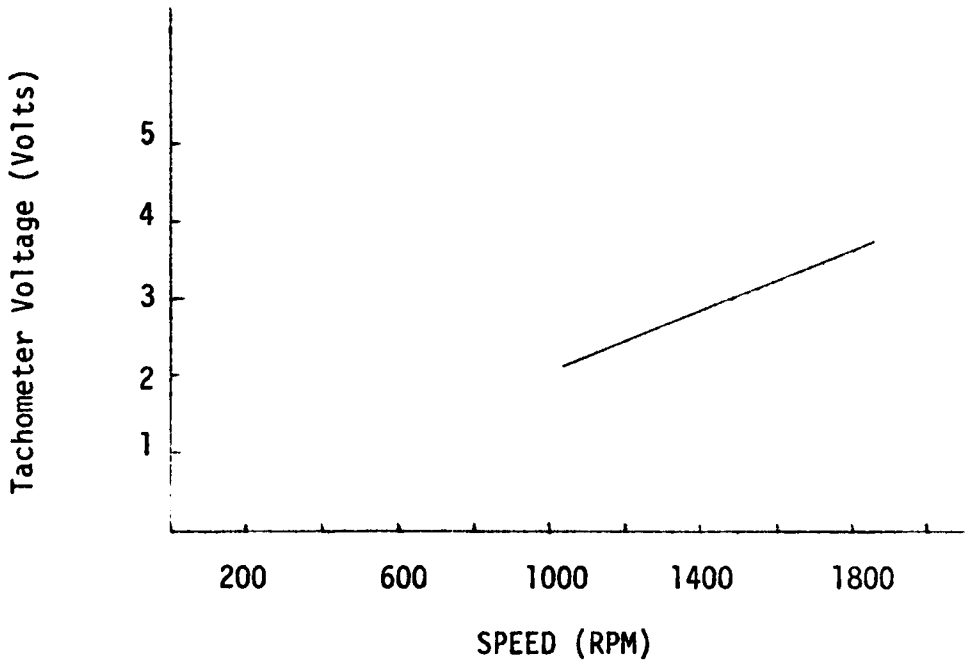


Figure 12. Tachometer Voltage vs. Speed

CHAPTER IV

DESIGN OF THE CONTROLLER

Techniques for deriving a sampled-data system representation from a known continuous systems model [8] and the design of sampled-data systems controllers [9, 10] are reviewed. The derivation of a control for the hardware power system simulator is presented. The various aspects of the needs of the controller are discussed.

I. SAMPLED-DATA SYSTEMS REPRESENTATION

Continuous systems are usually represented by differential equations. In sampled-data systems the energy difference of the states of the system at each sampling instant is of interest. Thus, sampled-data systems are described by difference equations which represent these energy differences. The z-transform can be used to solve difference equations in much the same way that the Laplace transform is used to solve differential equations.

An ideal sampler is represented by a train of unit impulses separated by the sampling period T

$$f^*(t) = \sum_{n=0}^{\infty} \delta(t - nT)$$

Using complex convolution, the sampled signal $f^*(t)$ can be transformed to

$$F^*(s) = F(z) = \sum_{\substack{\text{all poles} \\ \text{of } F(s)}} \text{Residues of } \left[\frac{F(\lambda)}{1 - e^{\lambda T} e^{-sT}}\right]$$

$$= \sum_{\substack{\text{all poles} \\ \text{of } F(s)}} \text{Residues of } \left[\frac{F(\lambda)}{1 - e^{\lambda T} z^{-1}}\right] \qquad (4.1)$$

where

$$z = e^{sT}$$

From these principles, an open loop transfer function in terms of z can be determined. The left half of the s-plane is mapped into the interior of the unit circle in the z-plane using the transformation $z = e^{sT}$. Similarly, the right half of the s-plane is mapped into the exterior of the unit circle in the z-plane.

Frequenty response techniques cannot be applied directly in the z-domain because $z = e^{sT}$ is a transcendental function. However, through the use of a bilinear transformation, the interior of the unit circle can be mapped into the left half of an auxilliary plane usually called the w-plane. This transformation converts the open loop transfer function back into a domain in which standard Bode frequency response design techniques can be used.

The bilinear transformation is

$$z = \frac{1 + W}{1 - W}, \quad W = \frac{1 - z^{-1}}{1 + z^{-1}}. \tag{4.2}$$

If $z = e^{j\omega T}$ is substituted into equation 4.2 then

$$\omega = \frac{1 - e^{-j\omega T}}{1 + e^{-j\omega T}} = j \tan\left(\frac{\omega T}{2}\right)$$

The W-plane frequency, $W = u + jV$, is related to the s-plane frequency, $S = \sigma + j\omega$, by

$$V = \tan\left(\frac{\omega T}{2}\right)$$

II. DESIGN OF THE CONTROLLER

Design Philosophy

The addition or subtraction of loading on the motor results in addition or subtraction of opposing load torque presented to the motor. This can be considered as an unwanted disturbance input to the system. In order to maintain 120 volts at the load of the d.c. generator, the speed of the motor must be kept near 1800 rpm. Thus, the general design philosophy is to regulate the speed of the motor to ignore random step inputs of load torque while maintaining constant speed.

System Specifications

Since steady-state speed regulation is the ultimate goal of the controller, it is desired that the error between the desired speed and the actual speed be kept within 1.0%. It is well known that the steady-state error is related to the velocity error constant by

$$e_{ss} = \frac{1}{K_v}$$

Thus,

$$K_v = 100$$

It is desired that the system remain stable during operation, therefore, a phase margin of 40 degrees was arbitrarily chosen.

Application of Bode Design Techniques to Design the Controller

A block diagram of the uncompensated system is shown in Figure 13. It can be shown [8] that the single sampler of the error signal is enough to represent the sampling action of the computer. The block that contains $(1 - e^{-sT})/s$ is the transfer function of a zero-order hold (ZOH) which represents the action of a digital-to-analog converter (D/A).

By considering T_L to be zero, the transfer function relating speed $\omega(s)$ to the input $V_t(s)$ is

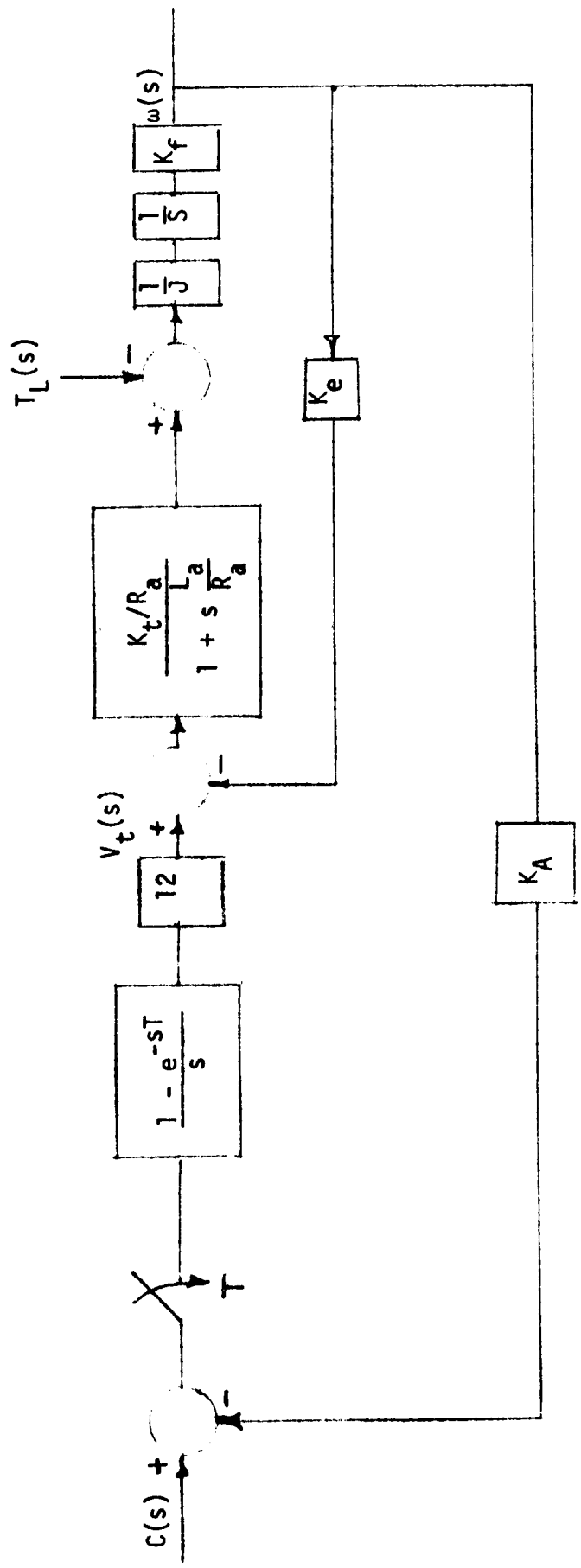


Figure 13. Block Diagram of the Uncompensated System

$$\frac{\omega(s)}{V_t(s)} = \frac{1.06}{(s + 5.76)\ (s + 175.04)}$$

The system takes the form shown in Figure 14.

For notational convenience, let

$$G_p(s) = \frac{(1 - e^{-sT})\ (12)\ (1.06)}{s(s + 5.76)\ (s + 175.04)}$$

and

$$H(s) = 0.002$$

R(s) is the desired reference speed for the motor. The transfer function relating $\omega(s)$ to the reference value R(s) is

$$\frac{\omega(s)}{R(s)} = \frac{G_p(s)}{1 + G_p(s)H(s)}.$$

Hence, the open loop transfer function of this system without compensation is

$$G_p(s)H(s) = \frac{(1 - e^{-sT})(.02544)}{s(s + 5.76)(s + 175.04)}. \qquad (4.3)$$

Choosing a sampling interval of .017 sec. and using equation 4.1 to z-transform equation 4.3

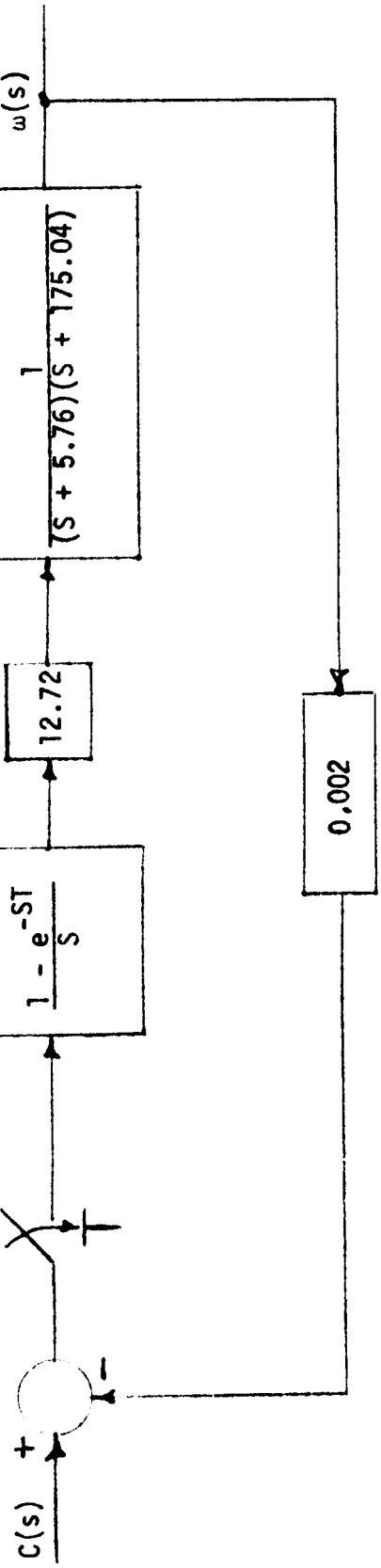


Figure 14. Final Block Diagram of the Uncompensated System

$$G_p H(z) = \frac{(z + .011)(z + .585)}{(z - .907)(z - .051)}. \tag{4.4}$$

Now, applying the bilinear transform defined in equation 4.2, equation 4.4 becomes

$$G_p H(w) = (.055) \frac{(1 + \frac{w}{1.02})(1 + \frac{w}{3.82})}{(1 + \frac{w}{.049})(1 + \frac{w}{1.107})} \tag{4.5}$$

The Bode diagram of equation 4.5 is shown in Figure 15. The error constant (K_v) is defined as

$$K_v = \frac{2}{T} \lim_{w \to 0} w\, G_p H(w). \tag{4.6}$$

It can easily be seen that equation 4.5 in its present form cannot meet the K_v specification. Thus, a compensator must contain the factor 1/w

$$G_c(w) = \frac{1}{w}. \tag{4.7}$$

Combining equations 4.5 and 4.7

$$G_c\, G_p\, H(w) = .055 \frac{(1 + \frac{w}{1.02})(1 + \frac{w}{3.82})}{w(1 + \frac{w}{.049})(1 + \frac{w}{1.107})}.$$

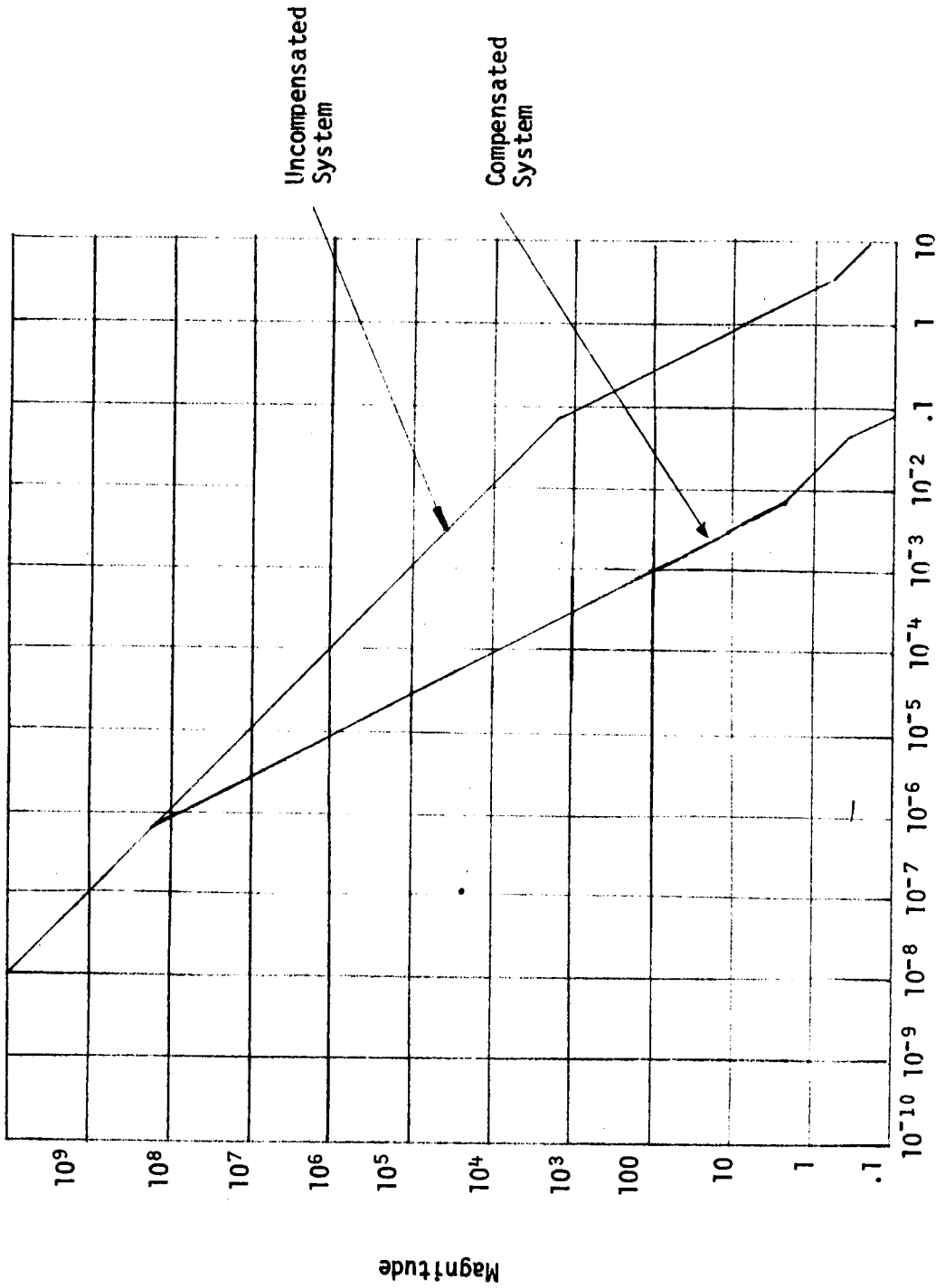


Figure 15. System Bode Plots

Hence, $K_v = .055$ for the compensated system.

Thus, in order to satisfy the K_v specification, additional gain, K, must be added to the system. Solving for this gain

$$K_v = 100 = \frac{2}{.017}(.055)K$$

$$K = 15.45$$

Thus, G_c becomes

$$G_c(w) = \frac{15.45}{w} . \tag{4.8}$$

Using a phase angle approximation [9], the phase margin is found to be

$$\theta_m \cong 16.3^\circ .$$

This does not meet the defined specification therefore, additional lag compensation is necessary. Using an accepted Bode design procedure [9], the additional lag compensation necessary in order to meet the phase margin specification causes $G_c(w)$ to become

$$G_c(w) = \frac{15.45\ (1 + \frac{w}{.007})}{w(1 + \frac{w}{7.3 \times 10^{-7}})} \tag{4.9}$$

Thus, all system specifications are met with the compensator defined by equation 4.9. The Bode diagram of the compensated system is also shown in Figure 15.

Transforming back to the z-domain using the bilinear transform defined in equation 4.2 yields

$$G_c(z) = \frac{2222.3 + 30.9\ z^{-1} - 2190.8\ z^{-2}}{1{,}369{,}864 - 2{,}739{,}726z^{-1} + 1.369{,}862z^{-2}} \qquad (4.10)$$

Figure 16 shows the block diagram of the compensated system.

The Use of the Computer As A Controller

In order to implement $G_c(z)$ on the PDP-11/20 computer, a difference equation describing the control signal that is applied to the system must be determined. $G_c(z)$ is the transfer function that relates the output control signal, COT(z), to the input error signal, EIN(z). Thus,

$$G_c(z) = \frac{COT(z)}{EIN(z)} = \frac{2222.30 + 30.9\ z^{-1} - 2190.89\ z^{-2}}{1{,}369{,}864 - 2{,}739{,}726z^{-1} + 1{,}369{,}862z^{-1}}.$$

Solving for COT(z)

$$COT(z) = .0016\ EIN(z) + 2.25 \times 10^{-5}\ z^{-1}\ EIN(z) - .0016\ z^{-2}\ EIN(z)$$

$$+ 2.0z^{-1}\ COT(z) - z^{-2}\ COT(z) \qquad (4.11)$$

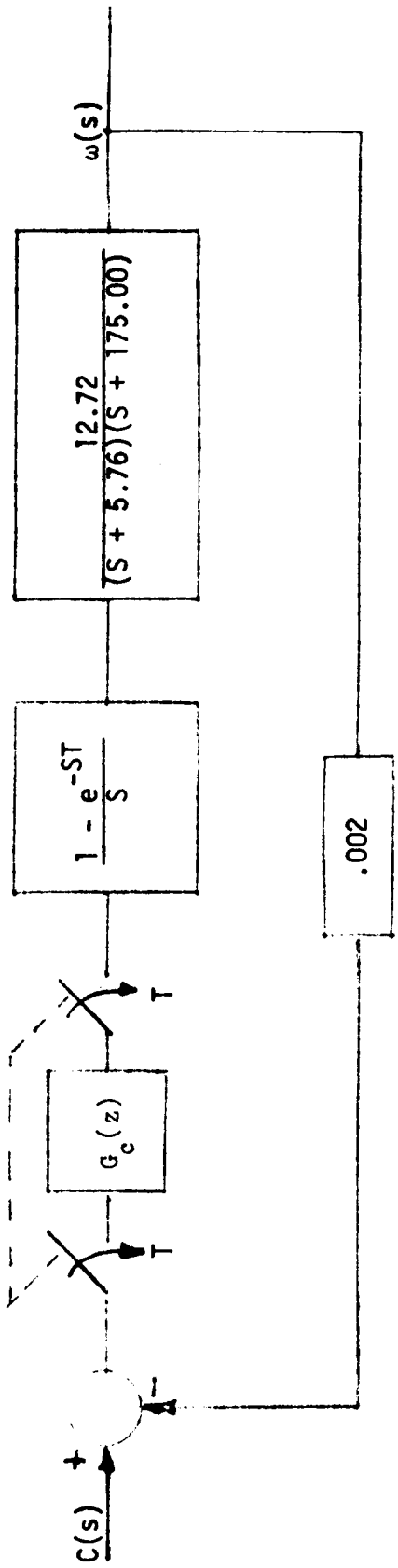


Figure 16. Block Diagram of the Compensated System

Using the real translation theorem [10] on equation 4.11, a difference equation can be found

$$COT[nt] = .0016\ EIN[nt] + 2.25 \times 10^{-5}\ EIN[(n-1)T] - .0016\ EIN[(n-2)T] + 2.0\ COT[(n-1)T] - COT[(n-2)T] \quad (4.12)$$

where

COT[nT] is the new control signal

EIN[nT] is the present value of the error signal

EIN[(n-1)T] is the first past value of the error signal

EIN[(n-2)T] is the second past value of the error signal

COT[(n-1)T] is the first past value of the control signal

COT[(n-2)T] is the second past value of the control signal.

Equation 4.12 is implemented with a computer program on the PDP-11/20 computer using the direct programming technique [10]. Figure 17 shows the position of the computer in the system.

The computer has been programmed to provide a real time interrupt rate of 60 Hz. A flow diagram of the program that is executed during each interrupt is shown in Figure 18. During each interrupt, a sample of the feedback signal is taken, the error signal is calculated, the control, COT[nT], is determined, and is applied to the system through the D/A converter.

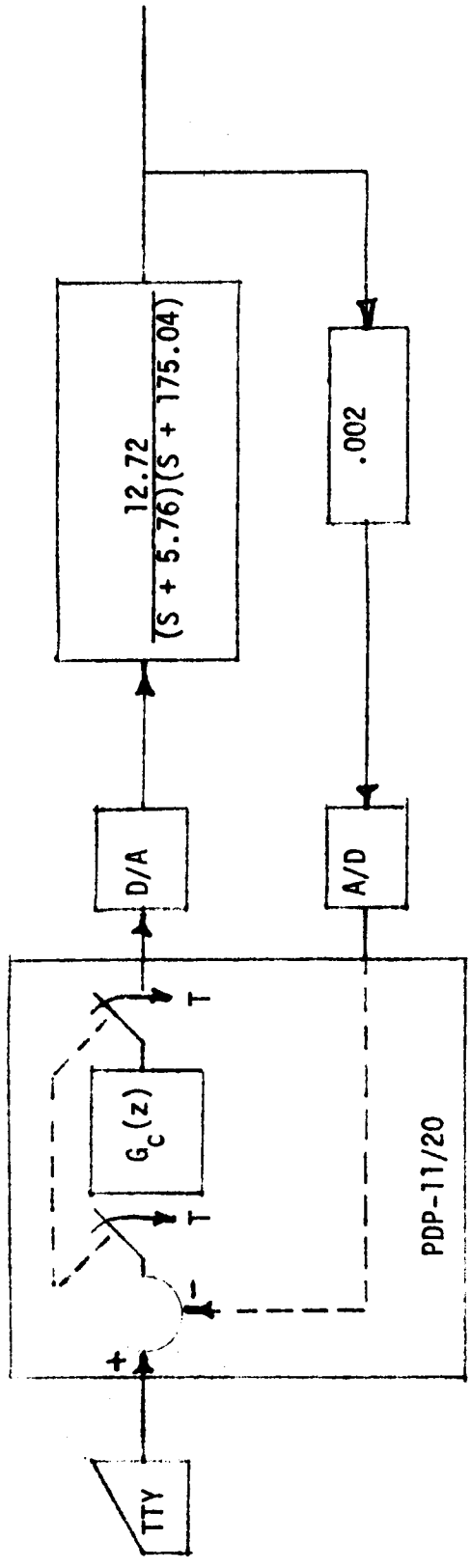


Figure 17. Block Diagram of the Compensated System Showing the Position of the Computer

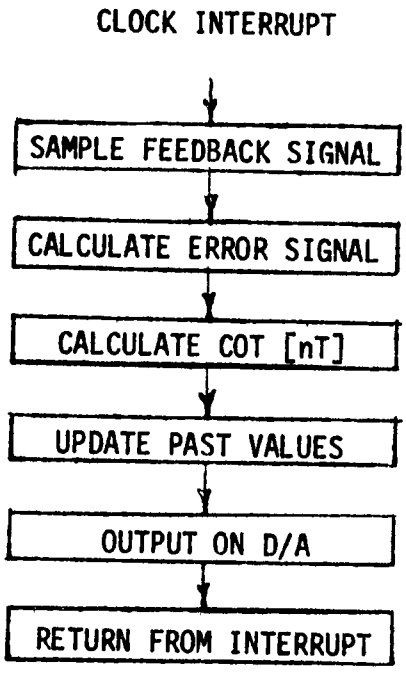


Figure 18. Flow Diagram of Clock Interrupt Service Routine

Calculations are performed by a set of standard Digital Equipment Corporation floating point math routines tailored specifically for the hardware available on the PDP-11/20 [11]. The overall controller program has been designed to provide a variable reference in order to bring the motor up to speed slowly before applying the compensator to the system. This is done in order to prevent unwanted current surges in the power amplifier. A general flow diagram of the complete program can be found in Figure 19. A more detailed description of the program is found in Appendix B.

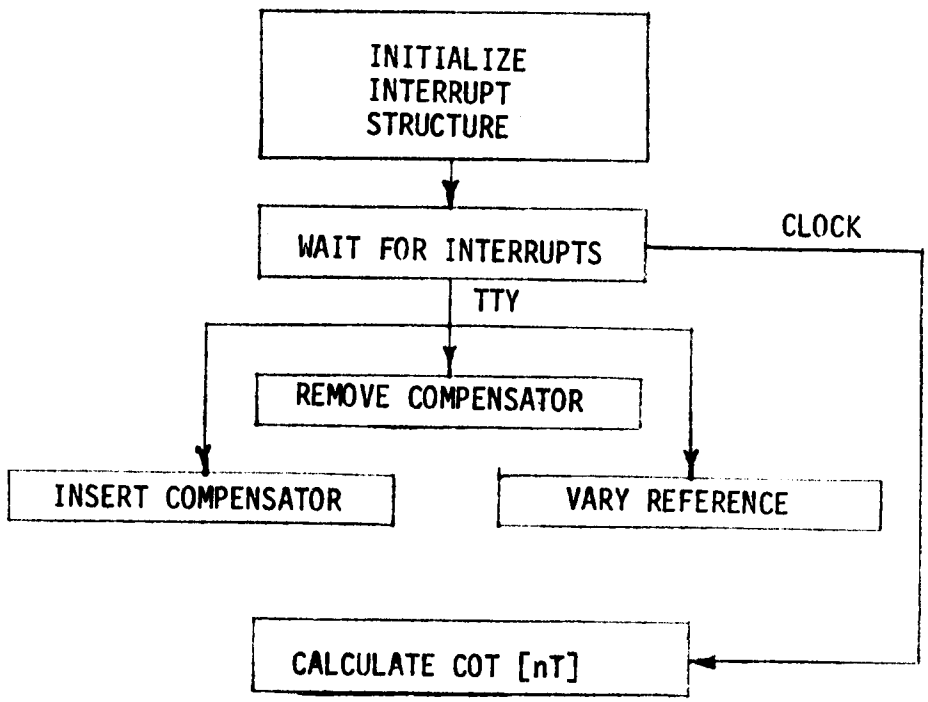


Figure 19. General Flow Diagram of Overall Controller Program

CHAPTER V

SIMULATOR PERFORMANCE

I. SYSTEM PERFORMANCE

The resistive load on the output of the d.c. generator is varied to change the load torque input for the d.c. motor. The system is investigated for four different loading conditions. The loads are chosen to present the system operation at various power levels and incremental load situations. These load changes are summarized in Table III with the value of load torque that each change represents.

Two varibles are monitored in order to determine the performance of the system. They are:

1. The armature current of the motor
2. The speed of the motor.

These varibles are investigated under various load conditions for the uncontrolled and controlled system operation.

TABLE III

LOAD CHANGES AND UNCONTROLLED PERFORMANCE

Load Resistance (Ω)	Load Torque (ft - lbs)	Uncontrolled Speed (RPM)	Uncontrolled Armature Current (AMPS)
(no load) ∞	0.79	1800	2.0
300	1.20	1600	3.3
200	1.35	1500	3.9
(full load) 171.4	1.40	1450	4.1

Uncontrolled Performance

Table III contains the variations of the motor speed and motor armature current at each load. The speed was adjusted to 1800 rpm at no load. Then the resistive load was applied to the system at intervals long enough to allow the monitored system parameters to reach new steady-state values. As expected, the speed dropped with each application of load to the system. The armature current increased with each load change due to the natural governing action of the motor.

Controlled Performance

A K_v specification of 100 provided a compensator with an acceptable level of performance. In Figure 20, the transient speed variations for the compensated system are shown for each load change described in Table III.

When the first load change occurred, the speed dropped to 1,625 rpm in .4 seconds. Control action taken by the digital compensator caused the speed to settle out at 1800 rpm in 4.0 seconds. The second load change reduced the speed to 1,675 rpm in .6 sec and the digital compensator corrected the speed to 1800 rpm in 2.4 seconds. In the third load condition, the speed dropped to 1700 rpm in .8 seconds, however the digital compensator returned the speed to 1800 rpm in 3.6 seconds.

The nature of the tachometer signal, discussed in the section on hardware performance, made it impossible to accurately determine the steady-state speed after each load change. Thus, an accurate check for

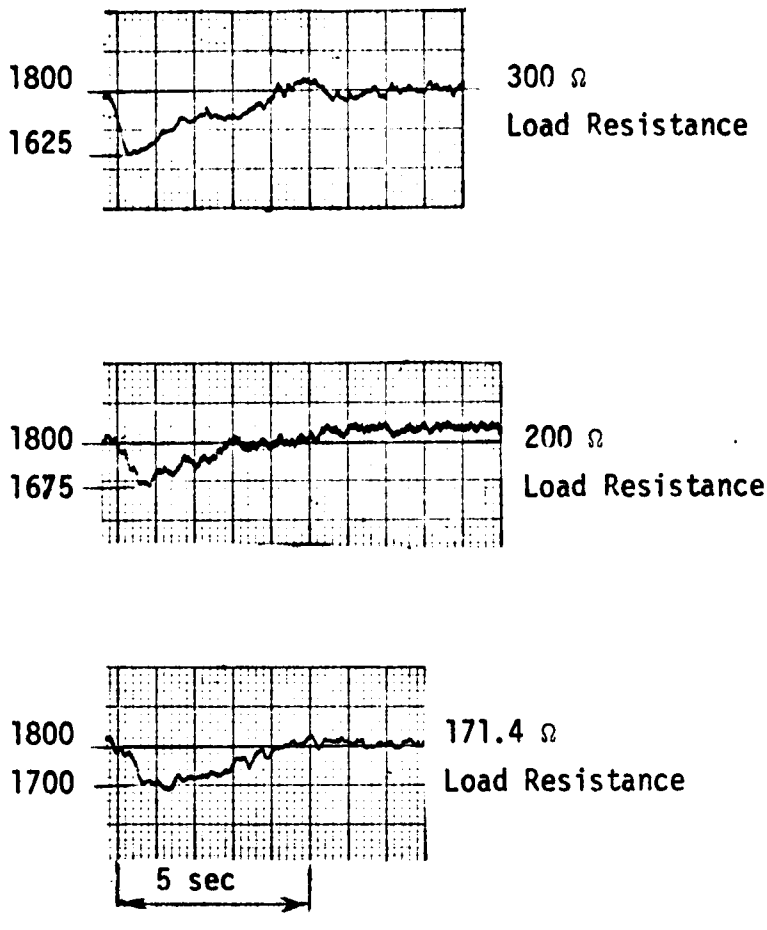


Figure 20. Motor Speed Variations of the Controlled System

the K_v specification was not made. However, the average value of the tachometer signal appeared to be approximately 3.6 volts which corresponds to 1800 rpm.

These curves show that the digital compensator did satisfactorily regulate the speed of the motor to approximately 1800 rpm in steady-state. Inaccuracies in the performance of the system are possibly due to the following reasons:

1. Noisy tachometer signal produced by the permanent magnet d.c. generator and perhaps by a flexible coupling with the shaft of the d.c. motor.
2. Inaccuracies in the identification of the parameters of the d.c. motor.
3. Roundoff error in the calculation of the digital compensator difference equation.

II. HARDWARE PERFORMANCE

Tachometer

The tachometer performed satisfactorily, but with two undesirable effects:

1. A very low frequency component was observed to be riding on the top of the generated d.c. level. The magnitude of this component, which was about .4 volts, caused the speed of the motor to oscillate about its steady-state value. This component could either be due to the characteristic function of the tachometer, or perhaps to the elastic coupling of the tachometer shaft to the shaft of the d.c. motor.

2. A noise component of approximately 0.1 volts was produced by the tachometer's characteristic function and was not attenuated by the low-pass filter. It was this noise component which made an exact measurement of the steady-state speed impossible. This noise can be observed in Figure 20.

PWM Controller

The PWM controller is a combination pulse-width modulator and power amplifier. The pulse-width modulator is constructed from operational amplifiers and is powered by a ± 15 Vdc supply. The power amplifier portion is biased with + 120 Vdc supplied by a d.c. generator and is electrically isolated from the pulse-width modulator portion of the circuit. At the output stage of the power amplifier, a transistor is used in a switching application. When the pulse-width modulator supplies a positive step, the output transistor saturates which applies + 120 volts to the terminals of the d.c. motor. When +120 volts is applied to the terminals of the d.c. motor, the armature current spikes to approximately 15 amps due to the action of the inductance in the armature windings. This spike is very short in duration. When the spike is extinguished, the current builds up at a constant rate consistent with the electrical time constant of the d.c. motor. When the pulse-width modulator provides a step change to zero, the output transistor is cut-off and the input terminal of the d.c. motor is effectively grounded. The armature current then starts to decay according to the electrical time constant until the output transistor is again saturated and the

process continues at 1 Khz rate. This situation is graphically illustrated in Figure 21a. Over several periods, this presents to the d.c. motor an average d.c. value of voltage and current.

Two problems were encountered using this arrangement. The first one had to do with the bias supply of the power amplifier portion. When the output voltage waveform is switched from the +120 Vdc position to the ground position in a step change, a positive voltage spike of approximately 400 volts appears on the +120 Vdc bias supply waveform due to the action of the circuit inductance in the d.c. generator windings and the chart capacitance. This power line "glitch" exceeded the collector-emitter voltage rating of the output transistor when it was attempted to bring the motor from standstill to full speed in one step. The "glitch" has been minimized by placing a 10 amp shunt load across the d.c. generator in addition to the load of the PWM controller. When the output waveform changes from ground to +120 Vdc in a step change, a negative voltage "glitch" appears on the bias supply waveform. This "glitch" has been minimized by placing a 1μf capacitor across the +120 Vdc bias supply. This situation is shown in Figure 21b.

The second problem is related to the armature current spike. The output transistor is rated at 10 amps and this spike would occasionally exceed this rating long enough to blow the transistor. This problem has been circumvented by paralleling two transistors in the output stage in the configuration shown in Figure 22. The 1.4 ohms placed in the emitter circuit of each transistor is there to aid in current division between the two transistors.

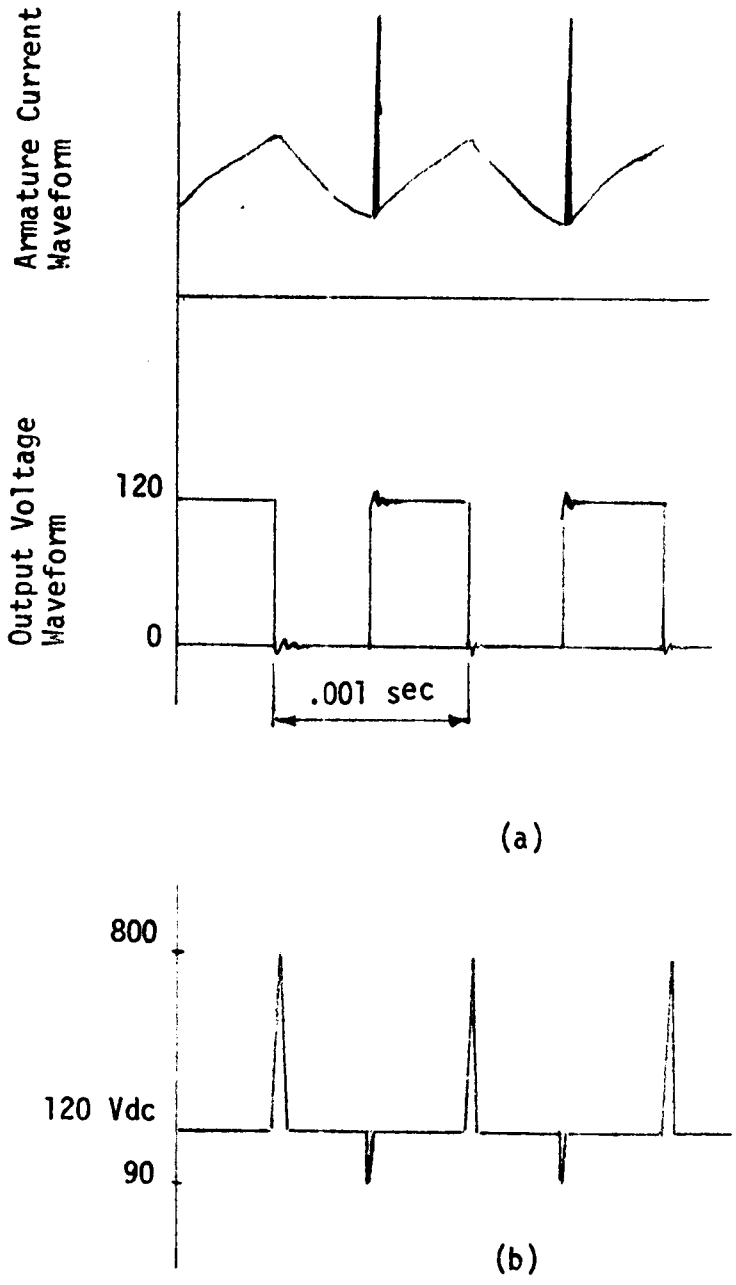


Figure 21. PWM Controller Waveforms

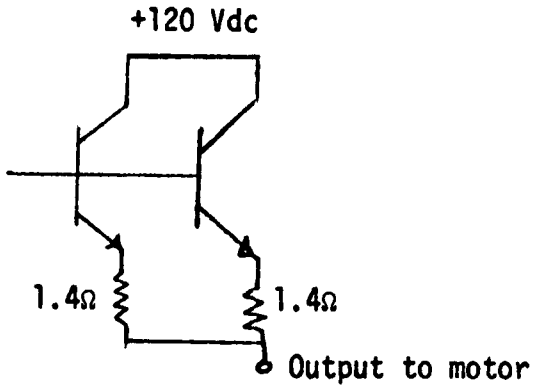


Figure 22. Output Configuration of the PWM Controller

III. PDP-11/20 COMPUTER

The digital compensator program performed satisfactorily except at the saturation limit of the D/A converter and PWM controller. The PWM controller responds only to inputs of 0-10 Vdc. When this range is exceeded on either end, the output of the controller remains at the maximum or minimum point. The D/A has a range of ± 10 volts. When the positive end of the D/A range is exceeded, the D/A output switches suddenly from + 10 volts to - 10 volts.

During system operation, it was observed that under certain load conditions, the value of the control signal required to return the motor speed to 1800 rpm exceeded +10 volts. This caused the D/A output to switch to -10 volts thus effectively halting the motor before it could be returned to 1800 rpm. In order to circumvent this problem, the gain of the PWM controller was doubled. The coefficients of the digital algorithm were then adjusted accordingly in order to

provide for K_v = 100. This adjustment decreased the range of response of the PWM controller to 0-5 volts, but it allowed the output signal of the D/A to remain well below +10 volts. The affect on the system's response is that when full load is applied to the system, it takes the system longer to bring the motor speed back to 1800 rpm than would be expected. This effect can be seen in Figure 20 (page 49).

CHAPTER VI

CONCLUSIONS

A dynamic model for an isolated control area has been developed and is compared to the dynamic model of a d.c. motor. The motor model has been found to be very similar to the model of an isolated control area and is used as a simulator. A digital compensator has been designed with acceptable performance. Based on studies in this thesis, the following recommendations are suggested to improve the performance:

1. A tachometer or some type of speed measuring device that provides a noise free output would improve the accuracy of the control.
2. The PWM controller's performance could be improved by using a regulated +120 Vdc power supply to supply the bias voltage on the power amplifier portion.
3. Addition of some proportional speed control should generally improve the performance.
4. As the size of the system is increased, the computer should be removed from the main control loop.

LIST OF REFERENCES

LIST OF REFERENCES

1. Michaels, Laurence, William Tessmer, John Muller, "The On-Line Power Systems Simulator," 1973 IEEE Intercon Technical Paper, 1973.

2. Elgard, Ollie I., Electric Energy Systems: An Introduction, New York: McGraw-Hill Book Co., 1971.

3. PDP-11 Processor Handbook, Digital Equipment Corporation, Maynard, Mass., 1971.

4. PDP-11 Peripherals and Interfacing Handbook, Digital Equipment Corporation, Maynard, Mass., 1971.

5. PDP-11 Paper Tape Software Programming Handbook, Digital Equipment Corporation, Maynard, Mass., 1973.

6. Pollack, Herman W., Applied Physics, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1964.

7. Graeme, Jerald G., Lawrence P. Huelsman, Gene E. Tobey, Operational Amplifiers, Design and Applications, New York: McGraw-Hill Book Co., 1971.

8. Kuo, B. C., Analysis and Synthesis of Sampled-Data Control Systems, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1963.

9. Eveleigh, Virgil W., Introduction to Control Systems Design, New York: McGraw-Hill Book Co., 1972.

10. Saucedo, Roberto, Earl E. Schiring, Introduction to Continuous and Digital Control Systems, New York: The MacMillian Co., 1968.

11. FPMP-11 User's Manual, Digital Equipment Corporation, Maynard, Mass., 1973.

APPENDIXES

APPENDIX A

THE PDP-11/20 MINICOMPUTER AND ITS PERIPHERALS

This appendix gives a general description of the Digital Equipment Corporation PDP-11/20 minicomputer and its peripherals. For a more detailed description of the hardware and the programming, see references 2, 4, and 5 in List of References.

I. THE PDP-11/20

The PDP-11/20 is a 16-bit, general-purpose, parallel logic computer. All communication between system components is done on a single, bidirectional, high speed bus called the Unibus. This allows the user to address device registers as memory, hence all PDP-11/20 instructions are available for input/output (I/O) operations.

Peripheral devices operate on a multi-line priority interrupt system. Polling of devices is eliminated since the interrupt hardware selects and begins executing the appropriate service routine automatically. A device's service priority may be set independent of the hardware interrupt priority.

The most efficient use of the minicomputer can be realized when it is programmed in assembly language. The advantages of using assembly language programming are fast execution time, efficient utilization of memory, and access to the entire instruction set of the computer.

These benefits are especially helpful when working with real-time control applications.

II. PERIPHERALS

Extended Arithmetic Element

The extended arithmetic element (KE-11A) is an option which performs multiplication, division, multiple position shifts, and normalization. The standard set of Digital Equipment Corporation floating point math routines were specifically written to use the extended arithmetic element. It is connected directly to the Unibus and is programmed as a peripheral.

Programmable Real-Time Clock

The programmable real-time clock (KW11-P) provides scheduled real-time interrupts at specified intervals. Two crystal-controlled clock rates of 100 KHZ and 10 KHZ are available along with 60 HZ line frequency and an external clock input. The clock rate and the mode of operation are selected by setting specific bits in the control and status register. In the repeated interrupt mode, a 16-bit synchronous binary counter is decremented at the programmed clock rate until a count of zero is obtained. At this time an interrupt occurs. The counter is automatically reloaded from a 16-bit data buffer and the count is resumed while the service routine is being executed.

Analog-to-Digital Conversion

The analog-to-digital (A/D) convertor (ADØ-D) employed is a multiplexed four-channel data acquisition device. It provides 11-bit bipolar digitization of the analog signal at its input. Conversion time is approximately 35 μsec with an aperature of 100 nanoseconds. Full scale range is ± 10.0 volts.

Conversion is initiated under program control by setting bit Ø in the A/D status register. Bit 7 in the status register is set upon completion of conversion. The digitized analog voltage is made available to the program through a 16-bit data register.

Digital-to-Analog Conversion

The digital-to-analog (D/A) convertor is a peripheral device capable to converting a 12-bit, 2's complement digital word into the equivalent analog voltage. The D/A conversion is initiated by loading a D/A buffer register with the desired value. Settling time is maximum of 20 μsec for a full scale step change.

Teletype

The standard ASR 33 teletype is used to type in instructions to the program at a rate of up to ten characters per second. It is also equipped with a low speed paper tape punch and reader which operates at the same rate. Data input may be accomplished under either program or interrupt control.

APPENDIX B

THE SPEED CONTROL PROGRAM

The Speed Control Program (SPCTL) enables a user to investigate the loading effects of an isolated power system. The program has been written to accept three commands which give the user the ability to initialize a desired speed from the teletype, to insert the compensator into the system when steady-state unloaded speed is reached, and to remove the compensator when desired.

I. OPERATING PROCEDURE AND SYSTEM COMMANDS

Initial Operating Procedure

Hardware connections. Two hardware connections are required before operation can begin. They are: (1) connect the output of the tachometer low pass filter to A/D Channel No. Ø; (2) connect D/A Channel No. 1 to the input of the PWM hardware controller. This line contains the output of the digital compensator.

Loading. Enter the address of the absolute loader (57500_8) into the switch register. Load the binary tape for SPCTL into the high-speed reader, move the HALT switch to the ENABLE position, and then depress the START switch. The binary tape is then read into memory.

Starting. SPCTL is started automatically upon completion of the reading process. If the program halts for any reason during operation, it can be restarted at address 3000_8.

Initial dialogue. Upon starting, the program will respond by printing an asterick(*) on the teletype. The asterick indicates the program's readiness to accept a command.

Initial operating conditions. The program sets up several initial operating conditions when it is started. They are: (1) The digital compensator is not inserted into the system; (2) the sampling interval is set at 17 msec; (3) the speed reference value is set to Ø.Ø; (4) coefficients for the digital algorithm are calculated and stored for use by the clock service routine; and (5) past values are properly initialized.

Program Commands

Table IV summarizes the commands to which SPCTL responds. The command character, its format, and a brief description are given. The X used as a format character for the R command denotes any decimal number. SPCTL will only respond to the commands listed in Table IV. Any attempt to use some other character will cause SPCTL to respond with a "?". SPCTL will always indicate its readiness to accept another command by printing an asterick on the teletype.

TABLE IV

SPCTL COMMAND

Command	Format	Description
R	RXX.XX	This command is used to set the motor to the desired operating speed.
C	C	This command is used to insert the compensator into the system.
K	K	This command removes the compensator from the system.

II. PROGRAM DESCRIPTION

The Speed Control Program operates strictly on a priority interrupt basis. The execution of any portion of the program, with the initialization portion being the only exception, is always in response to an interrupt from the real-time clock or the teletype keyboard. The program is divided into seven sections that are described below.

Program Initialization

This portion of the program sets up the initial operating conditions described under Initial Operating Conditions.

Teletype Interrupt Identification

This portion of the program identifies which command has been issued to the program. Once the identification has been made, program control is passed to the command service routine.

Clock Interrupt Service Routine

This section of the program determines the control for the system through the use of the digital compensator. Its function is more fully described in Chapter IV.

Speed Reference Routine

The Speed Reference Routine is the service routine for the R command. This routine forms a 10-bit number which represents a voltage value of 0-10 volts. The formation of this 10 bit number is done through the use of "look-up" tables which add the octal equivalent of the decimal number to an accumulator. The final value is then output on the D/A.

This command is the only command that disables the clock interrupt during its execution. Therefore, it is assumed that the motor is in the unloaded state in order to prevent unwanted current surges in the PWM controller.

Compensator Routine

The Compensator routine is the service routine for the C command. This routine sets a flag that is examined by other portions of program in order to determine whether the digital compensator is in the system. When the flag is set, the digital compensator is inserted into the system.

Kill Routine

The Kill Routine clears the compensator flag thus removing the digital compensator from the system.

Power Fail/Restart Routine

This routine is provided in the event of a power failure at the computer. It saves all registers that are vulnerable to destruction when the power to the computer fails. When power is returned to the computer, this routine provides for continuation of SPCTL execution at the point at which the power failure occurred.

APPENDIX C

ASSEMBLY LANGUAGE LISTING OF THE SPEED CONTROL PROGRAM

The assembly language program used as the controller for the hardware power system is given in this appendix. For an explanation of the use of the program, see Appendix B.

```
        ;
        ;UNIVERSITY OF TENNESSEE
        ;ELECTRICAL ENGINEERING DEPARTMENT
        ;FEBRUARY 1975
        ;
        ;
        ;PROGRAMMER  JOHN M. HICKEY III
        ;
        ;
        ;
        RØ=%Ø
        R1=%1
        R2=%2
        R3=%3
        R4=%4
        R5=%5
        SP=%6
        PC=%7
        CSR=17254Ø          ;CLOCK STATUS
        CDB=172542          ;COUNT SET BUFFER
        TKS=17756Ø          ;TTY READER STATUS
        TKB=177562          ;TTYREADER BUFFER
        TPS=177564          ;TTY PRINTER STATUS
        TPB=177566          ;TTY PRINTER BUFFER
        DAB2=176762         ;D/A NO. 2
        ADSR=17677Ø         ;A/D STATUS
        ADDB=176772         ;A/D DATA BUFFER
        .=4
        .WORD .+2,Ø,.+2,Ø,.+2,Ø,.+2,Ø
        .WORD PDN,34Ø
        .WORD .+2,Ø,.+2,Ø
        .=6Ø
        .WORD TTYIN,24Ø
        .=1Ø4
        .WORD CLOCK,3ØØ
        .=3ØØØ
SPCTL:  MOV #7ØØ,SP         ;SET UP STACK
        MOV #Ø52,TPB        ;"*" TO TTY
        TSTB TPS            ;PRINTED?
        BPL .-4
        MOV #366.,SETPT     ;INITIALIZE SETPOINT
        SETPT=.-2
        CLR RELOC           ;INITIALIZE RELOC
        MOV #771.,-(SP)
        MOV #1ØØ.,-(SP)
        MOV #13.,-(SP)
        MOV #1ØØØ.,-(SP)
        MOV #16.,-(SP)      ;
```

```
MOV #1.,-(SP)      ;
MOV #1ØØØ.,-(SP)   ;
MOV #23.,-(SP)     ;
MOV #1Ø.,-(SP)
MOV #1ØØØØ.,-(SP)
MOV #2.,-(SP)      ;SET UP STACK FOR
MOV #1Ø.,-(SP)     ;FLOATING POINT
MOV #1Ø24.,-(SP)           ;VALUES FOR THE
MOV #2Ø48.,-(SP)   ;DIGITAL
MOV #1Ø.,-(SP)     ;COMPENSATOR
JSR R4,$POLSH      ;ENTER POLSH MODE
$IR
$STR
K2Ø48.
$IR
$LOD
K2Ø48.
$DVR
$STR
K2Ø48.
$IR
$STR
K1Ø24.
$IR
$LOD
K1Ø24.
$DVR
$STR
K1Ø24.
$IR                ;FLOAT 2.Ø
$STR               ;STORE IT
K2.ØØØ
$IR                ;FLOAT 1ØK
$STR               ;STORE IT
K.ØØØ23
$IR                ;FLOAT 1Ø
$LOD               ;LOAD 1ØK
K.ØØØ23
$MLR               ;COMPUTE 1ØØK
$STR               ;STORE IT
K.ØØØ23
$IR                ;FLOAT 23
$LOD               ;LOAD 1ØK
K.ØØØ23
$DVR               ;COMPUTE .ØØØ23
$STR               ;STORE IT
K.ØØØ23
$IR                ;FLOAT 1ØØØ
```

```
        $STR            ;STORE IT
        K.Ø16
        $IR             ;FLOAT 1Ø
        $LOD            ;LOAD 1ØØØ
        K.Ø16
        $MLR            ;COMPUTE 1Ø,ØØØ
        $STR            ;STORE IT
        K.Ø16
        $IR             ;FLOAT 16
        $LOD            ;LOAD 1ØØØ
        K.Ø16
        $DVR            ;COMPUTE .Ø16
        $STR            ;STORE IT
        K.Ø16
        $IR             ;FLOAT 1ØØØ
        $STR            ;STORE IT
        K.Ø13
        $IR             ;FLOAT 13
        $LOD            ;LOAD 1ØØØ
        K.Ø13
        $DVR            ;COMPUTE .Ø13
        $STR            ;STORE IT
        K.Ø13
        .WORD .+2
        CLR R2
        MOV #1Ø.,RØ             ;SET UP RØ FOR CLR
        MOV #KEINØ.,R1          ;SET UP R1 FOR CLR
CLR:    CMP R2,RØ               ;END OF LOOP?
        BEQ GO
        CLR (R1)+               ;CLR LOCATION
        INC R2                  ;INC LOOP CNTR
        BR CLR
GO:     CLR RØ
        CLR R1
        CLR R2
        JSR R4,$POLSH           ;ENTER POLISH MODE
        $IR             ;FLOAT 1ØØ
        $STR            ;STORE IT
        KCOT1.
        $IR             ;FLOAT 771
        $LOD            ;LOAD 1ØØ
        KCOT1.
        $DVR            ;COMPUTE 7.71
        $STR            ;STORE IT
        KCOT1.
        $LOD            ;LOAD IT
        KCOT1.
        $STR            ;STORE IT IN COT2
```

```
        KCOT2.
        .WORD .+2           LEAVE POLISH MODE
        MOV #1ØØ,TKS        ;SET UP TTY INTERRUPT
        MOV #17Ø.,CDB       ;SET UP SAM. RATE
        MOV #113,CSR        ; START CLOCK
        CLR COMP            ;CLR COMPENSTAOR FLAG
LOOP:   BR LOOP             ;WAIT FOR INTERRUPTS
        ;DEFINE THE ADDRESSES
        ;FOR THE FLOATING POINT
        ;SUBROUTINES
        $SBR=33ØØØ
        $ID=34144
        $DI=3467Ø
        $ADR=33ØØ4
        $IR=3416Ø
        $RI=347ØØ
        $DVR=33526
        $MLR=34262
        $POLSH=35Ø3Ø
        $ERR=35Ø34
        $ERRA=35Ø44
        $ERVEC=35Ø64
        ;
        ;
        ;
TTYIN:  MOV TKB,RØ          ;GET CHARACTER
        JSR PC,ECHO         ;ECHO CHARACTER
        BIC #1776ØØ,RØ      ;CLR TRASH
        CMP #122,RØ         ;"R"?
        BEQ RROUT           ;YES BR RROUT
        CMP #1Ø3,RØ         ;"C"?
        BEQ CROUT           ;YES BR CROUT
        CMP #113,RØ         ;"K"?
        BEQ KROUT           ;YES BR KROUT
        JSR PC,SCRUP        ;NO SCREWUP
        ;
        ;
        ;
RROUT:  CMP #Ø1,COMP        ;COMPENSATED?
        BEQ BACK            ;YES BR BACK
        CLR CSR             ;DISABLE CLOCK
        MOV #INPUT,6Ø       ;RESET TTY VECTOR
        MOV R2,#R2SAV       ;SAVE R2
        R2SAV=.-2
        MOV R3,#R3SAV       ;SAVE R3
        R3SAV=.-2
        CLR R1
        CLR R2
        CLR R3
```

```
        INC #RFLAG          ;INDICATE REF FLAG
        RFLAG=.-2
BACK:   RTI
        ;
        ;
        ;
CROUT:  CMP #Ø1,COMP        ;ALREADY COMPENSATED?
        BEQ OUT1            ;YES BR OUT1
        INC #COMP           ;INDICATE COMPENSATOR FLAG
        COMP=.-2
OUT1:   JSR PC,READY        ;RETURN TO PROGRAM
        ;
        ;
        ;
KROUT:  CLR COMP            ;REMOVE COMPENSATOR
        JSR PC,READY        ;RETURN TO PROGRAM
        ;
        ;
        ;
ECHO:   MOV RØ,TPB          ;ECHO ON TTY
ECHOIN: TSTB TPS            ;PRINTED?
        .WORD 24Ø
        RTS PC
        ;

        ;
        ;THIS IS THE SERVICE
        ;ROUTINE FOR THE CLOCK INTERRUPTS.
        ;IT CALCULATES THE ERROR SIGNAL
        ;THAT IS APPLIED TO THE PWM
        ;CONTROLLER
CLOCK:  INC ADSR            ;SAMPLE SIGNAL
        TSTB ADSR           ;SAMPLED?
        BPL .-4
        CMP #Ø1,COMP        ;COMPENSATOR FLAG
        BEQ NOW             ;YES BR NOW
        JMP NOTN            ;NO BR NOTN
NOW:    MOV ADDB,RØ         ;GET SAMPLE
        MOV SETPT,-(SP)     ;SETPT TO STACK
        MOV RØ,-(SP)        ;SAMPLE TO STACK
        JSR R4,$POLSH       ;ENTER POLISH MODE
        $IR      ;FLOAT SAMPLE
        $LOD     ;LOAD SCALE FACTOR
        K1Ø24.
        $MLR     ;SCALE SAMPLE
        $STR     ;STORE SAMPLE
        KSAM
        $IR      ;FLOAT REFERENCE
        $LOD     ;LOAD SCALE FACTOR
```

```
K1Ø24.
$MLR     ;SCALE REFERENCE
$LOD     ;LOAD SAMPLE
KSAM
$SBR     ;CALCULATE ERROR(EIN)
$STR     ;SAVE PRESENT EIN
KEINØ.
$LOD     ;LOAD EIN
KEINØ.
$LOD     ;LOAD .Ø16
K.Ø16
$MLR     ;CALC. .Ø16*EIN(NT)
$LOD     ;LOAD EIN(N-1)T
KEIN1.
$LOD     ;LOAD .ØØØ23
K.ØØØ23
$MLR     ;CALC .ØØØ23*EIN(N-1)T
$ADR     ;ADD TWO TERMS
$LOD     ;LOAD 2.Ø
K2.ØØØ
$LOD     ;LOAD COT(N-1)T
KCOT1.   ;
$MLR     ;CALC. 2.Ø*COT(N-1)T
$ADR     ;ADD 3 TERMS
$LOD     ;LOAD .Ø16
K.Ø16
$LOD     ;LOAD EIN(N-2)T
KEIN2.
$MLR     ;CALC. .Ø16*EIN(N-2)T
$LOD     ;LOD COT(N-2)T
KCOT2.
$ADR     ;ADD TWO NEGATIVE TERMS
$SBR     ;SUBTRACT NEG. TERMS
$LOD     ;
KCOT1.
$STR     ;UPDATE COT2
KCOT2.
$STR     ;UPDATE COT1
KCOT1.
$LOD
KEIN1.
$STR     ;UPDATE EIN2
KEIN2.
$LOD
KEINØ.
$STR     ;UPDATE EIN1
KEIN1.
$LOD     ;GET PRESENT VALUE OF COT
```

```
            KCOT1.
            $LOD     ;LOAD SCALE FACTOR
            K2Ø48.
            $MLR     ;CONVERT TO COUNTS
            $RI      ;CONVERT TO INTEGER
            .WORD .+2
            MOV (SP)+,DAB2  ;OUTPUT ON D/A
NOTN:       RTI
            ;
            ;
            ;
K2Ø48.:     .WORD Ø,Ø
K1Ø24.:     .WORD Ø,Ø
KEINØ.:     .WORD Ø,Ø
KEIN1.:     .WORD Ø,Ø
KEIN2.:     .WORD Ø,Ø
KCOT1.:     .WORD Ø,Ø
KCOT2.:     .WORD Ø,Ø
XR5:        .WORD Ø
KSAM:       .WORD Ø,Ø
K2.ØØØ:     .WORD Ø,Ø
K.Ø16:      .WORD Ø,Ø
K.ØØØ23:          .WORD Ø,Ø
K.Ø13:      .WORD Ø,Ø

$LOD:       MOV R5,XR5
            MOV (R4)+,R5     ;MOV ADDR TO R5
            ADD #4,R5        ;CORRECT FOR LOW MANT
            MOV -(R5),-(SP)  ;LOW MANT TO STACK
            MOV -(R5),-(SP)  ;SGN EXP TO STACK
            MOV XR5,R5       ;RESTORE R5
            JMP @(R4)+       ;NEXT ROUTINE
            ;
            ;
$STR:       MOV R5,XR5       ;STORE R5
            MOV (R4)+,R5     ;MOV ADDR TO R5
            MOV (SP+,(R5)+   ;STORE SGN,EXP
            MOV (SP)+,@R5
            MOV XR5,R5       ;RESTORE R5
            JMP @(R4)+       ;NEXT ROUTINE
            ;
INPUT:      MOV TKB,RØ       ;MOV DATC TO RØ
            JSR PC,FKUP      ;VALID CHARA?
            CMP #Ø1,R2       ;FIRST DIGIT?
            BEQ DIG1         ;YES BR DIG1
            CMP #Ø2,R2       ;2ND DIGIT?
            BEQ DIG2         ;YES BR DIG2
            CMP #Ø3,R2       ;3RD DIGIT?
```

```
            BEQ DIG3          ;YES BR DIG
            TST REMAX         ;REF MAX?
            BEQ LAST          ;NO BR LAST
            CMP #Ø6Ø,RØ       ;YES ZERO?
            BEQ ELP4          ;YES BR ELP4
            JSR PC,SCRUP      ;SCREWUP
LAST:       BIC #17776Ø,RØ    ;CLR TRASH
            ASL RØ            ;FIX FOR INDEX
            ADD TBL3(RØ),R1   ;ADD IN NO.
ELP4:       MOV R1,#RELOC     ;CHNG REFERENCE
            RELOC=.-2
            MOV #Ø15,TPB      ;CARRIAGE RETN
TSTA:       TSTB TPS          ;PRINTED?
            BPL TSTA
            MOV #Ø12,TPB      ;LINE FEED
TSTB:       TSTB TPS          ;PRINTED?
            BPL TSTB
            MOV #Ø52,TPB      ;"*"
TSTC:       TSTB TPS          ;PRINTED?
            BPL TSTC
            MOV RELOC,RØ      ;
            ASL RØ            ;FIX FOR D/A
            MOV RØ,DAB2       ;OUTPUT ON D/A
            MOV #113,CSR      ;RE-ENABLE CLOCK
            MOV #TTYIN,6Ø     ;RESET TTY VECTOR
            MOV R2SAV,R2      ;RESTORE R2
            MOV R3SAV,R3      ;RESTORE R3
            CLR R1            ;CLR OLD VALUE
            CLR RFLAG         ;CLR REF FLAG
            CLR REMAX         ;CLR REF MAX
OUT:        RTI
            ;
            ;
            ;

DIG 1:      CMP #Ø6Ø,RØ       ;CHECK FOR ZERO
            BEQ OUT           ;YES BR OUT
            CMP #Ø61,RØ       ;CHECK FOR ONE
            BEQ LP1           ;YES BR LP1
            JSR PC, SCRUP     ;SCREWUP
LP1:        MOV #1Ø23.,R1     ;ADD IN NO.
            INC #REMAX        ;MAX FLAG
            REMAX=.-2
            BR OUT
            ;
            ;
            ;
DIG2:       TST REMAX         ;MAX FLAG?
```

```
        BEQ STRT        ;NO BR STRT
        CMP #Ø6Ø,RØ     ;ZERO?
        BEQ OUT
        JSR PC,SCRUP    ;NO SCREWUP

STRT:   BIC #17776Ø,RØ  ;CLR TRASH
        ASL RØ          ;FIX FOR INDEX
        ADD TBL1(RØ),R1 ;ADD IN NO.
        BR OUT
        ;
        ;
        ;
DIG3:   TST REMAX;      ;MAX FLAG?
        BEQ STRT1       ;NO BR STRT1
        CMP #Ø6Ø,RØ     ;YES ZERO?
        BEQ OUT         ;YES BR OUT
        JSR PC,SCRUP    ;NO SCREWUP
STRT1:  BIC #17776Ø,RØ  ;CLR TRASH
        ASL RØ          ;FIX FOR INDEX
        CDD TBL2(RØ),R1 ;ADD IN NO.
        BR OUT
        ;
        ;
        ;
TBL1:   .WORD Ø,146,314,462,63Ø,776,1144
        .WORD 1312,146Ø,1626
TBL2:   .WORD Ø,12,24,36,5Ø,62,74,1Ø6,12Ø,132
TBL3:   .WORD Ø,1,2,3,4,5,6,7,1Ø,11
FKUP:   MOV RØ,TPB              ;ECHO
TSTD:   TSTB TPS                ;ECHOED?
        BPL TSTD
        BIC #1777ØØ,RØ          ;RID OF TRASH
        CMP #Ø56,RØ             ;DECIMAL?
        BEQ RITE                ;YES BR RITE
        INC R2                  ;INC NO. CNTR
        CMP #Ø6Ø,RØ             ;ZERO?
        BEQ RETN                ;YES BR RETN
        CMP #Ø61,RØ             ;ONE?
        BEQ RETN                ;YES BR RETN
        CMP #Ø62,RØ             ;TWO?
        BEQ RETN                ;YES BR RETN
        CMP #Ø63,RØ             ;THREE?
        BEQ RETN                ;YES BR RETN
        CMP #Ø64,RØ             ;FOUR?
        BEQ RETN                ;YES BR RETN
        CMP #Ø65,RØ             ;FIVE?
        BEQ RETN                ;YES BR RETN
        CMP #Ø66,RØ             ;SIX?
```

```
        BEQ RETN            ;YES BR RETN
        CMP #Ø67,RØ         ;SEVEN?
        BEQ RETN            ;YES BR RETN
        CMP #Ø7Ø,RØ         ;EIGHT?
        BEQ RETN            ;YES BR RETN
        CMP #Ø71,RØ         ;NINE?
        BEQ RETN
SCRUP:  CLR R2              ;CLR CNTR
        MOVB #Ø15,TPB       ;CARRIAGE RETN
TST:    TSTB TPS            ;IS CHARA PRINTED
        BPL TST             ;+NO,-YES
        MOVB #Ø12,TPB       ;LINE FEED
TST1:   TSTB TPS            ;PRINTED?
        BPL TST1
        MOVB #Ø77,TPB       ; ?
TST2:   TSTB TPS            ;PRINTED?
        BPL TST2
READY:  MOVB #Ø15,TPB       ;CARRIAGE RETN
TST3:   TSTB TPS            ;PRINTED?
        BPL TST3
        MOVB #Ø12,TPB       ;LINE FEED
TST4:   TSTB TPS            ;PRINTED?
        BPL TST4
        MOVB #Ø52,TPB       ; *
TST5:   TSTB TPS            ;PRINTED
        BPL TST5
        CLR RFLAG
        CLR REMAX
        MOV #TTYIN,6Ø       ;RESET TTY VECTOR
        MOV #113,CSR        ;START CLOCK
DECI:   MOV (SP)+,RØ                ;CLR OFF OLD PC
        RTI
RETN:   RTS PC
        ;
        ;
RITE:   CMP #Ø1,RFLAG       ;REF FLAG?
        BEQ DECI2
        BR SCRUP
DECI2:  CMP #Ø2,R2          ;IN PLACE?
        BEQ DECI            ;YES BR DECI
        BR SCRUP
        ;
        ;
        ;
PDN:    MOV #PUP,Ø24        ;RESET VECTOR
        MOV CDB,-(SP)       ;SAVE SAM. RATE
        MOV RØ,-(SP)        ;SAVE RØ
```

```
        MOV R1,-(SP)      ;SAVE R1
        MOV R2,-(SP)      ;SAVE R2
        MOV R3,-(SP)      ;SAVE R3
        MOV R4,-(SP)      ;SAVE R4
        MOV R5,-(SP)      ;SAVE R5
        MOV SP,#R6SAV     ;SCVE SP
        R6SAV=.-2
        HALT
        ;
        ;
        ;
PUP:    MOV #PDN,Ø24      ;RESET VECTOR
        MOV R6SAV,SP      ;RESET STACK PTR.
        MOV (SP)+,R5      ;RESTORE R5
        MOV (SP)+,R4      ;RESTORE R4
        MOV (SP)+,R3      ;RESTORE R3
        MOV (SP)+,R2      ;RESTORE R2
        MOV (SP)+,R1      ;RESTORE R1
        MOV (SP)+,RØ      ;RESTORE RØ
        MOV (SP)+,CDB     ;RESTORE CLK RATE
        MOV #113,CSR      ;SET UP CLOCK
        MOV #1ØØ,TKS      ;SET UP TTY FOR INTER.
        RTI
        ;
        ;
        ;
        .END              SPCTL
```

VITA

John M. Hickey III was born in Knoxville, Tennessee, on May 29, 1951. He attended elementary school in Knoxville, Tennessee and Sevierville, Tennessee and graduated from Sevier County High School in 1969. He entered the University of Tennessee in July of 1969 and received his Bachelor of Science in Electrical Engineering in August of 1973. In September of 1973, he entered the Graduate School at the University of Tennessee to continue his studies toward a Master of Science degree in Electrical Engineering.