# DEVELOPMENT OF A DYNAMIC MODEL OF A MECHANICALLY DRIVEN POLYPHASE AC GENERATOR

by

**H. Scott Coombe**

Major Paper submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

**MASTER OF ENGINEERING**

in

Mechanical Engineering

**APPROVED:**

RG Mitchiner
R.G. Mitchiner, Chairman

Alfred L Wicks
A.L. Wicks

R.G. Leonard

April, 1992

Blacksburg, VA

# DEVELOPMENT OF A DYNAMIC MODEL OF A MECHANICALLY DRIVEN POLYPHASE AC GENERATOR

by

H. Scott Coombe

Committee Chairman: R.G. Mitchiner
Mechanical Engineering

(ABSTRACT)

This paper presents a computer model developed for the purpose of solving the differential equations of a three-phase, rotating field, salient pole, synchronous AC generator driven by a variable mechanical power source. The goal of this model is to provide a convenient means to accurately predict the mechanical and electrical dynamics of any engine-generator combination of the aforementioned design-type under various resistive and inductive electrical loading conditions. Intended practical uses of the model are for mechanical failure prediction, transient studies, and analysis for determining the suitability of the electrical power quality output of the generator for intended electrical loads.

## ACKNOWLEDGEMENTS

Many thanks to Dr. R.G. Mitchiner for providing guidance on various aspects of the development of the computer model, especially in the area of plotting routines.

Thanks also go to Mr. Jim Wright, an applications engineer at Marathon Electric Corporation for detailed performance data on their extensive product line of generators.

I appreciate the support given by several individuals at the US Army Belvoir Research, Development, and Engineering Center, Ft. Belvoir, VA.

I also appreciate the assistance from my wife, Maria.

# TABLE OF CONTENTS

1. Title Page i

2. Abstract ii

3. Acknowledgements iii

4. Table of Contents iv

5. Introductory Section 1

6. Review of Literature 2

7. Materials and Methods 6

8. Results / Discussion 21

9. Conclusions 23

10. Summary 24

11. Literature Cited 25

12. Appendix A 28

13. Appendix B 43

14. Appendix C 51

15. Appendix D 64

16. Vita 67

# INTRODUCTION

The computer model developed herein is a significant step in the overall goal of developing a model of an entire power system, in which there exists an internal combustion engine driven generator powering induction motor driven reciprocating machinery. Currently, a mature computer model exists for an induction motor powering various loads. The utility in modelling an entire power system is to accurately replicate the transmission of the effects of the reciprocating engine throughout the system and the reflection of load reciprocating machinery disturbances throughout the electrical side of the system. In this paper we will discuss, in detail, the development of a computer model which accurately simulates the electrical output performance of an AC generator based on variable mechanical inputs.

We will address the derivation of the engine-generator computer model in terms of the mathematical model chosen from the literature to represent the physical system of interest. The structure / design of the computer program is discussed, as well as how well the model output correlates to actual system performance.

Capabilities of the resulting model are demonstrated and practical uses of the model are discussed. Examples of mechanical and electrical output are provided and analyzed. Requirements for data input are itemized.

Finally, after demonstrating its capabilities, we will recommend future enhancements to the Engine-Generator model and steps necessary to achieve the ultimate goal of an entire power system model.

# REVIEW OF LITERATURE

The literature search for a mathematical model began first by identifying the features & capabilities of interest to this project. As a minimum, we are interested in the following characteristics:

a) Equations to describe the Electrical & Mechanical behavior of an AC generator

b) Synchronous-type generator

c) Rotating Field

d) Salient Pole Rotor Design

e) Resistive & Inductive Loading

f) Magnetic Saturation Compensation

g) Amortisseur / Damper Windings

h) Polyphase output

i) Transient & Steady-State Analysis

j) Generalized to allow greatest flexibility of design

It is relevant to first discuss the original derivation of the electrical and mechanical equations of synchronous machines. The individual to first pioneer this work was R.H. Park [1] of General Electric in the Journal of The American Institution of Electrical Engineers, now known as, simply, The Institution of Electrical Engineers. Although his "Two Reaction Theory of Synchronous Machines - Generalized Method of Analysis" was first published in 1929, it is still referenced in almost every related paper and textbook to date. This is attributed to the fact that the design of synchronous machines has, for the most part, been an evolutionary process, i.e. the basic principles of design are the same as they were in the early part of this century. In fact, R.H. Park's analysis has been widely used as a foundation for understanding and modelling of synchronous AC generators for the past 60 years. His mathematical model is very generalized, in fact it accounts for most of the above objectives, except magnetic saturation and the mechanical equations of motion.

The next step in identifying an acceptable model was to search recent literature from several journals and textbooks. The two journals found to be most helpful are:

"IEEE Transactions on Power Apparatus and Systems" [1-6],

and,

"The Journal of The (American) Institution of Electrical Engineers" [7-12]

Though all of the papers from the above organizations have their respective benefits, each also is somewhat undesirable for the intent of this paper. Most of the papers involve stability studies for large scale turbine or hydroelectric power plants utilizing high speed AC generators with solid cylindrical rotors. The models were developed by either power companies or companies such as General Electric or Westinghouse who build hardware for large scale commercial power plants. The assumption is made in these papers that the electric power system itself is an "infinite bus", therefore one would expect variations in electrical output due to changes in load, but no variations in frequency. Some of the recent papers in this category, though, such as [7] and [8] begin to provide numerical solution approaches to solving the differential equations of interest.

At this point, we have identified a multitude of mathematical models which suit most of the parameters, but none for small scale systems.

An article from IEEE Transactions on Magnetics [13] presented a mathematical model developed for small systems, but is a simplified model based on an oscillator approach. However, there is some helpful information on modelling the mechanical system.

Finally, after searching through numerous IEE and IEEE papers on the subject, "The Computer Journal" was found to contain an excellent paper on the subject. A paper entitled "Numerical solution of the differential equations of a synchronous generator", by I.R. Smith and L.A. Snider [14] was presented and was developed for a small scale power system, namely a 15 kVA generator. This model satisfies all the aforementioned criteria, except for modelling of the mechanical equations of motion. In addition it presents a suggested numerical integration approach. It also presents a comparison of theoretical vs. empirical results and includes actual design data from a "small" (15 kVA) generator. This is the model selected for the electrical portion of the AC generator.

The search for a mechanical mathematical model proved to be less time consuming. There are various textbooks on electric machinery which present a thorough analysis of the mechanical equations involved, namely the references "Transient Analysis of AC Machinery" [15], "Synchronous Machines" [16], and "Electric Machines: Dynamics and Steady State" [17]. The latter was chosen because of its classical presentation of the mechanical equations of motion.

In summary, the mathematical model for the electrical system is,

> **"Numerical solution of the differential equations of a synchronous generator" [14]**

and the mathematical model for the mechanical system is,

> **"Electric Machines: Dynamics and Steady State" [17]**

These models were chosen because they are sufficiently general in nature and satisfy the criteria for achieving the stated project objectives.

## MATERIALS AND METHODS

In this section we will discuss three basic topics:

1) the development of the Physical Model
2) the development of the Electrical Model, and
3) the development of the Fortran Code

### Physical Model

The first step in the modelling effort was to identify and understand the physical system to be modeled. The overall system to be modeled is as shown in Figure 1.

Isochronous Governor. An isochronous design as opposed to a mechanical droop-type governor was chosen because many modern electric power systems are of this type, especially those with output capacities of at least 15 kVA used where high quality power needs are anticipated. All US Army mobile electric power generators currently being produced (with ratings above 10 kVA) are of the isochronous governor type. Electronic governor control units / actuators are used to achieve constant speed (no-droop) operation. The governor is simulated in the computer model by constantly monitoring the rotor speed and its deviation from rated speed. As a result of this speed difference, a corrected Torque is delivered by the engine to compensate. Commercial electronic governors typically have, among other controls, a "gain" setting. This gain setting basically determines the magnitude of the output response to the engine. Therefore, a gain setting has been incorporated into the computer model which results in a governor response (i.e. engine torque response) as follows:

$$T_{mech(n+1)} = T_{mech(n)} - (rpm_{n+1} - rpm_n)(Gain)$$

Voltage Regulator. The voltage regulator module developed incorporates the functions of a normal electronic regulator and a DC excitation system

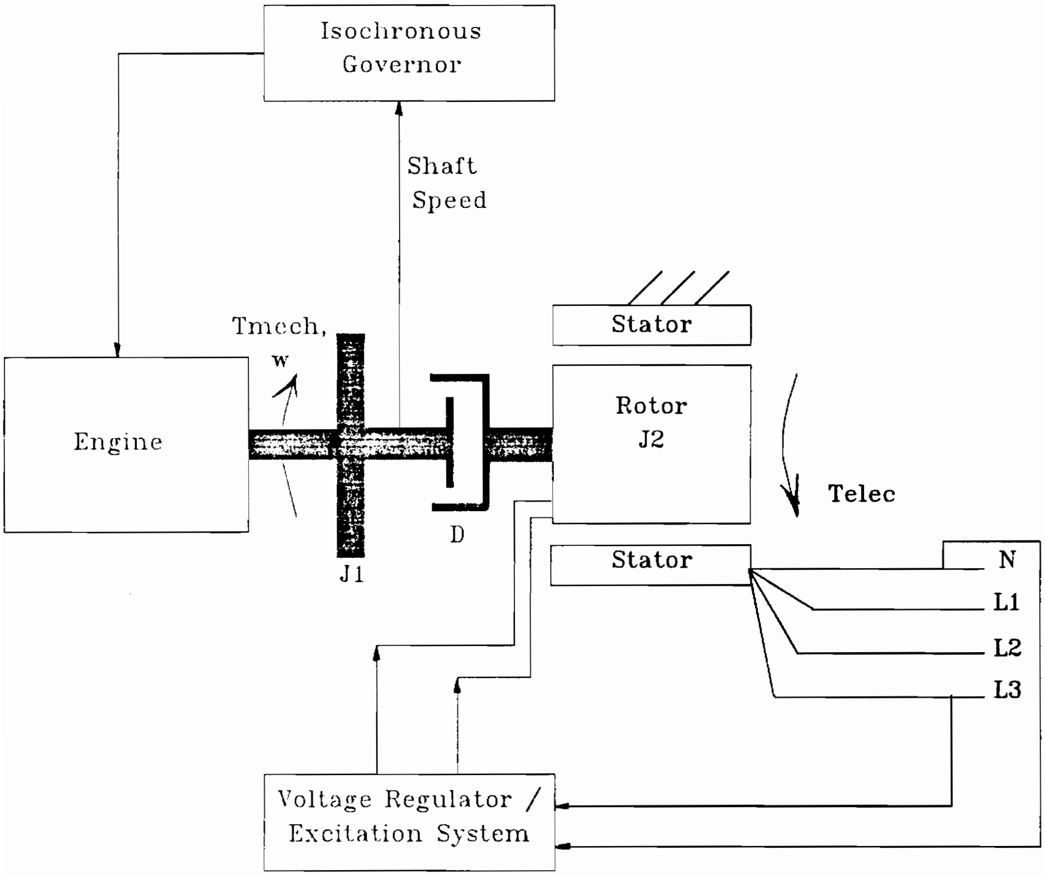


Figure 1 – Physical System to be Modelled

which supplies voltage to the main rotor field. The regulator is simulated in the computer model by constantly monitoring the sinusoidal output voltage of one phase of the output. The sinusoidally varying voltage of one phase is converted to a root-mean-squared equivalent (equivalent energy) such that it can be compared to the rated output, 120 volts, 240 volts, etc. As a result of this voltage difference, a corrected DC signal is sent to the rotor field. Similar to the electronic governor, commercial voltage regulators incorporate a "gain" setting which basically controls the magnitude of the output response. The voltage regulator relation used in this computer model is:

$$v_{DC(n+1)} = v_{DC(n)} - (v_{output\ rms(n+1)} - v_{output\ rms(n)})(Gain)$$

Based on the literature, a time delay of one period has been incorporated to simulate the delayed response in generating the new excitation voltage to the field.

Mechanical Equation of Motion. The system can be modelled adequately using a second order differential equation approach which takes into account the mass moment of inertia of the system, viscous damping, the electrical disturbing torque induced between the rotor and stator due to electrical loads, and the synchronizing mechanical torque provided by the engine:

$$(J_1 + J_2)d^2\theta/dt^2 + (D)d\theta/dt + T_{mech} = T_{elec}$$

where, $\theta$ = the angular displacement of the direct axis of the rotor with respect to a fixed reference coordinate system

J1 = the mass polar moment of inertia of the engine / flywheel combination.

J2 = the mass polar moment of inertia of the generator rotor and shaft combination.

D = the damping coefficient, including mechanical viscous friction plus electrical viscous damping from the field coil and damper coils.

Generator Design. Figure 2 depicts both a uniform gap (cylindrical rotor) design and a salient pole rotor design. Both are 4-pole machines. Because we are interested in a reciprocating engine driven generator, the salient pole design is the best choice for our analysis. Additional justification for this choice is as follows:

- The salient pole generator is the most common design by far in use today in low-speed, reciprocating engine applications
- The salient pole generator is significantly less expensive to manufacture
- The cylindrical construction is used in high-speed systems only because the salient pole type is difficult to build to withstand the stresses at high speeds and has high winding losses at high speeds. [18]

Other design choices are incorporated such as amortisseur ("damper") windings. Because reciprocating engines sometimes tend to "hunt", damper windings are often placed on the surface of rotors to decrease this tendency. The dampers are nothing more than short-circuited conductors imbedded into the pole faces. When hunting occurs, a shifting of the stator magnetic flux occurs, thereby inducing currents in the damper windings. Since any induced current opposes the action that produces it, the hunting action is opposed by the flow of the induced currents. [18] It is interesting to note that generator rotors driven by turbomachinery generally do not incorporate these windings because these systems do not have a tendency to hunt.

The baseline physical design for our particular model is similar to the

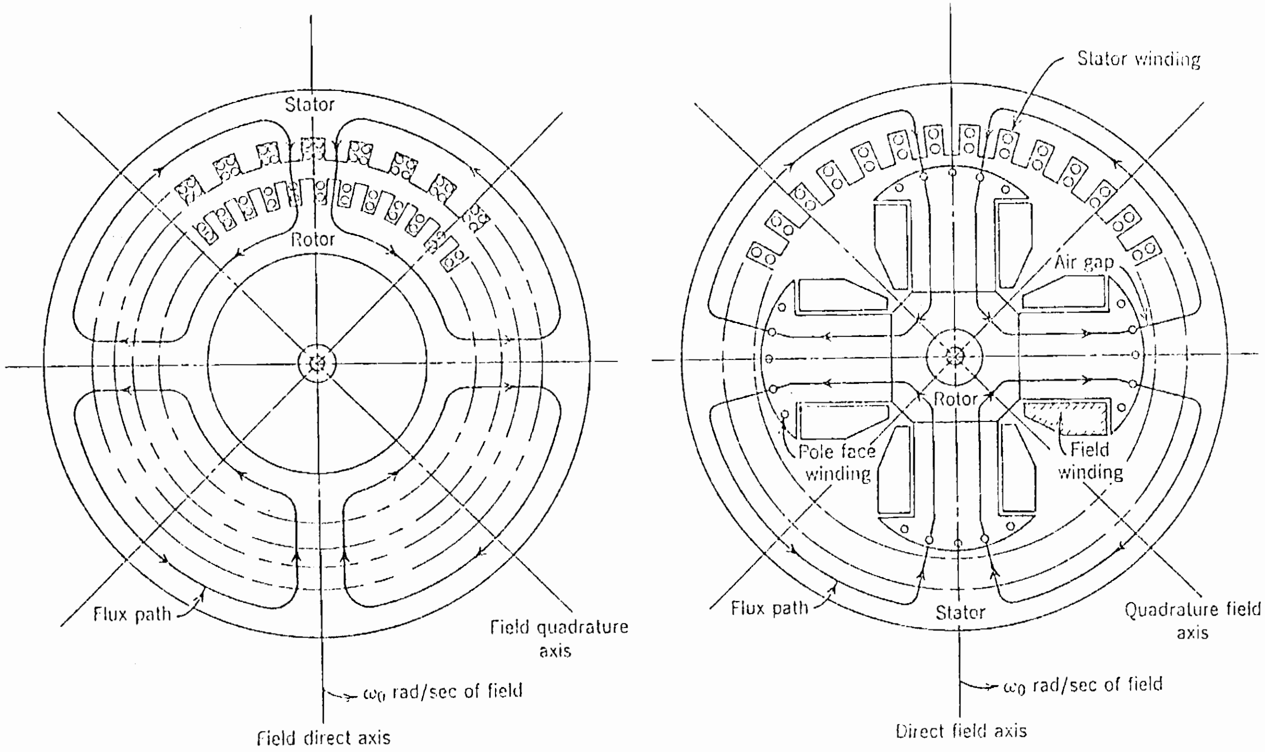


Figure 2 - Cylindrical Rotor Design (Left) and Salient Pole Rotor Design (Right)

rotating field, salient-pole design depicted in Figure 2 with the exception that there are six (6) poles instead of four (4), rotating field design with damper windings as shown. As the reader will notice later, the number of poles specified in the computer model developed herein is variable, but the initial analysis was conducted with a 6-pole design because the data was readily available for this design type to verify performance of the computer model. The equation relating the output frequency (Hz), shaft speed (rpm), and the number of rotor poles is as follows [18]:

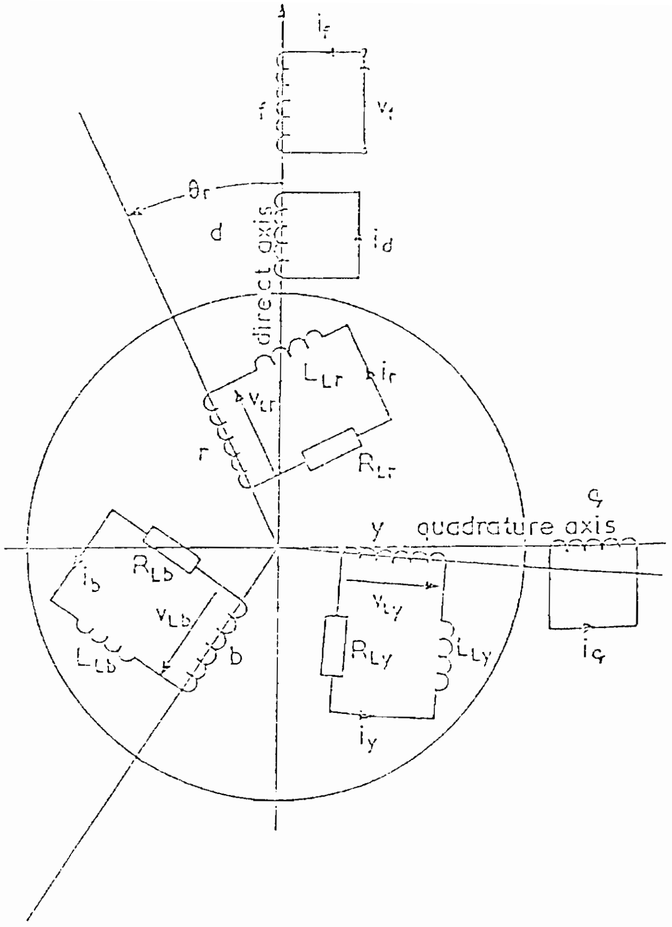
$$\textbf{Frequency} = \frac{\textbf{(Shaft Speed) (Number of Poles)}}{120}$$

The computer model permits the user to vary any of these three variables.

## Electrical Model

Now that the physical construction is specified, we turn to the electrical performance of the generator itself. A circuit model of the system is provided in Figure 3. Note that in this circuit model the load circuitry appears to be rotating. This is merely a convenient means of depicting the electrical circuit equivalent of the generator. Obviously, in a rotating field machine, the stator carries the load voltages and currents. Though Figure 3 adequately describes the physical circuits involved in our generator, it does not show the magnetic flux linkages responsible for producing the output voltage and current in the stator/armature. If our model is to be an accurate depiction of a real generator, these magnetic flux linkages must be incorporated.

The most important flux linkage is the mutual inductance between the rotor field circuit and the armature phase windings. However, there are several other inductance coefficients we must consider in a real system.
The inductance coefficients can be broken up into two categories, Self



$R_f$ = Resistance of Field Winding

$R_q$ = Resistance of Quadrature Axis Damper Windings

$R_d$ = Resistance of Direct Axis Damper Windings

$R_{Lr}, R_{Ly}, R_{Lb}$ = Resistances of Load

$L_{Lr}, L_{Ly}, L_{Lb}$ = Self Inductances of Load

$i_r, i_y, i_b$ = Armature Phase Currents

$i_d, i_y$ = Direct & Quadrature Axis Damper Currents

$v_f, i_f$ = Field Voltage & Current

$v_{Lr}, v_{Ly}, v_{Lb}$ = Load Voltages

$\theta_r, \theta_y, \theta_b$ = Angular Displacement of Armature Phase Windings from Direct Axis

**Figure 3 - Circuit Model of a 3-phase AC Generator**

Inductance and Mutual Inductance. They are applicable to the following windings:

**Self Inductance**

- Field Windings
- Armature Windings
- Amortisseur Windings

**Mutual Inductance**

- Armature phase-to-phase
- Armature-to-Amortisseur
- Armature-to-Field
- Field-to-Amortisseur

It is unnecessary to derive the relationships governing self- and mutual-inductance relations at this point. Park [1] Concordia [16] provide a more-than-thorough derivation of these relations. The equations governing self- and mutual-inductance will be presented herein as part of the matrix equations governing performance of the entire generator. Basically, each of the inductance relations vary sinusoidally and have a constant term which is determined empirically. The empirical term is largely a function of the non-uniform air gap between the salient-pole rotor and stator. Most generator manufacturers will provide this information although it is not part of their "boilerplate" or "Specification Sheet" information. At this point we are ready to present and solve the governing equations for our generator (see Figure 4). The matrix equation is in the form,

$$[v] = ([R] + [L]p)\ [i]$$

which can also be written

$$[v] = ([R] + [L]d/dt)\ [i].$$

At this point we have six (6) equations and six (6) unknowns, namely the six (6) currents,

$i_r$ - Armature Current (Phase 1)

$i_y$ - Armature Current (Phase 2)

$i_b$ - Armature Current (Phase 3)

$i_f$ - Field Current

$i_d$ - Direct Axis Damper Current

$i_q$ - Quadrature Axis Damper Current

It is interesting to note that, basically, the variation of $v_f$ (field voltage) is the driving force behind all the currents involved in the model.

Our goal at this point is to numerically solve the matrix equation of Figure 4 for all six (6) currents. First, we manipulate the above equation as follows;

$$[v] = [R][i] + [L]di/dt$$

$$[L]-1[v] = [L]-1[R][i] + di/dt$$

$$di/dt = [L]-1\ ([v] - [R][i])$$

This equation is of the form, $y' = f(x,y)$, and various established techniques are available for numerical solution of same. These numerical routines fall into basically two groups:

1. solution by successive substitution (Runga-Kutta methods); and
2. solution by numerical integration formulae (e.g. predictor-corrector methods).

Smith & Snider [14] recommend the use of Hamming's [19] predictor-corrector method because it is somewhat quicker, requiring less iterations for the same accuracy. However, Hamming's method is not self-starting, in fact four (4) previous values of the solution are required. Knowing the above and the author's lack of experience with Hamming's routines, a Runga-Kutta fourth-order routine [20] is used in this model. See Appendix A for the specific integration subroutine used.

$$
\begin{bmatrix} 0 \\ 0 \\ 0 \\ v_f \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix}
R_r + R_{Lr} + p(L_{Lr} + L_D\cos^2\theta_r + L_Q\sin^2\theta_r) & p(M_D\cos\theta_r\cos\theta_b + M_Q\sin\theta_r\sin\theta_b) & p(M_D\cos\theta_r\cos\theta_b + M_Q\sin\theta_r\sin\theta_b) & pM_f\cos\theta_r & pM_d\cos\theta_r & -pM_q\sin\theta_r \\
p(M_D\cos\theta_y\cos\theta_r + M_Q\sin\theta_y\sin\theta_r) & R_y + R_{Ly} + P(L_{Ly} + L_D\cos^2\theta_y + L_Q\sin^2\theta_y) & p(M_D\cos\theta_y\cos\theta_b + M_Q\sin\theta_y\sin\theta_b) & pM_f\cos\theta_y & pM_d\cos\theta_y & -pM_q\sin\theta_y \\
p(M_D\cos\theta_b\cos\theta_r + M_Q\sin\theta_b\sin\theta_r) & p(M_D\cos\theta_b\cos\theta_y + M_Q\sin\theta_b\sin\theta_b) & R_b + R_{Lb} + p(L_{Lb} + L_D\cos^2\theta_b + L_Q\sin^2\theta_b) & pM_f\cos\theta_b & pM_d\cos\theta_b & -pM_q\sin\theta_b \\
pM_f\cos\theta_r & pM_f\cos\theta_y & pM_f\cos\theta_b & R_f + pL_f & pM_{fd} & 0 \\
pM_d\cos\theta_r & pM_d\cos\theta_y & pM_d\cos\theta_b & pM_{fd} & R_d + PL_d & 0 \\
-pM_q\sin\theta_r & -pM_q\sin\theta_y & -pM_q\sin\theta_b & 0 & 0 & R_q + pL_q
\end{bmatrix} \bullet \begin{bmatrix} i_r \\ i_y \\ i_b \\ i_f \\ i_d \\ i_q \end{bmatrix}
$$

Figure 4 - Flux Linkage Equation

As part of the above equation, we have the need to numerically invert a (6 X 6) matrix as well. A Gauss-Jordan Reduction method [20] was used for this task (see Appendix A).

After solving for the three (3) armature phase currents, we can compute the load voltages as follows:

$$\begin{vmatrix} v_{Lr} \\ v_{Ly} \\ v_{Lb} \end{vmatrix} = \begin{bmatrix} R_{Lr} + L_{Lr}p & & \\ & R_{Ly} + L_{Ly}p & \\ & & R_{Lb} + L_{Lb}p \end{bmatrix} \cdot \begin{vmatrix} i_r \\ i_y \\ i_b \end{vmatrix}$$

where p = d/dt

Electric Power Output. Now that the armature voltages and currents have been calculated, we can express the instantaneous power output as follows:

$$\mathbf{P_{elec} = (v_{Lr})(i_r) + (v_{Ly})(i_y) + (v_{Lb})(i_b)}$$

which is accurate for both resistive and inductive loading.

Electrical Torque. The electromagnetic torque induced in the rotor from the armature currents can be expressed as a function of the armature phase voltages and currents as follows:

$$\mathbf{T_{elec} = \frac{(v_{Lr})(i_r) + (v_{Ly})(i_y) + (v_{Lb})(i_b)}{(eff)(d\theta/dt)}}$$

or, using the relation for power output, it may be expressed as follows:

$$\mathbf{T_{elec} = \frac{P_{elec}}{(eff)(d\theta/dt)}}$$

where, **eff** = the generator efficiency which may vary with load, and

**do/dt** = the angular velocity of the rotor shaft

## Fortran Code

The code developed for this model was written using a Fortran compiler developed for the personal computer. The source code is included in Appendix A. The logical flow of the program is shown in Figure 5.

Note near the end of the Figure 5, the variable, time, is incremented rather than the angle of the rotor. Through much experience with this model a "maximum practical" time step has been determined to be 4-5 time steps per half period of the fundamental frequency. This provides accurate output and a reasonable representation of a sine wave. In terms of computer hardware requirements for running this model, only a few are important, namely

1) It is advisable that a personal computer with at least an 80386 microprocessor operating at 20 MHz be used.
2) A math coprocessor is a must for the extensive floating point calculations necessary.

The model has been designed to be executed by either of two methods:

1) The user can input system data into the data file called "data.txt", or,
2) A user-friendly "front-end" was created which prompts the user to input system data which is then automatically written to the data file, "data.txt". The only requirement here is that the user must have the software program DBase III+. A sample of the user input screens are provided in Figure 6. Source code for this "front-end" is provided as Appendix B.

Two sets of plots are available after compiling the program:

1) <u>Electrical</u> Voltage (Phases 1-3)

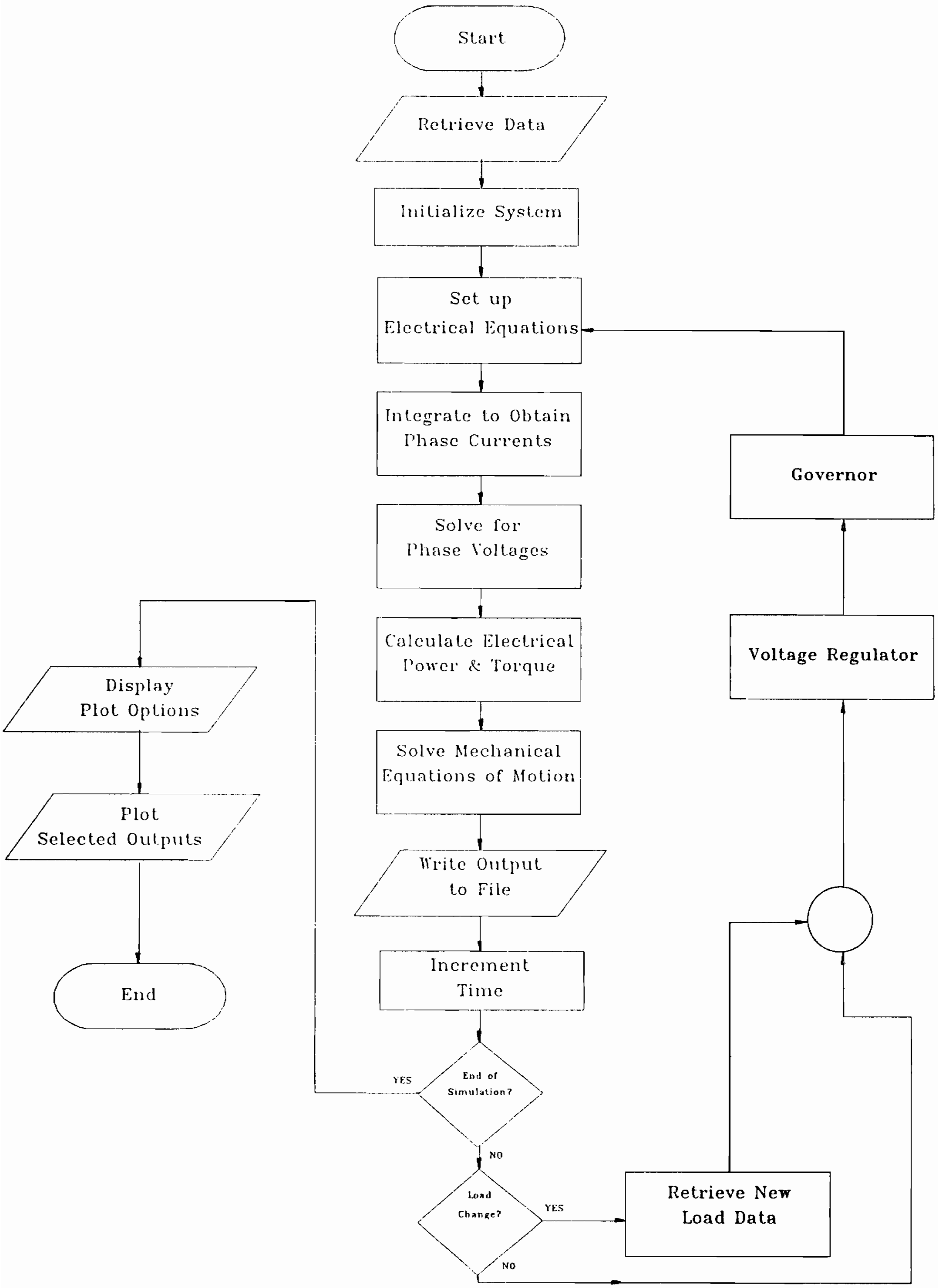
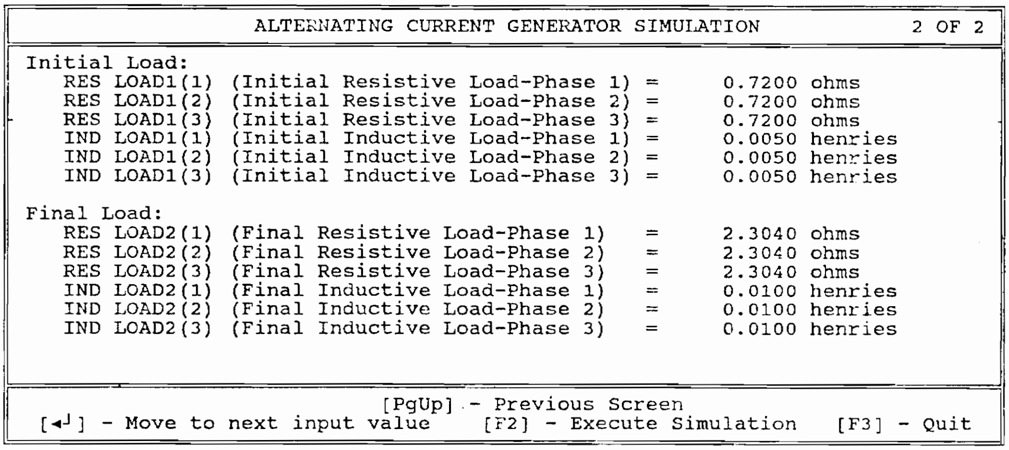


Figure 5 – Flowchart of Fortran Program

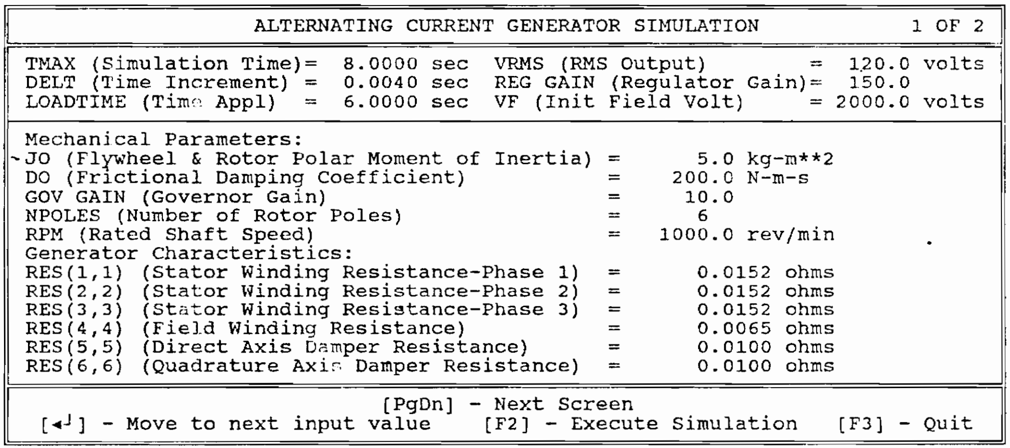


```
                 ALTERNATING CURRENT GENERATOR SIMULATION               2 OF 2

 Initial Load:
    RES LOAD1(1)  (Initial Resistive Load-Phase 1)  =      0.7200 ohms
    RES LOAD1(2)  (Initial Resistive Load-Phase 2)  =      0.7200 ohms
    RES LOAD1(3)  (Initial Resistive Load-Phase 3)  =      0.7200 ohms
    IND LOAD1(1)  (Initial Inductive Load-Phase 1)  =      0.0050 henries
    IND LOAD1(2)  (Initial Inductive Load-Phase 2)  =      0.0050 henries
    IND LOAD1(3)  (Initial Inductive Load-Phase 3)  =      0.0050 henries

 Final Load:
    RES LOAD2(1)  (Final Resistive Load-Phase 1)    =      2.3040 ohms
    RES LOAD2(2)  (Final Resistive Load-Phase 2)    =      2.3040 ohms
    RES LOAD2(3)  (Final Resistive Load-Phase 3)    =      2.3040 ohms
    IND LOAD2(1)  (Final Inductive Load-Phase 1)    =      0.0100 henries
    IND LOAD2(2)  (Final Inductive Load-Phase 2)    =      0.0100 henries
    IND LOAD2(3)  (Final Inductive Load-Phase 3)    =      0.0100 henries

                            [PgUp] - Previous Screen
  [◄┘] - Move to next input value    [F2] - Execute Simulation    [F3] - Quit
```



```
                 ALTERNATING CURRENT GENERATOR SIMULATION               1 OF 2

 TMAX (Simulation Time)=  8.0000 sec  VRMS (RMS Output)         =  120.0 volts
 DELT (Time Increment) =  0.0040 sec  REG GAIN (Regulator Gain)=  150.0
 LOADTIME (Time Appl)  =  6.0000 sec  VF (Init Field Volt)      = 2000.0 volts

 Mechanical Parameters:
 JO (Flywheel & Rotor Polar Moment of Inertia) =      5.0 kg-m**2
 DO (Frictional Damping Coefficient)           =    200.0 N-m-s
 GOV GAIN (Governor Gain)                      =     10.0
 NPOLES (Number of Rotor Poles)                =     6
 RPM (Rated Shaft Speed)                       =   1000.0 rev/min
 Generator Characteristics:
 RES(1,1) (Stator Winding Resistance-Phase 1)  =      0.0152 ohms
 RES(2,2) (Stator Winding Resistance-Phase 2)  =      0.0152 ohms
 RES(3,3) (Stator Winding Resistance-Phase 3)  =      0.0152 ohms
 RES(4,4) (Field Winding Resistance)           =      0.0065 ohms
 RES(5,5) (Direct Axis Damper Resistance)      =      0.0100 ohms
 RES(6,6) (Quadrature Axis Damper Resistance)  =      0.0100 ohms

                            [PgDn] - Next Screen
  [◄┘] - Move to next input value    [F2] - Execute Simulation    [F3] - Quit
```

**Figure 6 - User Input Screens**

| | |
|---|---|
| | Currents (Phases 1-3) |
| 2) <u>Mechanical</u> | Mechanical Torque |
| | Electrical Torque |
| | Frequency |
| | Shaft Acceleration |
| | Instantaneous Power |
| | Shaft Position |

## RESULTS / DISCUSSION

Model output is provided for the following loading condition:

- Generator is initially operating at no load.
- One step application of balanced 3-phase 15 kW (15 kVA @ 1.0 power factor), 100% resistive load.
- One step removal of balanced 3-phase 7.5 kW (7.5 kVA @ 1.0 power factor), 100% resistive load. Remaining load = 7.5 kW.
- Output is provided in Appendix C.

By reviewing this output, the reader can verify the following:

1) The voltage output ramps up due to increasing excitation voltage to the peak-to-peak value of 1.414(120.0) = 169.7 volts after both load changes.
2) The current in each phase reaches its rated load value of 58.9 amps peak-to-peak, 41.7 amps rms.
3) From the mechanical output, we can see the model is well-behaved, in that it responds to transients well. Note that the transient from the second load change does not show up on some of the plots. The model incorporates a "zoom" feature which will allow the user to zoom in on small transients.
4) Note that the Electrical Power Output plot approaches 15 kW as it should for rated load, 1.0 power factor.

To further verify performance of the model, the model was executed with a fairly large reactive load, simulating the effect of an inductive motor load (see Appendix D). The zoom feature was used here to show the effect an inductive load has on the voltage. The first thing the reader will notice is that the output is very nearly a sinusoid, but the significant point here is that the voltage trace for phase 1 is lagging the current in phase 1 as it should. The voltage and current are always in phase for purely resistive loads.

# CONCLUSIONS

The mathematical models chosen [14] and [17] coupled with the modified numerical solution approach used [14] resulted in an excellent modelling system for our interests. The model adequately represents the physical phenomena in a compact computer program which is fairly easy to use and provides results in a timely manner. From the testing conducted on the model to date the output is consistent with what we would normally expect, but the model will not be considered "mature" until tested and utilized by others over the long term.

Although useful at this point in its development, future enhancements should be made in the following areas:

1) The added capability of compensating for magnetic saturation. Saturation typically occurs near rated load and voltage is no longer a linear function of current. This feathre was not incorporated, but the mathematical model is available in [14]. When upgrading the model to account for saturation, it may be necessary to use a faster integration routine, such as Hamming's Method. [19]
2) Incorporate cyclic torque profile to simulate the actual torque variation which is provided by a reciprocating engine.
3) The ultimate goal, as stated previously, is to link the model provided herein with an existing motor model such that transient conditions may be traced throughout an actual power generating system.

# SUMMARY

From the extensive literature search conducted in support of this paper, it is clear that there is, and has been very little work in the area of modelling an engine-generator system of the type we are interested in here, namely small-to-medium scale power output, low speed, reciprocating engine driven. Most of the modelling efforts to date have been focused on large scale commercial power plants. It is the opinion of the author that much effort is needed in the area in which this paper has focused.

The reader should note that this model is not to be considered a final product which can be used to comprehensively predict the characteristics of an engine-generator system. There are limitations to uses of the model as have been discussed. The effort presented herein provides a useful tool, but should be considered a first step in the development of a model which will replicate the operation of an engine-generator under all possible uses.

It is interesting to note the many uses for this model. Obviously, transient studies, mechanical failure prediction, etc. will be the main uses, but other uses come to mind such as:

1) using the model as a design tool for generator, governor, voltage regulator manufacturers, and
2) as an instructional tool for students, industry, etc. I do not think this potential use should be overlooked, because it is very difficult to demonstrate, in a convenient and timely fashion, the operation of and engine-generator system.

# LITERATURE CITED

[1] R. H. Park, "Two-Reaction Theory of Synchronous Machines, Generalized Method of Analysis-Part I," Trans. AIEE, pp. 716-730, July 1929.

[2] C. Concordia, "Synchronous Machines with Solid Cylindrical Rotor-Part II," Trans. AIEE, pp. 1650-1657, February 1990.

[3] I. Gaiver, "A Complete Equivalent Circuit of a Synchronous Machine," Trans AIEE, pp. 204-209, June 1958

[4] D.B. Mehta and B Adkins, "Transient Torque and Load Angle of a Synchronous Generator Following Several Types of System Disturbance," Trans IEE, pp. 61-74, February 1960.

[5] D. B. Breedon, R. W. Ferguson, "Fundamental Equations for Analogue Studies of Synchronous Machines," Trans. AIEE, pp. 297-306, June 1956.

[6] R. H. Park, "Two-Reaction Theory of Synchronous Machines-II," Trans. AIEE, pp. 352-355, June 1933.

[7] G. Gross, M. C. Hall, "Synchronous Machine and Torsional Dynamics Simulation in the Computation of Electromagnetic Transients," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-97, No. 4, pp. 1074-1086, July/August 1978.

[8] P. Kundar, P. L. Dandeno, "Implementation of Advanced Generator Models into Power System Stability Programs," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-102, No. 7, July 1983.

[9] R. J. Wamser, I. W. Slitsker, "Power Flow Solution by the Newton-Raphson Method in Transient Stability Studies," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-103, No. 8, pp. 2299-2307, August 1984.

[10] C. C. Lee, O. T. Tan, "A Weighted-Least-Squares Parameter Estimator for Synchronous Machines," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-96, No. 1, pp. 97-101, January/February 1977.

[11] The IEEE Working Group on Computer Modelling of Excitation Systems,"Excitation System Models for Power Systems Stability Studies," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, No. 2, pp. 484-509, February 1981.

[12] J. M. Undrill, "Structure in the Computation of Power System Non-Linear Dynamical Response," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-88, No. 1, pp. 1-6, January 1969.

[13] B. Kaplan, D. Kottick, "Compact Representation of Synchronous Motors and Unregulated Synchronous Generators," IEEE Transactions on Magnetics, Vol. MAG-21, No. 6, pp. 2657-,2663 November 1983.

[14] I. R. Smith, L. A. Snider, "Numerical Solution of the Differential Equations of a Synchronous Generator," The Computer Journal, Vol. 14, No. 4, pp. 350-355, 1972.

[15] W. V. Lyon, "Transient Analysis of Alternating Current Machines," Wiley, New York, 1954.

[16] C. Concordia, "Synchronous Machines," Wiley, New York 1951.

[17] G. J. Thaler, M. L. Wilcox, "Electrical Machines: Dynamics and Steady State," John Wiley & Sons, Inc., 1966.

[18] E. C. Lister, "Electric Circuits and Machines," McGraw-Hill, 1960.

[19] R. Hamming, "Stable Predictor-Corrector Techniques for Ordinary Differential Equations," J. Assoc. Comp. Mach., Vol. 6, pp.37-47, 1959.

[20] M. L. James, G. M. Smith, J. C. Wolford, "Applied Numerical Methods for Digital Computation with FORTRAN and CSMP," Harper & Row, Publishers, Inc., 1967.

# **APPENDIX A**

## FORTRAN SOURCE CODE FOR DEVELOPMENT OF A DYNAMIC MODEL OF A MECHANICALLY DRIVEN POLYPHASE AC GENERATOR

```
C     ****************************************************************
C     * THIS PROGRAM SOLVES THE EQUATIONS OF A SYNCHRONOUS GENERATOR *
C     *                                                              *
C     * WRITTEN BY:  H. SCOTT COOMBE                                 *
C     *        FOR:  ENGR 5904                                       *
C     * INSTRUCTOR:  R.G. MITCHINER                                  *
C     *       DATE:  SUMMER I THRU SPRING 1992                       *
C     ****************************************************************
C     DEFINE VARIABLES USED DURING PROGRAM
C     ****************************************************************
C        RES(4,4),LF  =  RESISTANCE AND SELF INDUCTANCE OF FIELD
C                        WINDING
C           RES(1,1)  =  RESISTANCES OF ARMATURE PHASE WINDINGS
C           RES(2,2)
C           RES(3,3)
C              LD,LQ  =  DIRECT & QUADRATURE AXIS COEFFICIENTS OF
C                        ARMATURE PHASE SELF INDUCTANCE
C              MD,MQ  =  DIRECT & QUADRATURE AXIS COEFFICIENTS OF
C                        ARMATURE PHASE / PHASE MUTUAL INDUCTANCE
C  RES(5,5),RES(6,6)  =  RESISTANCES OF DIRECT & QUADRATURE AXIS
C                        DAMPER WINDINGS
C            LD1,LQ1  =  SELF INDUCTANCES OF DIRECT & QUADRATURE AXIS
C                        DAMPER WINDINGS
C            MD1,MQ1  =  DIRECT AND QUADRATURE AXIS COEFFICIENTS OF
C                        ARMATURE-PHASE / DAMPER MUTUAL INDUCTANCE
C                MFD  =  FIELD / DIRECT-AXIS-DAMPER MUTUAL INDUCTANCE
C                MF1  =  DIRECT-AXIS COEFFICIENT OF ARMATURE-PHASE /
C                        FIELD MUTUAL INDUCTANCE
C          LOAD1(1,1)  =  RESISTANCES OF INITIAL LOAD
C          LOAD1(2,2)
C          LOAD1(3,3)
C          LOAD2(1,1)  =  RESISTANCES OF FINAL LOAD
C          LOAD2(2,2)
C          LOAD2(3,3)
C        LL1,LL2,LL3  =  SELF INDUCTANCES OF LOAD
C            AMP2(1)  =  ARMATURE PHASE CURRENTS
C            AMP2(2)
C            AMP2(3)
C            ID1,IQ1  =  DIRECT & QUADRATURE AXIS DAMPER CURRENTS
C         VF,AMP1(4)  =  FIELD VOLTAGE & CURRENT
C            VOLT(1)  =  LOAD VOLTAGES
C            VOLT(2)
C            VOLT(3)
C                 ID  =  RESULTANT DIRECT-AXIS CURRENT
C                 IQ  =  RESULTANT QUADRATURE-AXIS CURRENT
C           A1,A2,A3  =  ANGULAR DISPLACEMENT OF ARMATURE PHASE
C                        WINDINGS FROM DIRECT-AXIS
C                RPM  =  REVOLUTIONS PER MINUTE
C              HERTZ  =  CYCLES PER SECOND
C               TMAX  =  RUN TIME FOR SIMULATION
C               DELT  =  TIME INCREMENT
C                     T  =  TIME ELAPSED
C              DELTA  =  ROTOR ANGLE INCREMENT
C           VCEILING  =  CEILING VOLTAGE OF VOLTAGE REGULATOR
C                 JO  =  MASS MOMENT OF INERTIA OF ROTOR
C                    DO  =  SHAFT DAMPING COEFFICIENT
C            ETORQUE  =  ELECTRICAL TORQUE
C            MTORQUE  =  MECHANICAL TORQUE FROM ENGINE
```

```
C     ****************************************************************
C     DECLARE VARIABLES
C     ****************************************************************
C
C     SETUP PLOTTING COMMON
C
      COMMON/PLOTTING/NSTORE,VPLT(4,2000),APLT(6,2000),DPLT(6,2000)
C
C     ****************************************************************
        REAL LF1,LD,LQ,MD,MQ,LD1,LQ1,MD1,MQ1,MFD,MF1,LL1,LL2,
     1   LL3,ID,IQ,IND(6,6),INDI(6,6),JO,
     1   LIND(3),LOADTOT(6,6),LASTA1,LDIND1(3),LDIND2(3),
     1   LOADTIME,MTK1(6),MTK2(6),MTK3(6),ZTK1(6),ZTK2(6),
     1   ZTK3(6),LOAD1(3),LOAD2(3),V0(6),LOAD(3),OLDA1,NPOLES,
     1  RESAMP(6),VRA(6),VOLT(6),AMP1(6),AMP2(6),AKS(6),X(3),
     1  U(6,6),V(6,6),RES(6,6),PREVOLT4,PEAK,
     1  IDS,MTORQUE,NEWFREQ,MAXTORQ,MAX,VR(3),VL(3),
     1  OLDTORQ
        PI=3.141592654
        TWOPI=2.*PI
        XL=TWOPI*FREQ
C     ****************************************************************
C     OPEN FILES TO BE USED DURING COURSE OF PROGRAM
C     ****************************************************************
      OPEN (UNIT=1,FILE='DATA.TXT',ACCESS='SEQUENTIAL',
     1      FORM='FORMATTED',STATUS='OLD')
C     ****************************************************************
C     READ IN DATA
C     ****************************************************************
      READ (1,*) TMAX,DELT,RPM,VF,NPOLES,LOADTIME,VRMS,
     1        RES(1,1),RES(2,2),RES(3,3),RES(4,4),RES(5,5),RES(6,6),
     1        LOAD2(1),LOAD2(2),LOAD2(3),
     1        LOAD1(1),LOAD1(2),LOAD1(3),
     1        LDIND2(1),LDIND2(2),LDIND2(3),
     1        LDIND1(1),LDIND1(2),LDIND1(3),
     1        JO,DO,GOVGAIN,REGGAIN
      CLOSE (UNIT=1)
C     ****************************************************************
C     ESTABLISH # OF ITERATIONS
C     ****************************************************************
      NOLOOPS=TMAX/DELT
C     ****************************************************************
C      ESTABLISH SAMPLING INTERVAL FOR PLOTTING
C
C     NCNT = 1  PLOT EVERY POINT IN THE PLOTTING ARRAYS
C            N  PLOT EVERY NTH POINT IN THE PLOTTING ARRAYS
C
C     NSTORE = COUNTER FOR CURRENT STORAGE LOCATION IN PLOTTING ARRAYS
C
C     ****************************************************************
      NSTORE=0
      NOUT=0
      IF(NOLOOPS.GT.2000) THEN
                            NCNT=IFIX(FLOAT(NOLOOPS/2000)+.5)
                         ELSE
                            NCNT=1
                         ENDIF
C
```

```
C   ZERO PLOTTING ARRAYS
C
      DO 10 I=1,2000
      DO 10 K=1,6
      IF(K.GE.5) THEN
                   APLT(K,I)=0.
                   DPLT(K,I)=0.
               ELSE
                   APLT(K,I)=0.
                   VPLT(K,I)=0.
                   DPLT(K,I)=0.
               ENDIF
10    CONTINUE
C
C
C     ****************************************************************
C     ESTABLISH INITIAL LOAD MATRIX
C     ****************************************************************
      CALL LOADRES (RES,LOADTIME,T,LOAD1,LOAD2,LDIND1,LDIND2,
     1              LOAD,LAODTOT,LIND)
C     ****************************************************************
C     ESTABLISH VOLTAGE MATRIX
C     ****************************************************************
      DO 380 N=1,6
        VOLT(N)=0.
  380 CONTINUE
        VOLT(4)=VF
C     ****************************************************************
C     ESTABLISH INITIAL ANGULAR VELOCITY OF ROTOR, "RPS"
C     ****************************************************************
        RPS=RPM*PI/30.
        FREQ=RPM*NPOLES/120.
C     ****************************************************************
C     ****************************************************************
C                               BEGIN MAIN LOOP
C     ****************************************************************
C     ****************************************************************
      DO 501 L=1,NOLOOPS
      IF(L.LE.100)GO TO 494
C     ****************************************************************
C     CALCULATE PHASE ANGLES
C     ****************************************************************
        A2=A1-TWOPI/3.
        A3=A1-2.*TWOPI/3.
C     ****************************************************************
C     DETERMINE LOAD
C     ****************************************************************
      XL=TWOPI*FREQ
      CALL LOADRES (RES,LOADTIME,T,LOAD1,LOAD2,LDIND1,LDIND2,
     1            LOAD,LOADTOT,LIND)
C     ****************************************************************
C     FIELD VOLTAGE ROUTINE
C     ****************************************************************
      VCEILING=10000000.
      CALL RMS (VRMS,VCEILING,A1,OLDA1,MAX,PREVRMS,VOLT(1),VOLT(4),
     1          RMSV1,REGGAIN)
      IF(L.LE.5) THEN
        VOLT(4)=VF
```

```
         END IF
        OLDA1=A1
        PREVRMS=VOLT(4)
C      ***************************************************************
C      CALCULATE RESULTANT DIRECT AXIS CURRENT IN ROTOR, "ID"
C      ***************************************************************
       IQ=(-2./3.)*(AMP2(1)*SIN(A1)+AMP2(2)*SIN(A2)+AMP2(3)*SIN(A3))
       ID=(AMP2(1)+AMP2(2)+AMP2(3))/3.
       IDS=(2./3.)*(AMP2(1)*COS(A1)+AMP2(2)*COS(A2)+AMP2(3)*COS(A3))
C      ***************************************************************
C      CALCULATE MACHINE PARAMETERS
C      ***************************************************************
C
C      A "1" AFTER VARIABLE DENOTES LOWER CASE SUBSCRIPT, I.E. "LF1"
C        DENOTES "LSUBf", "LF" DENOTES "LSUBF"
C
  388    LF1=10.41*(1.+.1358*ID-.08034*ID**2+.01038*ID**3-
     1       .0004295*ID**4)
         MF1=.243*(1.+.1358*ID-.08034*ID**2+.01038*ID**3-
     1       .0004295*ID**4)
         LD1=(7.44/1000.)*(1.+.0554*ID-.04534*ID**2+.00659*ID**3-
     1       .0002949*ID**4)
         MD1=(6.1/1000.)*(1.+.1334*ID-.07473*ID**2+.009819*ID**3-
     1       .0004106*ID**4)
         MFD=.225*(1.+.1334*ID-.07473*ID**2+.009819*ID**3-
     1       .0004106*ID**4)
C        MFD=.225*(1.+.1573*ID-.08855*ID**2+.01188*ID**3-
C    1       .0002949*ID**4)
         LD=(8.02/1000.)*(1.-.0003*ID-.03166*ID**2+.004299*ID**3-
     1       .0001723*ID**4)
         MD=(3.96/1000.)*(1.-.0003*ID-.03166*ID**2+.004299*ID**3-
     1       .0001723*ID**4)
         LQ1=2.6/1000.
         LQ=5.62/1000.
         MQ1=3.05/1000.
         MQ=1.99/1000.
         LL1=0.
         LL2=0.
         LL3=0.
C      ***************************************************************
C      SET UP ELEMENTS OF "L" MATRIX
C      ***************************************************************
         IND(1,1)=LL1+LD*COS(A1)**2+LQ*SIN(A1)**2
         IND(1,2)=MD*COS(A1)*COS(A2)+MQ*SIN(A1)*SIN(A2)
         IND(1,3)=MD*COS(A1)*COS(A3)+MQ*SIN(A1)*SIN(A3)
         IND(1,4)=MF1*COS(A1)
         IND(1,5)=MD1*COS(A1)
         IND(1,6)=-MQ1*SIN(A1)
         IND(2,1)=IND(1,2)
         IND(2,2)=LL2+LD*COS(A2)**2+LQ*SIN(A2)**2
         IND(2,3)=MD*COS(A2)*COS(A3)+MQ*SIN(A2)*SIN(A3)
         IND(2,4)=MF1*COS(A2)
         IND(2,5)=MD1*COS(A2)
         IND(2,6)=-MQ1*SIN(A2)
         IND(3,1)=IND(1,3)
         IND(3,2)=IND(2,3)
         IND(3,3)=LL3+LD*COS(A3)**2+LQ*SIN(A3)**2
         IND(3,4)=MF1*COS(A3)
```

```
        IND(3,5)=MD1*COS(A3)
        IND(3,6)=-MQ1*SIN(A3)
        IND(4,1)=IND(1,4)
        IND(4,2)=IND(2,4)
        IND(4,3)=IND(3,4)
        IND(4,4)=LF1
        IND(4,5)=MFD
        IND(4,6)=0.
        IND(5,1)=IND(1,5)
        IND(5,2)=IND(2,5)
        IND(5,3)=IND(3,5)
        IND(5,4)=IND(4,5)
        IND(5,5)=LD1
        IND(5,6)=0.
        IND(6,1)=IND(1,6)
        IND(6,2)=IND(2,6)
        IND(6,3)=IND(3,6)
        IND(6,4)=IND(4,6)
        IND(6,5)=IND(5,6)
        IND(6,6)=LQ1
C      **************************************************************
C      CONVERT INDUCTANCE INTO INDUCTIVE REACTANCE
C      INDUCTIVE REACTANCE = 2*PI*FREQ*IND(I,J) or XL*IND(I,J)
C      **************************************************************
        DO 80 I=1,6
        DO 79 J=1,6
          IND(I,J)=XL*IND(I,J)
   79   CONTINUE
   80   CONTINUE
C      **************************************************************
C      INVERT INDUCTANCE MATRIX
C      **************************************************************
        CALL INVIND (IND,INDI)
C      **************************************************************
C      SOLVE THE EQUATION "di/dt=L(-1)(V-Ri)"
C                    OR  "DIDT=INDI*(VOLT-LOAD*AMP)"
C      USING RUNGA-KUTTA 4TH ORDER NUMERICAL INTEGRATION ROUTINE
C      **************************************************************
      DO 332 I=1,6
      AMP1(I)=AMP2(I)
  332 CONTINUE
C
C      INTRODUCE VOLTAGE MATRIX (ALL ZERO EXCEPT FOR VF)
C
      DO 449 I=1,6
      V0(I)=0.
  449 CONTINUE
      V0(4)=VOLT(4)
      CALL INTEGR8 (ZTK1,ZTK2,ZTK3,DELT,AMP1,LOADTOT,V0,INDI,AMP2,AKS,
     1  MTK1,MTK2,MTK3)
      DO 450 I=1,6
      ZTK1(I)=MTK1(I)
      ZTK2(I)=MTK2(I)
      ZTK3(I)=MTK3(I)
  450 CONTINUE
C      **************************************************************
C      AT THIS POINT WE HAVE ALL CURRENTS, BUT NOT THE LOAD
C      VOLTAGES.  WE MUST NOW SOLVE:
```

```
C                               "V=(R + Lp)i"
C                             "V=Ri + L(di/dt)"
C     ****************************************************************
C     VARIABLES:  R = RES(i),   X = L(di/dt)
C     ****************************************************************
      DO 477 I=1,3
        VR(I)=LOAD(I)*AMP2(I)
        VL(I)=LIND(I)*AKS(I)/DELT
  477 CONTINUE
      DO 481 I=1,3
        VOLT(I)=VR(I)+VL(I)
  481 CONTINUE                                  ,
C     ****************************************************************
C     CALCULATE ELECTRICAL TORQUE ON SHAFT
C     ****************************************************************
C     ****************************************************************
C      CALCULATE RMS ETORQUE
C     ****************************************************************
       CALL TORQRMS(VEL,POWR,ETRQRMS)
C     ****************************************************************
C     CALCULATE MECHANICAL TORQUE ON SHAFT
C     ****************************************************************
      MTORQUE=(MTORQUE+(RPM-VEL*60./TWOPI)*GOVGAIN)
C     ****************************************************************
C     CALCULATE ELECTRICAL POWER OUTPUT
C     ****************************************************************
      CALL POWER(A1,LASTA1,CMAX,AMP2,VOLT,PREPWR,POWR,RMSV1,RMSAMP)
C     ****************************************************************
      LASTA1=A1
      PREPWR=POWR
C     ****************************************************************
C     CALCULATE ANGULAR VELOCITY & ACCELERATION OF ROTOR SHAFT
C     ****************************************************************
      CALL DYNAMICS (RPM,TWOPI,MTORQUE,ETRQRMS,JO,DO,DELT,RPS,A0,VEL,
     1   ACCEL,DELTA,DISP,A1,FREQ,NPOLES)
      RPS=VEL
      A0=A1
      GO TO 496
C     ****************************************************************
 494  CALL INITIAL(RES,A1,VOLT,AMP2,MTORQUE,ETRQRMS,FREQ,ACCEL,VEL,
     1     DISP,TWOPI,RPS,DELT,ID,NPOLES,VRMS,RESITORQ)
C     ****************************************************************
C     ****************************************************************
C     STORE VOLTAGES, CURRENTS AND DYNAMICS IN ARRAYS
C     SAMPLING AS NECESSARY
C     ****************************************************************
C     ****************************************************************
  496 IF(NOUT.EQ.0) THEN
                        NSTORE=NSTORE+1
                        VPLT(1,NSTORE)=VOLT(1)
                        VPLT(2,NSTORE)=VOLT(2)
                        VPLT(3,NSTORE)=VOLT(3)
                        VPLT(4,NSTORE)=VOLT(4)
                        APLT(1,NSTORE)=AMP2(1)
                        APLT(2,NSTORE)=AMP2(2)
                        APLT(3,NSTORE)=AMP2(3)
                        APLT(4,NSTORE)=AMP2(4)
                        APLT(5,NSTORE)=AMP2(5)
```

```
                      APLT(6,NSTORE)=AMP2(6)
                          DPLT(1,NSTORE)=MTORQUE
                          DPLT(2,NSTORE)=ETRQRMS
                          DPLT(3,NSTORE)=FREQ
                          DPLT(4,NSTORE)=ACCEL
                          DPLT(5,NSTORE)=POWR
                          DPLT(6,NSTORE)=DISP
                      ENDIF
       NOUT=NOUT+1
       IF(NOUT.GE.NCNT) NOUT=0
C
  495  T=T+DELT
  501 CONTINUE
C     ****************************************************************
C     NOW CALL PLOTTING ROUTINE TO PRESENT MENUS AND DO ACTUAL
C     PLOTTING
C
      CALL HSCPLOT
C     ****************************************************************
      CALL FINITT(0,0)
C     STOP
      END
C     ****************************************************************
      SUBROUTINE INITIAL(RES,A1,VOLT,AMP2,MTORQUE,ETRQRMS,FREQ,
     1  ACCEL,VEL,DISP,TWOPI,RPS,DELT,ID,NPOLES,VRMS,RESITORQ)
C     ****************************************************************
       REAL MTORQUE,VOLT(4),AMP2(6),RES(6,6),ID,NPOLES
       ACCEL=0.
       VEL=RPS
       DISP=DISP+VEL*DELT
       FREQ=(VEL*60./TWOPI)*NPOLES/120.
       DELTA=VEL*DELT
         DISP=DISP+DELTA
         A1=A1+DELTA
        A1=A1-(IFIX(A1/TWOPI)*TWOPI)
        A2=A1-TWOPI/3.
        A3=A1-2.*TWOPI/3.
       DO 40 I=1,6
         AMP2(I)=0.
   40    CONTINUE
       VOLT(1)=VRMS*SQRT(2.)*COS(A1)
       VOLT(2)=VRMS*SQRT(2.)*COS(A2)
       VOLT(3)=VRMS*SQRT(2.)*COS(A3)
       VOLT(4)=20.
       AMP2(4)=VOLT(4)/RES(4,4)
C
       ID=AMP2(4)
       MTORQUE=RESITORQ
C
       ETRQRMS=1.
       RETURN
       END
C     ****************************************************************
      SUBROUTINE INVIND(U,V)
C     ****************************************************************
C                 "L" MATRIX INVERSION ROUTINE
C     READ IN MATRIX TO BE INVERTED
C     CALCULATE ELEMENTS OF REDUCED MATRIX
```

```
C
      REAL U(6,6),V(6,6)
        DO 210 I=1,6
        DO 200 K=1,6
        V(I,K)=U(I,K)
  200   CONTINUE
  210   CONTINUE
        DO 260 K=1,6
C
C     CALCULATE NEW ELEMENTS OF PIVOT ROW
C
        DO 240 J=1,6
        IF(J.EQ.K) GO TO 240
        V(K,J)=V(K,J)/V(K,K)
  240   CONTINUE
C
C     CALCULATE ELEMENT REPLACING PIVOT ELEMENT
C
        V(K,K)=1./V(K,K)
C
C     CALCULATE NEW ELEMENTS NOT IN PIVOT ROW OR PIVOT COLUMN
C
        DO 255 I=1,6
        DO 255 J=1,6
        IF(I.EQ.K) GO TO 255
        IF(J.EQ.K) GO TO 255
        V(I,J)=V(I,J)-V(K,J)*V(I,K)
  255   CONTINUE
C
C     CALCULATE REPLACEMENT ELEMENTS FOR PIVOT COLUMN EXCEPT PIVOT
C       ELEMENT
C
        DO 260 I=1,6
        IF(I.EQ.K) GO TO 260
        V(I,K)=-V(I,K)*V(K,K)
  260   CONTINUE
      RETURN
      END
C     *****************************************************************
      SUBROUTINE RMS (VRMS,VCEILING,A1,OLDA1,MAX,PREVRMS,VOLT1,VOLT4,
     1              RMSVOLT,REGGAIN)
C     *****************************************************************
      REAL OLDA1,MAX
      RMSVOLT=VRMS
      ABSVALUE=ABS(VOLT1)
 1000 IF(ABSVALUE.GT.MAX) THEN
        MAX=ABSVALUE
        END IF
 1500 IF(A1-OLDA1.LT.0.) THEN
      GO TO 2000
      END IF
      GO TO 2500
 2000 RMSVOLT=MAX/SQRT(2.)
      MAX=0.
 2500 VOLT4=(PREVRMS+(VRMS-RMSVOLT)*REGGAIN)
C
C     PREVENT EXCITATION VOLTAGE FROM EXCEEDING CEILING OR DROPPING
C     BELOW ZERO VOLTS
```

```
C
        IF(VOLT4.GT.VCEILING) THEN
        VOLT4=VCEILING
        END IF
        IF(VOLT4.LE.0.) THEN
        VOLT4=0.
        END IF
      RETURN
      END
C     ***************************************************************
C     SUBROUTINE TORQRMS(PREVRMS,NEWANG,OLDANG,VALUE,MAX,PEAK,RMS)
C     ***************************************************************
C     ***************************************************************
      SUBROUTINE TORQRMS(VEL,POWR,ETRQRMS)
C     ***************************************************************
      REAL ETRQRMS,POWR,VEL
      ETRQRMS=POWR/VEL
      RETURN
      END
C     ***************************************************************
      SUBROUTINE IDRMS(NEWANG,OLDANG,PIE,VALUE,MAX,PEAK,ROOTMEAN)
C     ***************************************************************
      REAL ROOTMEAN,NEWANG,OLDANG,PIE,VALUE,MAX,PEAK,ABSVALUE,ZERO
      ZERO=0.
      ABSVALUE=ABS(VALUE)
 1001 IF(ABSVALUE.GT.MAX) THEN
        PEAK=ABSVALUE
        GO TO 1501
        END IF
      PEAK=MAX
 1501 IF(NEWANG-OLDANG.LT.ZERO) THEN
      GO TO 2001
      END IF
      GO TO 2501
 2001 ROOTMEAN=PEAK/SQRT(2.)
 2501 PEAK=0.
      RETURN
      END
C     ***************************************************************
      SUBROUTINE DYNAMICS (RPM,TWOPI,MTORQUE,ETRQRMS,JO,DO,DELT,
     1 RPS,A0,VEL,ACCEL,DELTA,DISP,A1,FREQ,NPOLES)
C     ***************************************************************
      REAL MTORQUE,JO,NPOLES
      ACCEL=(MTORQUE-DO*RPS-ETRQRMS)/JO
      VEL=RPS+ACCEL*DELT
      DELTA=VEL*DELT
      DISP=DISP+DELTA
      A1=A0+DELTA
      A1=A1-(IFIX(A1/TWOPI)*TWOPI)
      FREQ=(VEL*60./TWOPI)*NPOLES/120.
      RETURN
      END
C     ***************************************************************
      SUBROUTINE POWER (A1,LASTA1,CMAX,CUR,EMF,PREPWR,VOLTAMP,RMSV1,
     1                RMSAMP)
C     ***************************************************************
      REAL CUR(6),EMF(6),LASTA1
C
```

```
C
                 ABSVALUE=ABS(CUR(1))
 1000       IF(ABSVALUE.GT.CMAX) THEN
                  CMAX=ABSVALUE
                  END IF
 1500       IF(A1-LASTA1.LT.0.) THEN
                  GO TO 2000
                  END IF
                  VOLTAMP=PREPWR
            GO TO 2600
 2000       RMSAMP=CMAX/SQRT(2.)
                  CMAX=0.
 2500             VOLTAMP=3.*(RMSAMP*RMSV1)
C
C
C     VOLTAMP=CUR(1)*EMF(1)+CUR(2)*EMF(2)+CUR(3)*EMF(3)
 2600 RETURN
      END
C     *****************************************************************
      SUBROUTINE LOADRES (RES,TIME,T,LOAD1,LOAD2,LDIND1,LDIND2,
     1                    LOAD,LOADTOT,LIND)
C     *****************************************************************
      REAL RES(6,6),TIME,T,LOAD1(3),LOAD2(3),LOAD(3),LIND(3),
     1     LDIND1(3),LDIND2(3),LOADTOT(6,6)
      IF(T.GE.TIME) THEN
          LIND(1)=LDIND1(1)
          LIND(2)=LDIND1(2)
          LIND(3)=LDIND1(3)
          LOADTOT(1,1)=LOAD1(1)+RES(1,1)
          LOADTOT(2,2)=LOAD1(2)+RES(2,2)
          LOADTOT(3,3)=LOAD1(3)+RES(3,3)
          LOAD(1)=LOAD1(1)
          LOAD(2)=LOAD1(2)
          LOAD(3)=LOAD1(3)
      ELSE
          LIND(1)=LDIND2(1)
          LIND(2)=LDIND2(2)
          LIND(3)=LDIND2(3)
          LOADTOT(1,1)=LOAD2(1)+RES(1,1)
          LOADTOT(2,2)=LOAD2(2)+RES(2,2)
          LOADTOT(3,3)=LOAD2(3)+RES(3,3)
          LOAD(1)=LOAD2(1)
          LOAD(2)=LOAD2(2)
          LOAD(3)=LOAD2(3)
      END IF
        LOADTOT(4,4)=RES(4,4)
        LOADTOT(5,5)=RES(5,5)
        LOADTOT(6,6)=RES(6,6)
      RETURN
      END
C     *****************************************************************
      SUBROUTINE INTEGR8 (TJ1,TJ2,TJ3,TDELT,TAMP1,TLOAD,TVOLT,
     1  TINDI,TAMP2,TKS,TK1,TK2,TK3)
C     *****************************************************************
        REAL DIDT1(6),DIDT2(6),DIDT3(6),DIDT4(6)
        REAL TAMP1(6),TAMP2(6)
        REAL RAMP1(6),RAMP2(6),RAMP3(6),RAMP4(6)
        REAL VRA1(6),VRA2(6),VRA3(6),VRA4(6)
```

```
      REAL TVOLT(6)
      REAL TINDI(6,6)
      REAL TJ1(6),TJ2(6),TJ3(6),TK1(6),TK2(6),
     1  TK3(6),TK4(6),TKS(6)
      REAL TDELT,TLOAD(6,6)
C     "I" IS USED TO DESIGNATE THE SIX (6) INDEPENDENT EQUATIONS
C     "J" IS INCREMENTED FROM 1-6 FOR EACH INDEPENDENT EQUATION
      DO 387 I=1,6
        RAMP1(I)=0.
        RAMP2(I)=0.
        RAMP3(I)=0.
        RAMP4(I)=0.
      DO 391 J=1,6
        RAMP1(I)=RAMP1(I)+TLOAD(I,J)*TAMP1(J)
        RAMP2(I)=RAMP2(I)+TLOAD(I,J)*(TAMP1(J)+TJ1(J)/2.)
        RAMP3(I)=RAMP3(I)+TLOAD(I,J)*(TAMP1(J)+TJ2(J)/2.)
        RAMP4(I)=RAMP4(I)+TLOAD(I,J)*(TAMP1(J)+TJ3(J))
  391 CONTINUE
  387 CONTINUE
      DO 390 I=1,6
        VRA1(I)=TVOLT(I)-RAMP1(I)
        VRA2(I)=TVOLT(I)-RAMP2(I)
        VRA3(I)=TVOLT(I)-RAMP3(I)
        VRA4(I)=TVOLT(I)-RAMP4(I)
  390 CONTINUE
      DO 319 I=1,6
        DIDT1(I)=0.
        DIDT2(I)=0.
        DIDT3(I)=0.
        DIDT4(I)=0.
      DO 394 J=1,6
        DIDT1(I)=DIDT1(I)+TINDI(I,J)*VRA1(J)
        DIDT2(I)=DIDT2(I)+TINDI(I,J)*VRA2(J)
        DIDT3(I)=DIDT3(I)+TINDI(I,J)*VRA3(J)
        DIDT4(I)=DIDT4(I)+TINDI(I,J)*VRA4(J)
  394 CONTINUE
  319 CONTINUE
      DO 402 I=1,6
        TK1(I)=TDELT*DIDT1(I)
        TK2(I)=TDELT*DIDT2(I)
        TK3(I)=TDELT*DIDT3(I)
        TK4(I)=TDELT*DIDT4(I)
  402 CONTINUE
      DO 393 I=1,6
      TKS(I)=(TK1(I)+2.*TK2(I)+2.*TK3(I)+TK4(I))/6.
      TAMP2(I)=TAMP1(I)+TKS(I)
  393 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE HSCPLOT
      COMMON/PLOTTING/NSTORE,VPLT(4,2000),APLT(6,2000),DPLT(6,2000)
      DIMENSION X(2001),Y(2001)
      CALL INITT(12)
      CALL BINITT
      DO 2 K=1,2000
2     X(K+1)=FLOAT(K)
```

```
        X(1)=FLOAT(NSTORE)
1       Y(1)=X(1)
        CALL NEWPAG
        CALL ANMODE
        WRITE(*,10)
10      FORMAT(/,30X,'MASTER MENU',//
     C  ,/,10X,'1 PLOT OUTPUT VOLTAGES'
     C  ,/,10X,'2 PLOT OUTPUT CURRENTS'
     C ,/,10X,'3 PLOT DYNAMICS'
     C  ,/,10X,'4 QUIT (HALT EXECUTION)'
     C ,//,10X,'** NOTE **   PRESSING "Z" AT OUTPUT SCREEN LEVEL'
     C ,/,10X,'             WILL ACTIVATE ZOOM FEATURE',//)
        WRITE(*,11)
11      FORMAT('    TYPE THE NUMBER OF THE MENU ITEM DESIRED >>',$)
        READ(*,*,ERR=1,END=1) IRPLY
        IF(IRPLY.EQ.4) RETURN
        IF(IRPLY.LT.1.OR.IRPLY.GT.4) GO TO 1
C
C  VOLTAGES
C
        IF(IRPLY.EQ.1) THEN
21                      WRITE(*,20)
20                      FORMAT
     1                  (13X,' SELECT THE PHASE (1-3) DESIRED >>',$)
                        READ(*,*,ERR=21,END=21) JRPLY
                        IF(JRPLY.LT.1.OR.JRPLY.GT.4) THEN
                               CALL TCSBEL
                               CALL TCSBEL
                               CALL TCSBEL
                               GO TO 21
                                                      ELSE
                                 DO 22 K=1,NSTORE
22                               Y(K+1)=VPLT(JRPLY,K)
                                 CALL NEWPAG
                                 CALL BINITT
23                               CALL CHECK(X,Y)
                                 CALL DSPLAY(X,Y)
                                 CALL TINPUT(IOP)
                                 IF(IOP.EQ.90.OR.IOP.EQ.122) THEN
                                                             CALL ZUUM
                                                             CALL NEWPAG
                                                             GO TO 23
                                                             ENDIF
                                 GO TO 1
                                                      ENDIF
                       ENDIF
C
C
C  CURRENTS
C
        IF(IRPLY.EQ.2) THEN
31                      WRITE(*,30)
30                      FORMAT
     1                  (13X,' SELECT THE PHASE (1-3) DESIRED >>',$)
                        READ(*,*,ERR=31,END=31) KRPLY
                        IF(KRPLY.LT.1.OR.KRPLY.GT.6) THEN
                               CALL TCSBEL
                               CALL TCSBEL
```

```
                               CALL TCSBEL
                               GO TO 31
                                                      ELSE
                                 DO 32 K=1,NSTORE
32                               Y(K+1)=APLT(KRPLY,K)
                                 CALL NEWPAG
                                 CALL BINITT
33                               CALL CHECK(X,Y)
                                 CALL DSPLAY(X,Y)
                                 CALL TINPUT(IOP)
                                 IF(IOP.EQ.90.OR.IOP.EQ.122) THEN
                                                             CALL ZUUM
                                                             CALL NEWPAG
                                                             GO TO 33
                                                             ENDIF
                                 GO TO 1
                                                      ENDIF
                       ENDIF
C
C  DYNAMICS
C
        IF(IRPLY.EQ.3) THEN
41                   WRITE(*,40)
40                   FORMAT(10X,'1  -  MECHANICAL TORQUE'
     C                  ,/,10X,'2  -  ELECTRICAL TORQUE'
     C                  ,/,10X,'3  -  FREQUENCY'
     C                  ,/,10X,'4  -  SHAFT ACCELERATION'
     C                  ,/,10X,'5  -  INSTANTANEOUS POWER'
     C                  ,/,10X,'6  -  SHAFT POSITION'
     C                  ,/,16X,' SELECT OUTPUT (1-6) DESIRED >>',$)
                        READ(*,*,ERR=41,END=41) LRPLY
                        IF(LRPLY.LT.1.OR.LRPLY.GT.6) THEN
                               CALL TCSBEL
                               CALL TCSBEL
                               CALL TCSBEL
                               GO TO 41
                                                      ELSE
                                 DO 42 K=1,NSTORE
42                               Y(K+1)=DPLT(LRPLY,K)
                                 CALL NEWPAG
                                 CALL BINITT
43                               CALL CHECK(X,Y)
                                 CALL DSPLAY(X,Y)
                                 CALL TINPUT(IOP)
                                 IF(IOP.EQ.90.OR.IOP.EQ.122) THEN
                                                             CALL ZUUM
                                                             CALL NEWPAG
                                                             GO TO 43
                                                             ENDIF
                                 GO TO 1
                                                      ENDIF
                       ENDIF
C
         END
C
C
        SUBROUTINE ZUUM
C
```

```
      WRITE(*,101)
 101    FORMAT(/,'   SELECT NEW WINDOW CORNER, PRESS SPACE BAR')
      CALL VCURSR (ICHAR,X1,Y1)
      CALL MOVEA(X1,Y1)
      CALL MOVREL(-10,0)
      CALL DRWREL(20,0)
      CALL MOVREL(-10,-10)
      CALL DRWREL(0,20)
      WRITE(*,202)
 202    FORMAT('   SELECT OPPOSITE CORNER, PRESS SPACE BAR')
      CALL VCURSR (ICHAR,X2,Y2)
      XMIN = X1
      IF (X2.LT.X1) XMIN = X2
      YMIN = Y1
      IF (Y2.LT.Y1) YMIN = Y2
      DELX = ABS(X2-X1)
      DELY = ABS(Y2-Y1)
      YAVG = (Y2+Y1)/2.0
      XAVG = (X2+X1)/2.0
      IF (DELX.EQ.0.0) DELX = 0.001
      IF (DELY.EQ.0.0) DELY = 0.001
      YMIN = YAVG - DELY/2.0
      XMIN = XAVG - DELX/2.0
      XMAX = XAVG + DELX/2.0
      YMAX = YAVG + DELY/2.0
      CALL BINITT
      CALL COMSET(IBASEX(11),XMIN)
      CALL COMSET(IBASEY(11),YMIN)
      CALL COMSET(IBASEX(12),XMAX)
      CALL COMSET(IBASEY(12),YMAX)
      RETURN
      END
C
     SUBROUTINE STROUT(IX,IY,KSTR,NCHR)
     DIMENSION KSTR(1),KAS(80)
     CALL KAM2AS(NCHR,KSTR,KAS)
     CALL MOVABS(IX,IY)
     CALL ANSTR(NCHR,KS)
     RETURN
     END
```

# **APPENDIX B**

## DBase III+ SOURCE CODE FOR

## "Front-End"

## (USER INPUT SCREEN)

```
******************************************************************
* THIS PROGRAM SERVES AS THE PARAMETER DATA ENTRY SCREEN FOR THE *
*   SIMULATION THAT SOLVES THE EQUATIONS OF A SYNCHRONOUS        *
*   GENERATOR.                                                   *
*                                                                *
* WRITTEN BY:   H. SCOTT COOMBE                                  *
*        FOR    ENGR 5904                                        *
* INSTRUCTOR:   R.G. MITCHINER                                   *
*       DATE:   SUMMER I THRU SPRING 1992                        *
******************************************************************
******************************************************************
* SETTING UP THE SCREEN I/O ENVIRONMENT                          *
******************************************************************
SET ECHO OFF
SET TALK OFF
SET SAFETY OFF
SET CONFIRM ON
SET STATUS OFF
SET SCOREBOARD OFF
SET KEY -1 TO EXECUTE_SIMULATION
SET KEY -2 TO QUIT_PROGRAM
STORE .F. TO DONE, QUIT, SCREEN_2
PUBLIC DONE, QUIT, SCREEN_1, SCREEN_2
CLEAR
@ 0,0 TO 23,79 DOUBLE
TString = "ALTERNATING CURRENT GENERATOR SIMULATION"
SET COLOR TO W+
@ 1,CENTER(TString) SAY TString
SET COLOR TO
@ 2,0 SAY CHR(204)
@ 2,79 SAY CHR(185)
@ 2,1 TO 2,78 DOUBLE
**************************************************************
*  OPENING THE DATABASE FILE THAT CONTAINS THE INPUT VALUES  *
*  AND ASSIGNING THE DATABASE VALUES INTO THE VARIABLES      *
**************************************************************
SELECT 1
USE DATA
GO TOP
*
TMAX1        = TMAX
DELT1        = DELT
RPM1         = RPM
VF1          = VF
NPOLES1      = NPOLES
LOADTIME1    = LOADTIME
VRMS1        = VRMS
RES11        = RES1
RES21        = RES2
RES31        = RES3
RES41        = RES4
RES51        = RES5
RES61        = RES6
JO1          = JO
DO1          = DO
GOV_GAIN1    = GOV_GAIN
REG_GAIN1    = REG_GAIN
************************************************************
```

```
* VARIABLES CONTAINED ON SCREEN #2                         *
************************************************************
LOAD111      = LOAD11
LOAD121      = LOAD12
LOAD131      = LOAD13
IND_LOAD111 = IND_LOAD11
IND_LOAD121 = IND_LOAD12
IND_LOAD131 = IND_LOAD13
LOAD211      = LOAD21
LOAD221      = LOAD22
LOAD231      = LOAD23
IND_LOAD211 = IND_LOAD21
IND_LOAD221 = IND_LOAD22
IND_LOAD231 = IND_LOAD23
*
DO SCREEN_1
*
@ 20,0 SAY CHR(204)
@ 20,79 SAY CHR(185)
@ 20,1 TO 20,78 DOUBLE
SET COLOR TO W+
@ 22,3 SAY "["+chr(17)+chr(217)+"]"
@ 22,38 SAY "[F2]"
@ 22,66 SAY "[F3]"
SET COLOR TO
@ 22,8 SAY "- Move to next input value"
@ 22,43 SAY "- Execute Simulation"
@ 22,71 SAY "- Quit"
BEGIN SEQUENCE
     DO WHILE .NOT. DONE
          IF SCREEN_1 = .T.
           BEGIN SEQUENCE
             @ 3,26 GET TMAX1 PICTURE "99.9999"
             READ
             @ 3,26 SAY TMAX1
             @ 4,26 GET DELT1 PICTURE "99.9999"
             READ
             @ 4,26 SAY DELT1
             @ 5,26 GET LOADTIME1 PICTURE "99.9999"
             READ
             @ 5,26 SAY LOADTIME1
             @ 3,66 GET VRMS1 PICTURE "9999.9"
             READ
             @ 3,66 SAY VRMS1
             @ 4,66 GET REG_GAIN1 PICTURE "9999.9"
             READ
             @ 4,66 SAY REG_GAIN1
             @ 5,66 GET VF1 PICTURE "9999.9"
             READ
             @ 5,66 SAY VF1
             @ 8,52 GET JO1 PICTURE "9999.9"
             READ
             @ 8,52 SAY JO1
             @ 9,52 GET DO1 PICTURE "9999.9"
             READ
             @ 9,52 SAY DO1
             @ 10,52 GET GOV_GAIN1 PICTURE "9999.9"
             READ
```

```
   @ 10,52 SAY GOV_GAIN1
   @ 11,54 GET NPOLES1 PICTURE "99"
   READ
   @ 11,54 SAY NPOLES1
   @ 12,52 GET RPM1 PICTURE "9999.9"
   READ
   @ 12,52 SAY RPM1
   @ 14,50 GET RES11 PICTURE "999999.9999"
   READ
   @ 14,50 SAY RES11
   @ 15,50 GET RES21 PICTURE "999999.9999"
   READ
   @ 15,50 SAY RES21
   @ 16,50 GET RES31 PICTURE "999999.9999"
   READ
   @ 16,50 SAY RES31
   @ 17,50 GET RES41 PICTURE "999999.9999"
   READ
   @ 17,50 SAY RES41
   @ 18,50 GET RES51 PICTURE "999999.9999"
   READ
   @ 18,50 SAY RES51
   @ 19,50 GET RES61 PICTURE "999999.9999"
   READ
   @ 19,50 SAY RES61
 END
ELSE
 BEGIN SEQUENCE
   @ 4,52 GET LOAD111 PICTURE "999999.9999"
   READ
   @ 4,52 SAY LOAD111
   @ 5,52 GET LOAD121 PICTURE "999999.9999"
   READ
   @ 5,52 SAY LOAD121
   @ 6,52 GET LOAD131 PICTURE "999999.9999"
   READ
   @ 6,52 SAY LOAD131
   @ 7,52 GET IND_LOAD111 PICTURE "999999.9999"
   READ
   @ 7,52 SAY IND_LOAD111
   @ 8,52 GET IND_LOAD121 PICTURE "999999.9999"
   READ
   @ 8,52 SAY IND_LOAD121
   @ 9,52 GET IND_LOAD131 PICTURE "999999.9999"
   READ
   @ 9,52 SAY IND_LOAD131
   @ 12,52 GET LOAD211 PICTURE "999999.9999"
   READ
   @ 12,52 SAY LOAD211
   @ 13,52 GET LOAD221 PICTURE "999999.9999"
   READ
   @ 13,52 SAY LOAD221
   @ 14,52 GET LOAD231 PICTURE "999999.9999"
   READ
   @ 14,52 SAY LOAD231
   @ 15,52 GET IND_LOAD211 PICTURE "999999.9999"
   READ
   @ 15,52 SAY IND_LOAD211
```

```
             @ 16,52 GET IND_LOAD221 PICTURE "999999.9999"
             READ
             @ 16,52 SAY IND_LOAD221
             @ 17,52 GET IND_LOAD231 PICTURE "999999.9999"
             READ
             @ 17,52 SAY IND_LOAD231
           END
          ENDIF
     ENDDO
END
CLOSE ALL
IF QUIT = .F.
     ERRORLEVEL(1)
ELSE
     ERRORLEVEL(0)
ENDIF
*
********************************************************************
*                  ---> FUNCTION Center <---                       *
*                                                                  *
*  Purpose:  To return to Starting column position when centering  *
*            a passed-in text string.                              *
********************************************************************
FUNCTION Center
*
Parameters in_string, in_length
*
IF TYPE("in_length") = "U"
     in_length = 80
ENDIF
RETURN (in_length/2 - LEN(in_string)/2)
*
********************************************************************
*             ---> PROCEDURE Screen_1 <---                         *
*                                                                  *
*  Purpose:  To display Screen 1 and store the input values for    *
*            the variables displayed on Screen 1.                  *
********************************************************************
PROCEDURE SCREEN_1
*
SET KEY 18 TO
SET KEY 3  TO SCREEN_2
SCREEN_1 = .T.
*
@ 4,2 CLEAR TO 19,78
@ 21,2 CLEAR TO 21,78
*
TString = "1 OF 2"
@ 1,72 SAY TString
*
@ 3,2 SAY "TMAX (Simulation Time)= "+STR(TMAX1)+" sec"
@ 4,2 SAY "DELT (Time Increment) = "+STR(DELT1)+" sec"
@ 5,2 SAY "LOADTIME (Time Appl)  = "+STR(LOADTIME1)+" sec"
@ 3,39 SAY  "VRMS (RMS Output)        = "+STR(VRMS1)+" volts"
@ 4,39 SAY  "REG GAIN (Regulator Gain)= "+STR(REG_GAIN1)
@ 5,39 SAY  "VF (Init Field Volt)     = "+STR(VF1)+" volts"
@ 6,0 SAY CHR(199)
@ 6,79 SAY CHR(182)
```

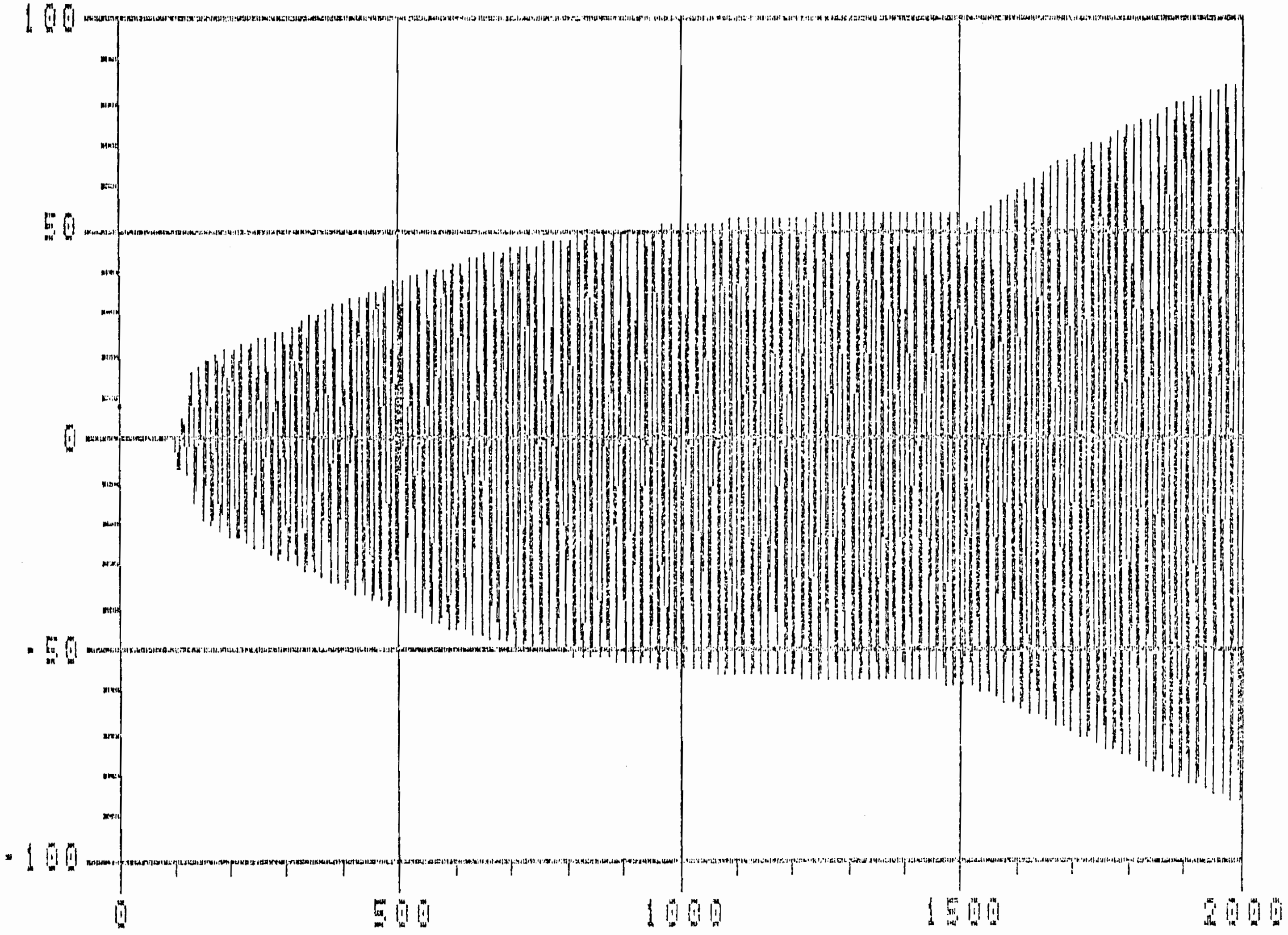
```
@ 6,1 TO 6,78
SET COLOR TO W+
@ 7,2 SAY "Mechanical Parameters:"
SET COLOR TO
@ 8,2 SAY "JO (Flywheel & Rotor Polar Moment of Inertia) =
"+STR(JO1)+" kg-m**2"
@ 9,2 SAY "DO (Frictional Damping Coefficient)          =
"+STR(DO1)+" N-m-s"
@ 10,2 SAY "GOV GAIN (Governor Gain)                     =
"+STR(GOV_GAIN1)
@ 11,2 SAY "NPOLES (Number of Rotor Poles)               =
"+STR(NPOLES1)
@ 12,2 SAY "RPM (Rated Shaft Speed)                      =
"+STR(RPM1)+" rev/min"
SET COLOR TO W+
@ 13,2 SAY "Generator Characteristics:"
SET COLOR TO
@ 14,2 SAY "RES(1,1) (Stator Winding Resistance-Phase 1)  =
"+STR(RES11)+" ohms"
@ 15,2 SAY "RES(2,2) (Stator Winding Resistance-Phase 2)  =
"+STR(RES21)+" ohms"
@ 16,2 SAY "RES(3,3) (Stator Winding Resistance-Phase 3)  =
"+STR(RES31)+" ohms"
@ 17,2 SAY "RES(4,4) (Field Winding Resistance)           =
"+STR(RES41)+" ohms"
@ 18,2 SAY "RES(5,5) (Direct Axis Damper Resistance)      =
"+STR(RES51)+" ohms"
@ 19,2 SAY "RES(6,6) (Quadrature Axis Damper Resistance)  =
"+STR(RES61)+" ohms"
*
SET COLOR TO W+
@ 21,30 SAY "[PgDn]"
SET COLOR TO
@ 21,37 SAY '- Next Screen'
*
IF SCREEN_2 = .T.
   SCREEN_2 = .F.
   BREAK
ENDIF
RETURN
*******************************************************************
*              ---> PROCEDURE Screen_2 <---                       *
*                                                                 *
*  Purpose:  To display Screen 2 and store the input values for   *
*            the variables displayed on Screen 2.                 *
*******************************************************************
PROCEDURE SCREEN_2
*
SET KEY 3 TO
SET KEY 18 TO SCREEN_1
*
SCREEN_1 = .F.
SCREEN_2 = .T.
*
@ 3,1 CLEAR TO 19,78
@ 7,0 SAY CHR(186)
@ 7,79 SAY CHR(186)
@ 21,1 CLEAR TO 21,78
```

```
*
TString = "2 OF 2"
@ 1,72 SAY TString
*
SET COLOR TO W+
@ 3,2 SAY "Initial Load:"
SET COLOR TO
@ 4,5 SAY "RES LOAD1(1) (Initial Resistive Load-Phase 1)
="+STR(LOAD111)+" ohms"
@ 5,5 SAY "RES LOAD1(2) (Initial Resistive Load-Phase 2)
="+STR(LOAD121)+" ohms"
@ 6,5 SAY "RES LOAD1(3) (Initial Resistive Load-Phase 3)
="+STR(LOAD131)+" ohms"
@ 7,5 SAY "IND LOAD1(1) (Initial Inductive Load-Phase 1)
="+STR(IND_LOAD111)+ " henries"
@ 8,5 SAY "IND LOAD1(2) (Initial Inductive Load-Phase 2)
="+STR(IND_LOAD121)+ " henries"
@ 9,5 SAY "IND LOAD1(3) (Initial Inductive Load-Phase 3)
="+STR(IND_LOAD131)+ " henries"
*
SET COLOR TO W+
@ 11,2 SAY "Final Load:"
SET COLOR TO
@ 12,5 SAY "RES LOAD2(1) (Final Resistive Load-Phase 1)
="+STR(LOAD211)+" ohms"
@ 13,5 SAY "RES LOAD2(2) (Final Resistive Load-Phase 2)
="+STR(LOAD221)+" ohms"
@ 14,5 SAY "RES LOAD2(3) (Final Resistive Load-Phase 3)
="+STR(LOAD231)+" ohms"
@ 15,5 SAY "IND LOAD2(1) (Final Inductive Load-Phase 1)
="+STR(IND_LOAD211)+ " henries"
@ 16,5 SAY "IND LOAD2(2) (Final Inductive Load-Phase 2)
="+STR(IND_LOAD221)+ " henries"
@ 17,5 SAY "IND LOAD2(3) (Final Inductive Load-Phase 3)
="+STR(IND_LOAD231)+ " henries"
*
SET COLOR TO W+
@ 21,30 SAY "[PgUp]"
SET COLOR TO
@ 21,37 SAY '- Previous Screen'
BREAK
RETURN
*******************************************************************
*             ---> PROCEDURE Execute_Simulation <---              *
*                                                                 *
*  Purpose:  To call the Generate Simulation Program Executable   *
*******************************************************************
PROCEDURE EXECUTE_SIMULATION
*
SELECT 1
GO TOP
REPLACE TMAX WITH TMAX1
REPLACE DELT WITH DELT1
REPLACE RPM WITH RPM1
REPLACE VF WITH VF1
REPLACE NPOLES WITH NPOLES1
REPLACE LOADTIME WITH LOADTIME1
REPLACE VRMS WITH VRMS1
```

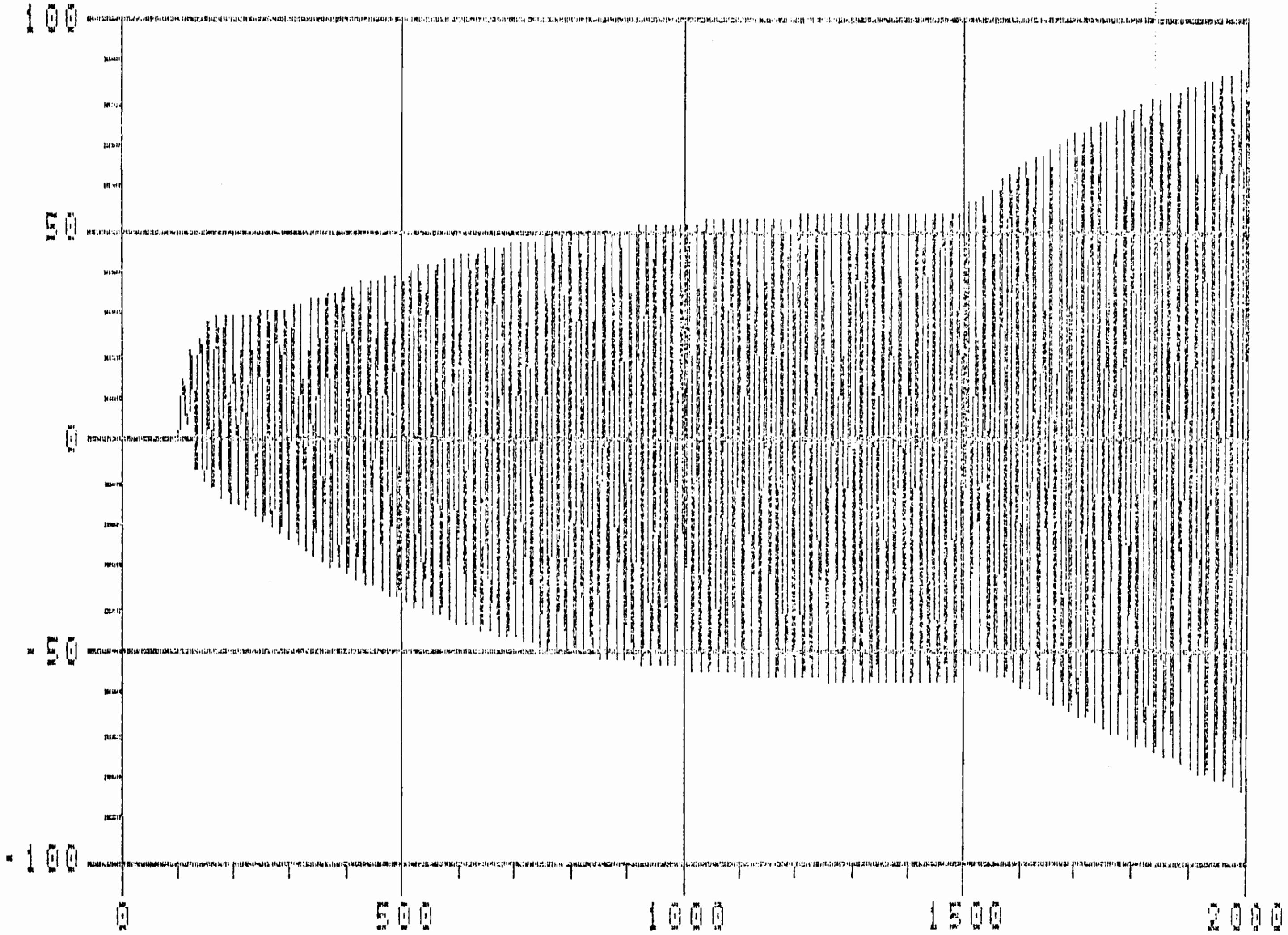
```
REPLACE RES1 WITH RES11
REPLACE RES2 WITH RES21
REPLACE RES3 WITH RES31
REPLACE RES4 WITH RES41
REPLACE RES5 WITH RES51
REPLACE RES6 WITH RES61
REPLACE JO WITH JO1
REPLACE DO WITH DO1
REPLACE GOV_GAIN WITH GOV_GAIN1
REPLACE REG_GAIN WITH REG_GAIN1
*
*  Saving the Screen 2 variables
*
REPLACE LOAD11 WITH LOAD111
REPLACE LOAD12 WITH LOAD121
REPLACE LOAD13 WITH LOAD131
REPLACE IND_LOAD11 WITH IND_LOAD111
REPLACE IND_LOAD12 WITH IND_LOAD121
REPLACE IND_LOAD13 WITH IND_LOAD131
REPLACE LOAD21 WITH LOAD211
REPLACE LOAD22 WITH LOAD221
REPLACE LOAD23 WITH LOAD231
REPLACE IND_LOAD21 WITH IND_LOAD211
REPLACE IND_LOAD22 WITH IND_LOAD221
REPLACE IND_LOAD23 WITH IND_LOAD231
*
GO TOP
COPY TO DATA.TXT ALL DELIMITED
*
*  Updating the public variables that "get us out" of this program
*
DONE = .T.
QUIT = .F.
*
*  Breaking out of the BEGIN SEQUENCE-END structure to execute the
simulation
*
BREAK
RETURN
*******************************************************************
*              ---> PROCEDURE QUIT_PROGRAM <---                   *
*                                                                 *
*  Purpose:  To QUIT the Generator Simulation Program             *
*******************************************************************
PROCEDURE QUIT_PROGRAM
*
*  Updating the public variables that "get us out" of this program
DONE = .T.
QUIT = .T.
*
*  Breaking out of the BEGIN SEQUENCE-END structure to execute the
simulation
*
BREAK
RETURN
```

# **APPENDIX C**

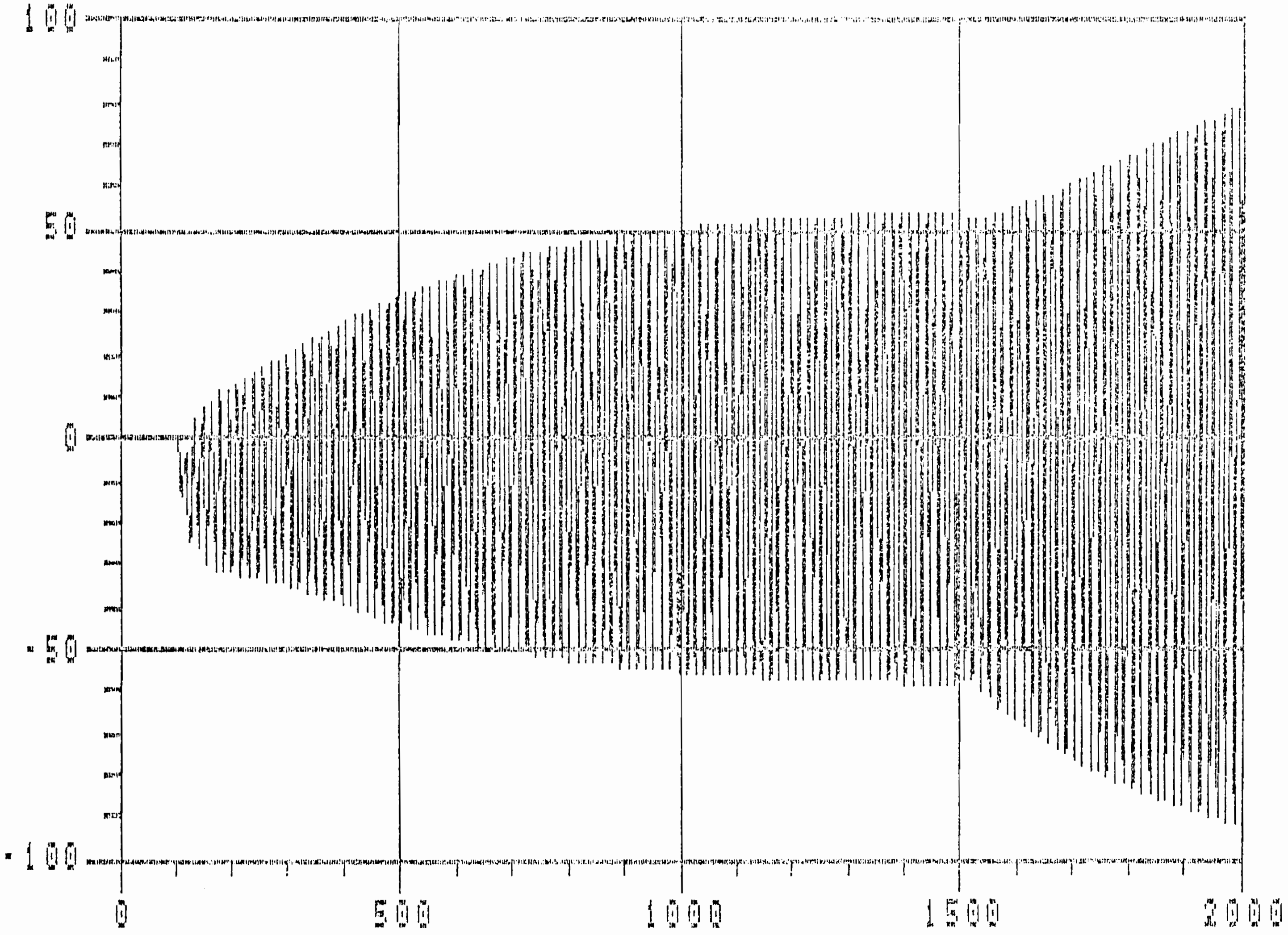
## SAMPLE OUTPUT DATA



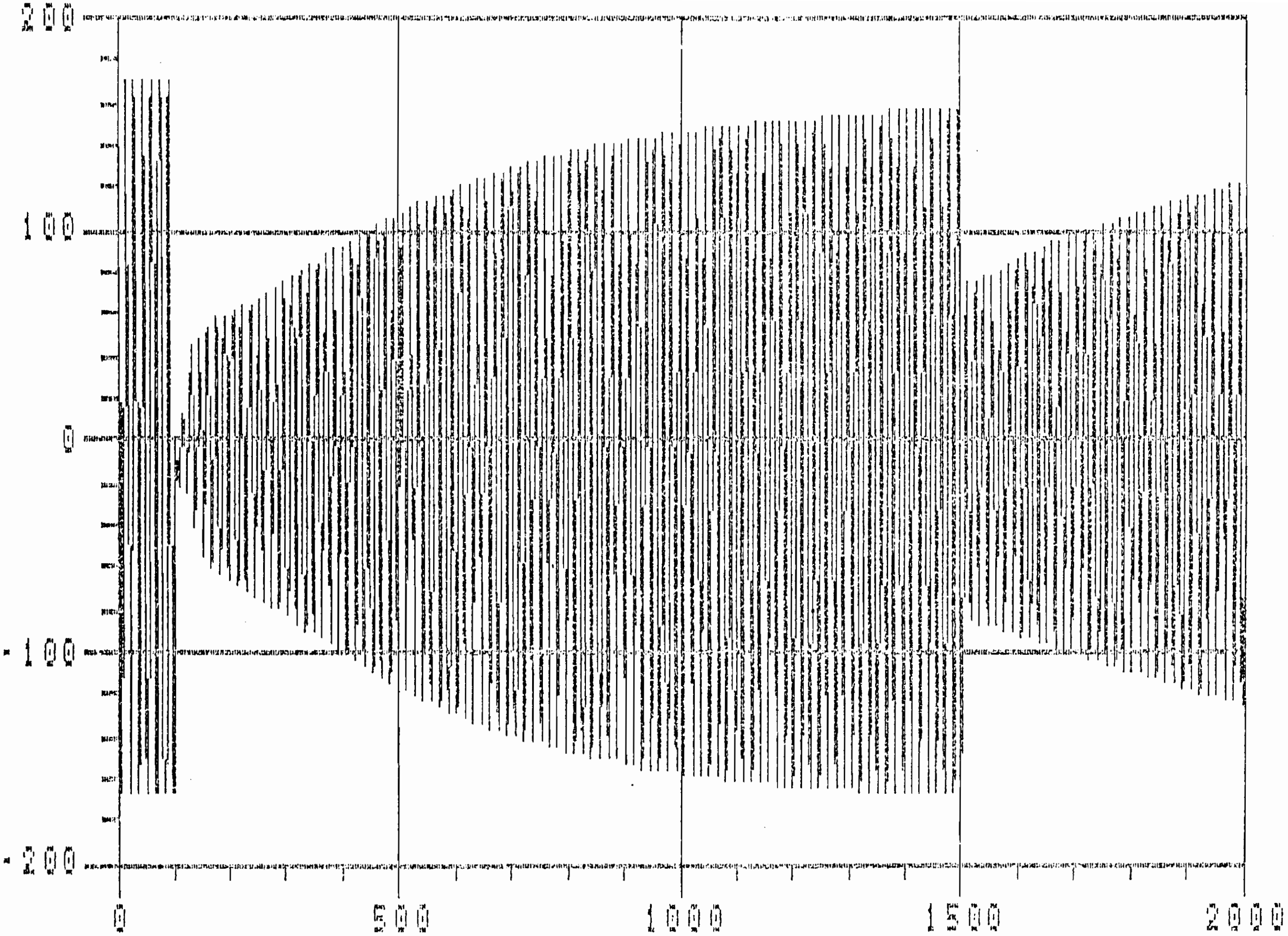
Phase 1 Current (amps) vs. Time



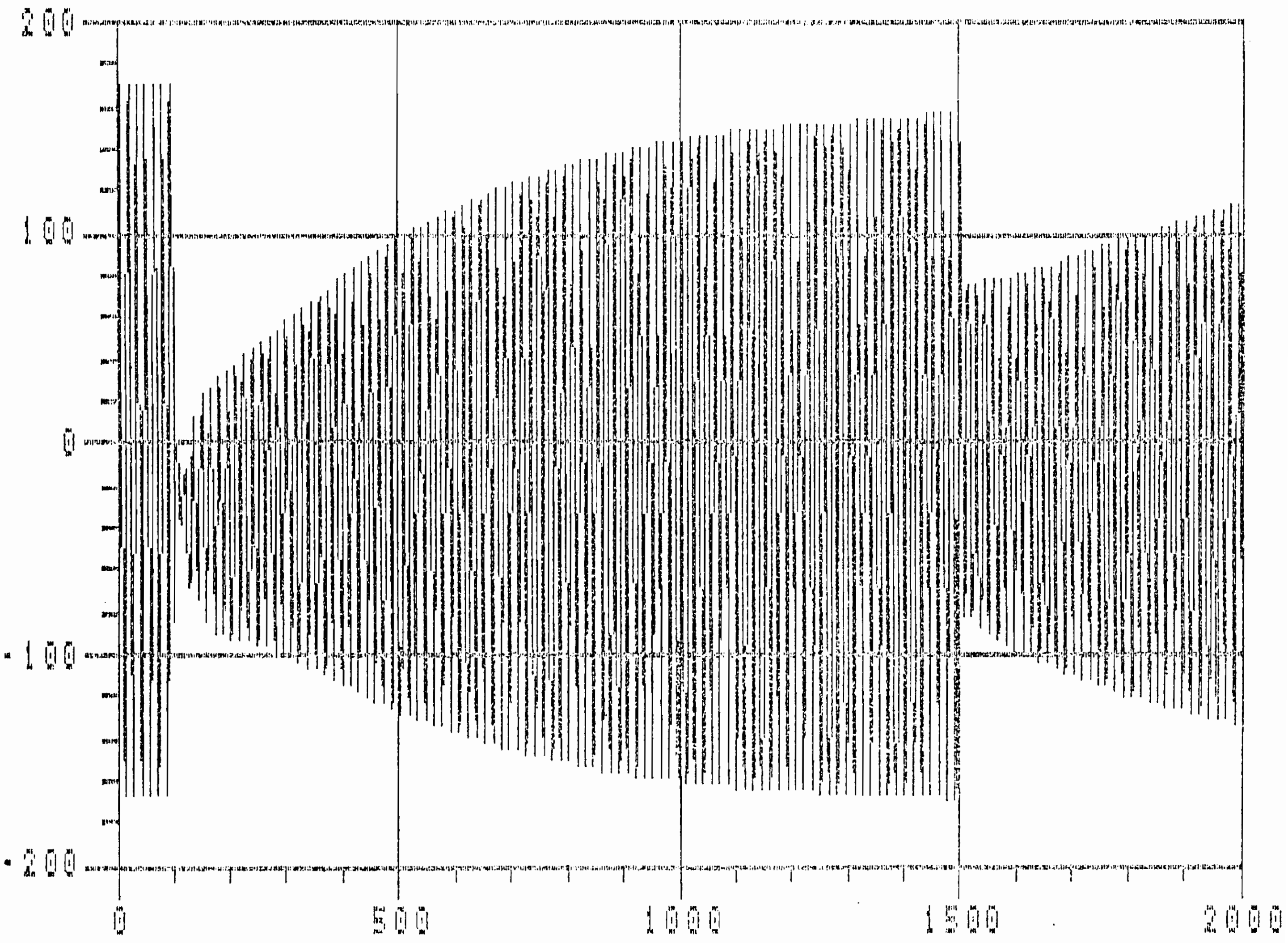
Phase 2 Current (amps) vs. Time



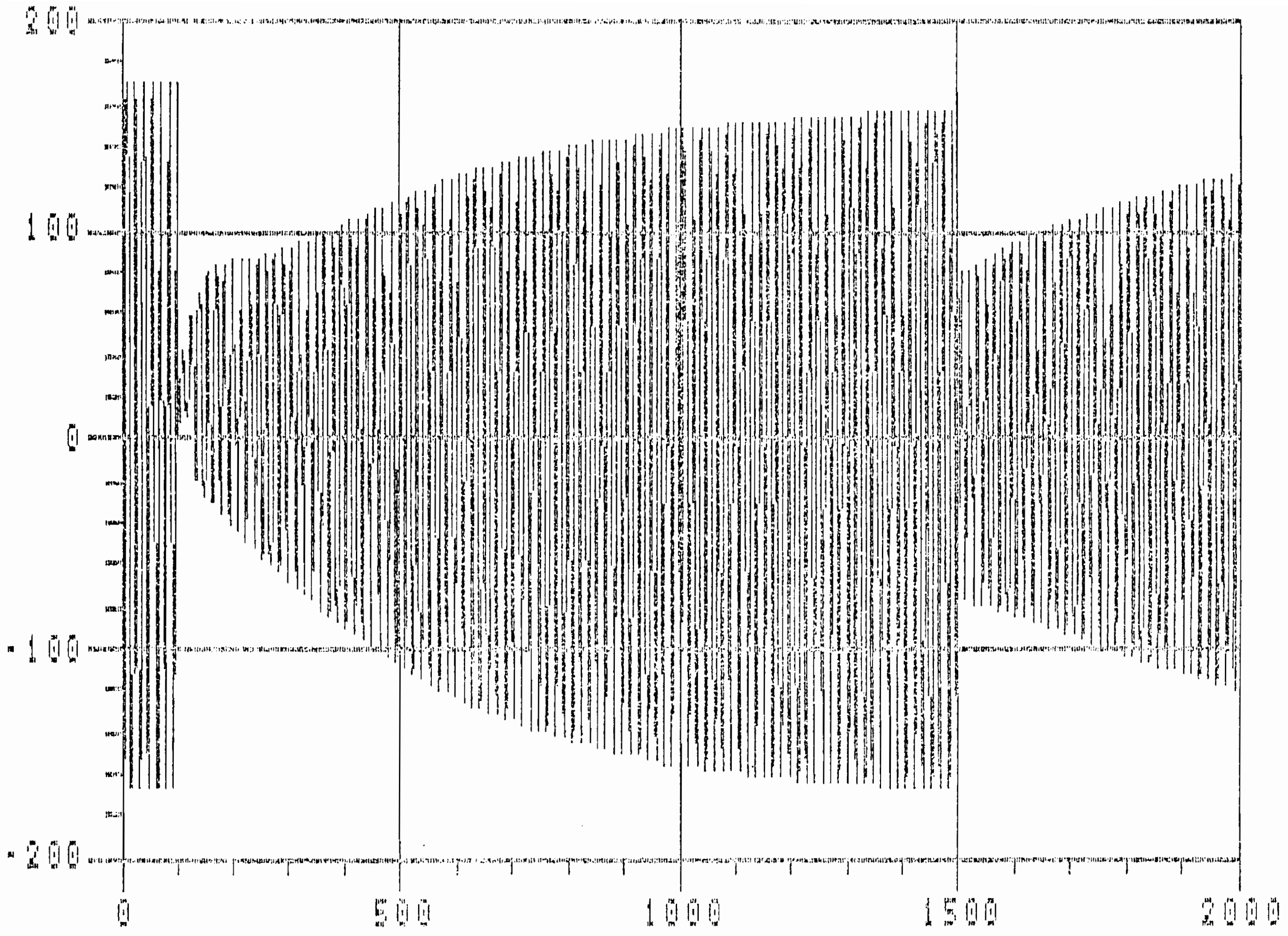
Phase 3 Current (amps) vs. Time



Phase 1 Voltage (volts) vs. Time

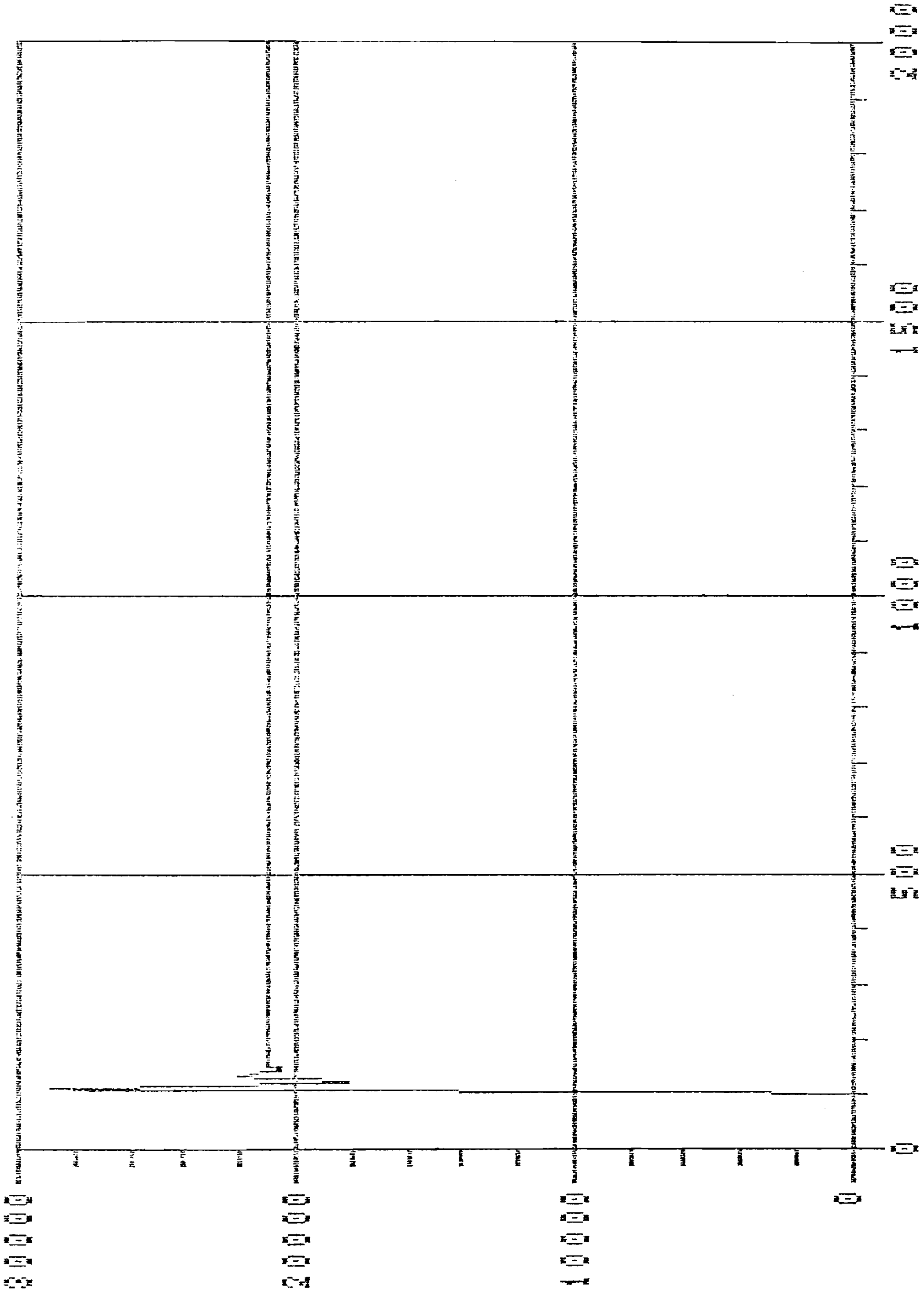


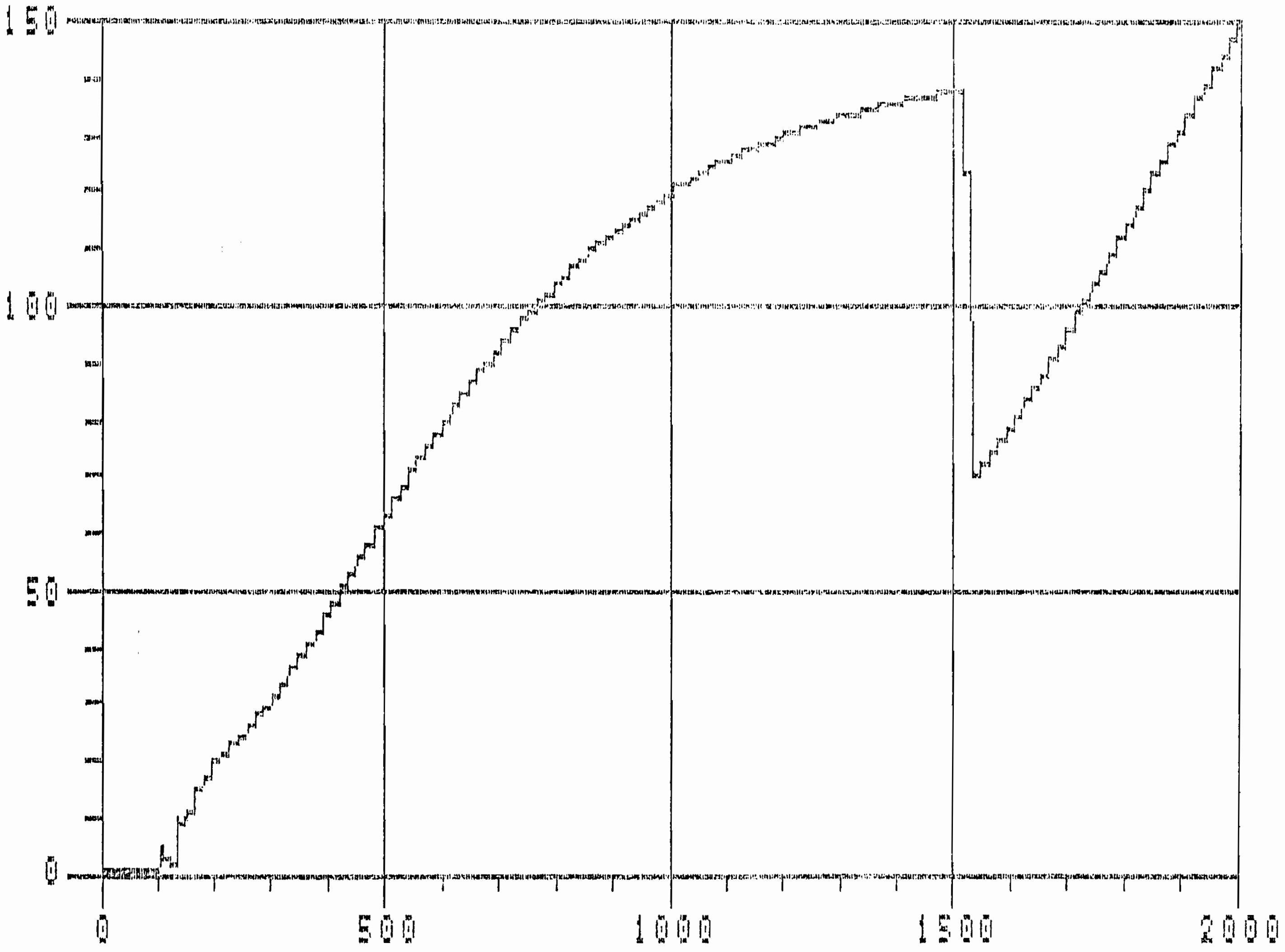
Phase 2 Voltage (volts) vs. Time



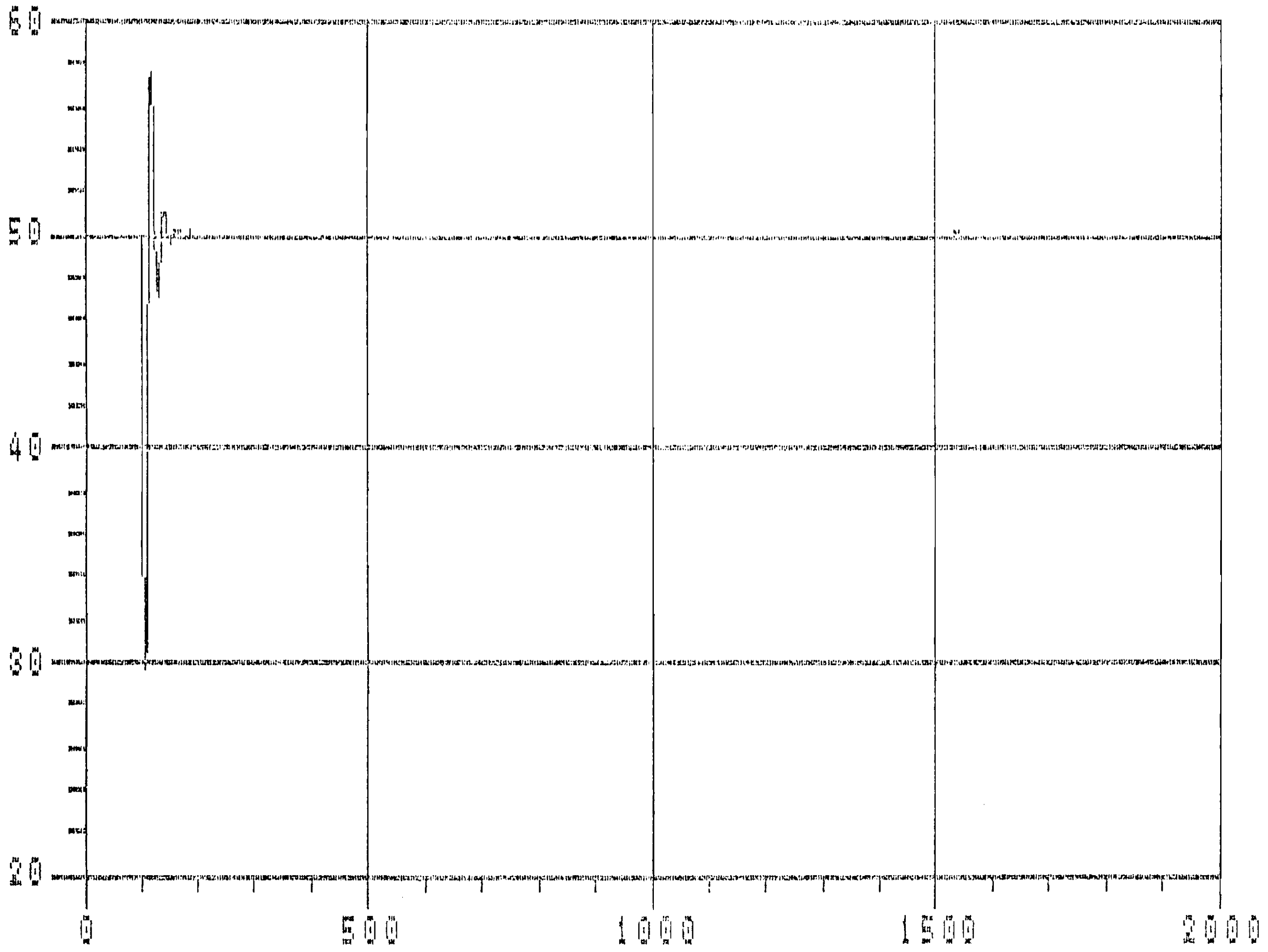
Phase 3 Voltage (volts) vs. Time

**Mechanical Torque (N-m) vs. Time**

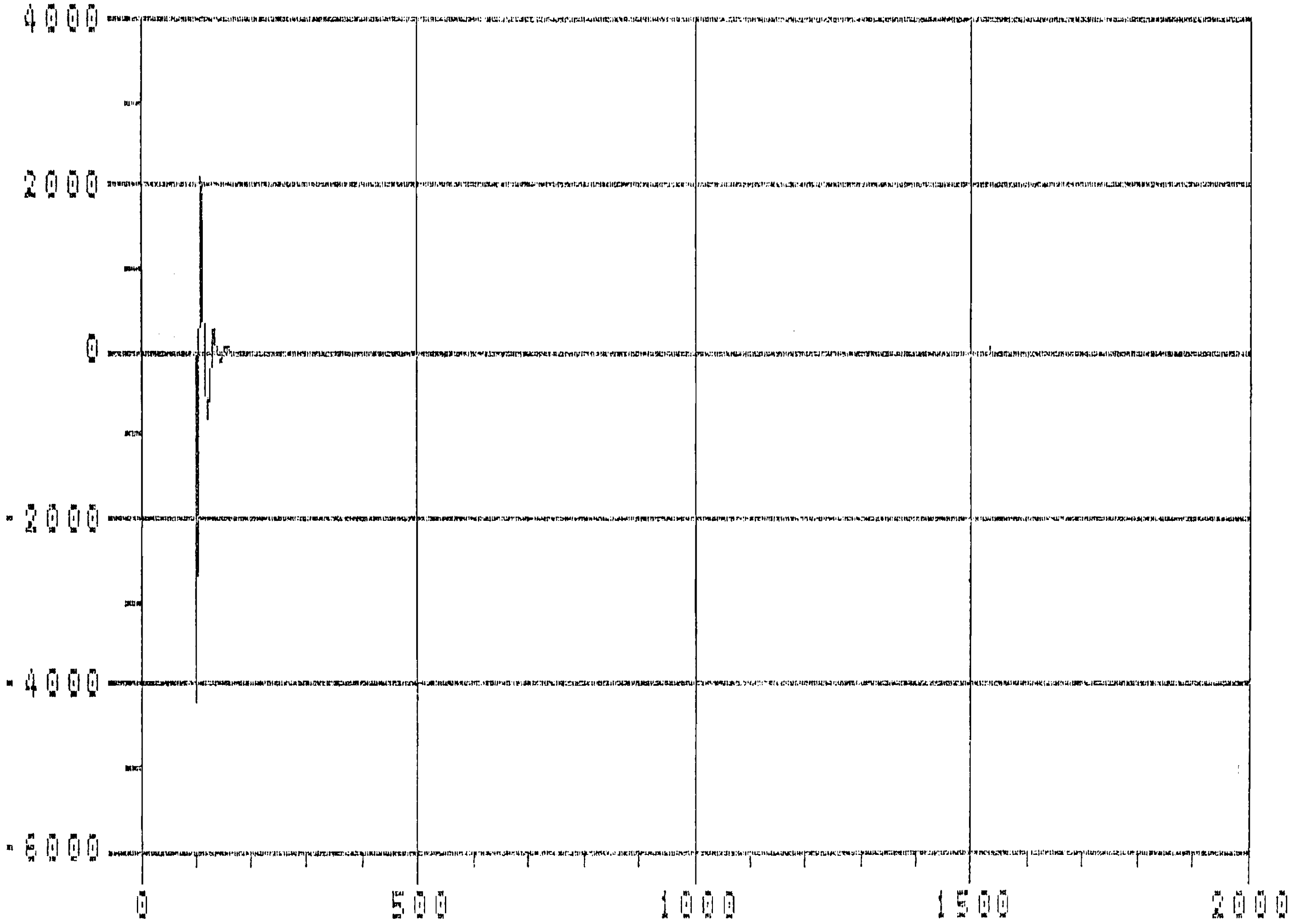




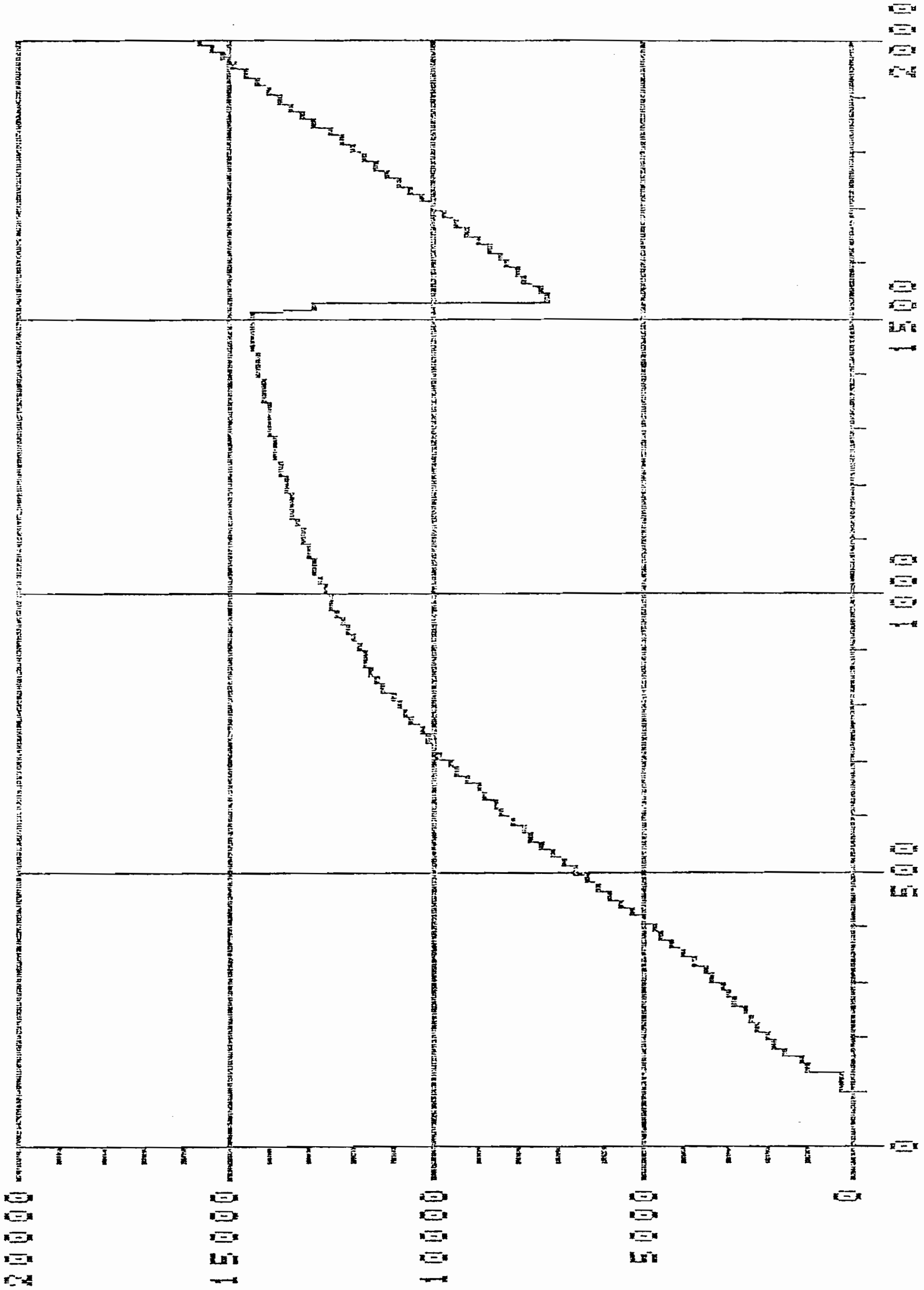
Electrical Torque (N-m) vs. Time



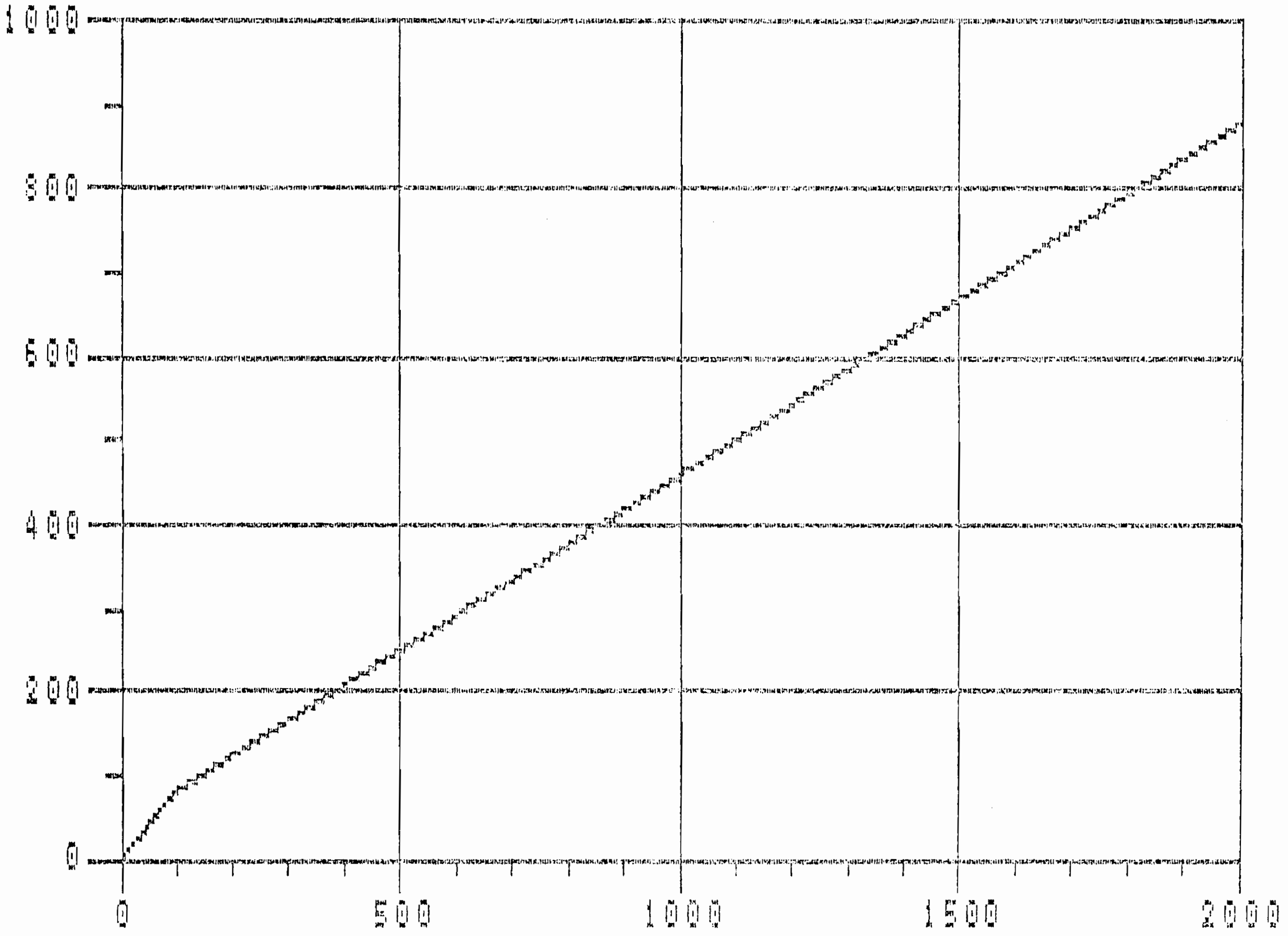
Frequency (Hz) vs. Time



Shaft Acceleration ($rad/s^2$) vs. Time



**Electrical Power Output (Watts) vs. Time**

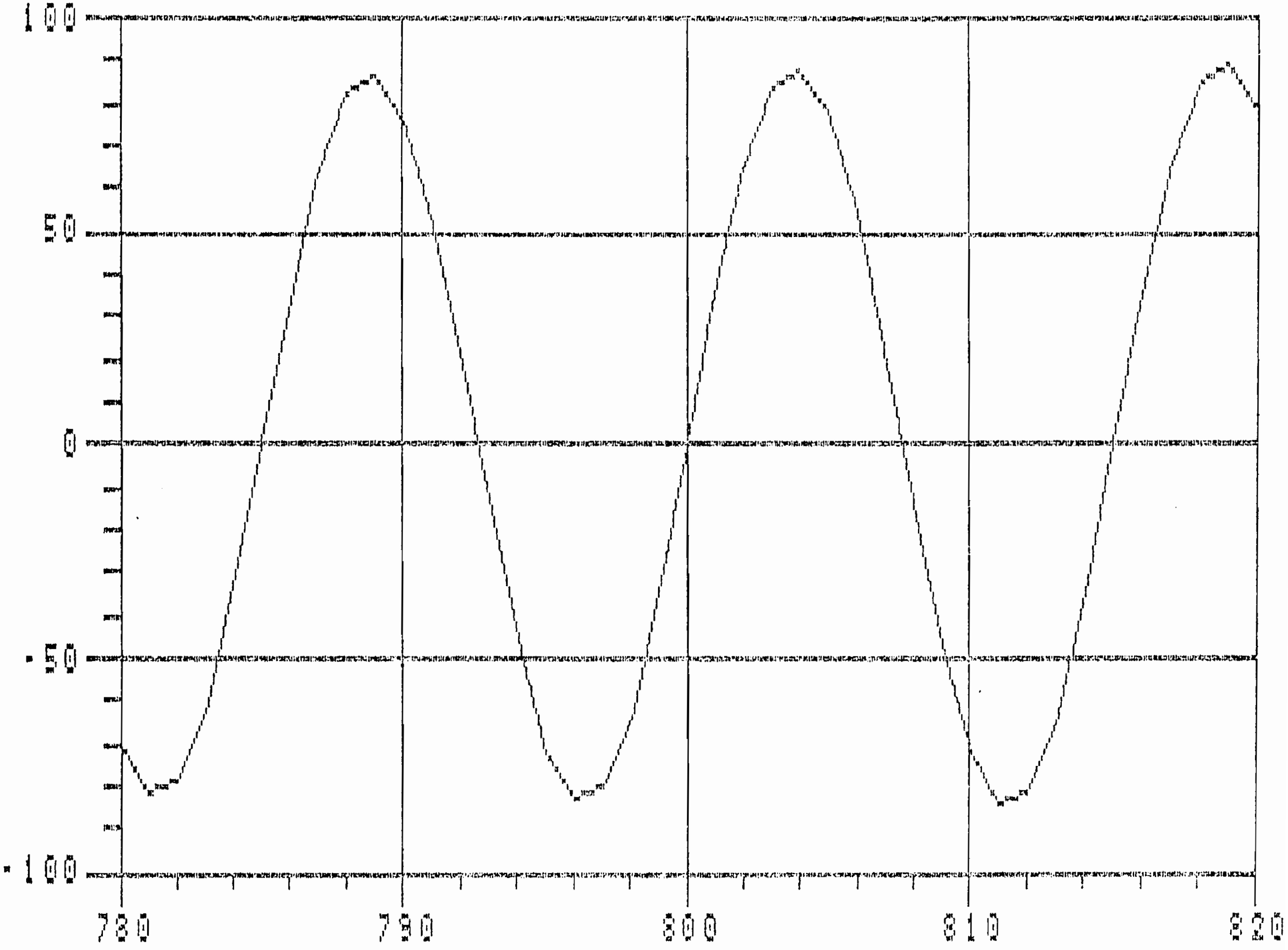


Shaft Position (rad) vs. Time

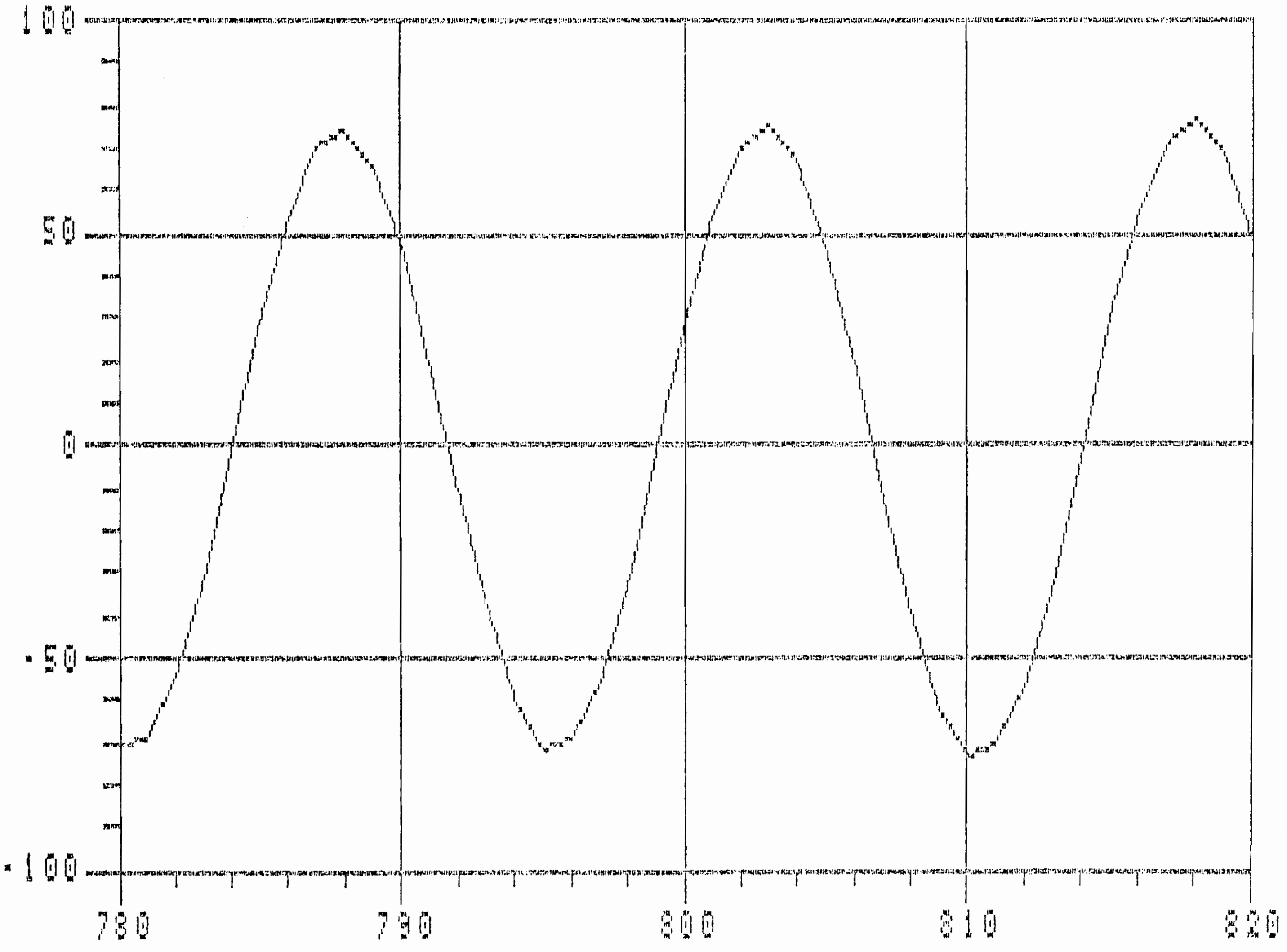
# APPENDIX D

## INDUCTIVE LOADING

## PLOTS OF PHASE 1 VOLTAGE & CURRENT



Phase 1 Current (amps) vs. Time



Phase 1 Voltage (volts) vs. Time

# VITA

H. SCOTT COOMBE

DOB: 21 August 1962

I graduated from Virginia Polytechnic Institute & State University in 1984 with a B.S. in Mechanical Engineering. Since then, I have been employed by the US Army Belvoir Research, Development, & Engineering Center, Fort Belvoir, VA. In nearly 8 years in the Power Generation Division at Belvoir, I have contributed to and/or managed various product developments such as noise enclosures for electric power generators, and the development of an experimental generator whose capabilities included noise suppression, infrared suppression, and nuclear hardening. I am currently assigned lead project engineering responsibilities for the development of a new family of mobile electric power generators for the Army and DOD. I am also a licensed Professional Engineer in the state of Virginia.