1986-06

Marine gas turbine modeling for modern control design

Herda, Vincent J.

https://hdl.handle.net/10945/21966

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MARINE GAS TURBINE MODELING
FOR
MODERN CONTROL DESIGN

by

Vincent J. Herda

June 1986

Thesis Advisor: David Smith

REPORT DOCUMENTATION PAGE

REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

SECURITY CLASSIFICATION AUTHORITY

DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT

PERFORMING ORGANIZATION REPORT NUMBER(S)

5 MONITORING ORGANIZATION REPORT NUMBER(S)

NAME OF PERFORMING ORGANIZATION
aval Postgraduate School

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION

ADDRESS *(City, State, and ZIP Code)*
Monterey, California 93943-5000

7b. ADDRESS *(City, State, and ZIP Code)*

NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

ADDRESS *(City, State, and ZIP Code)*

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

1 TITLE *(Include Security Classification)*
Marine Gas Turbine Modeling for Modern Control Design

2 PERSONAL AUTHOR(S) Vincent J. Herda

3a TYPE OF REPORT
asters Thesis

13b TIME COVERED
FROM ________ TO Jun86

14 DATE OF REPORT *(Year, Month, Day)*
1986 June 19

15 PAGE COUNT
134

6 SUPPLEMENTARY NOTATION

7 COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Marine gas turbine, Modeling, Modern Control

9 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The search for improved performance of U.S. Navy ships has led to more complex propulsion systems consisting of multiple,intracting inputs. Classical control theory does not effectively exploit these interactions. Modern Control Theory provides a systematic method of dealing with multiple intracting inputs to achieve improved system performance. One of the most highly developed modern control techniques is the linear quadratic regulator (LQR) method. Essential to the application of this method is the formulation of a state space description of the plant. In this paper a nonlinear dynamic propulsion system model is developed from experimental data and used to formulate a state space model.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL

22b TELEPHONE *(Include Area Code)*

22c OFFICE SYMBOL

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Marine Gas Turbine Modeling for Modern Control Design

by

Vincent J. Herda
Lieutenant, United States Navy
B.S., U.S. Naval Academy, 1980

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

AND

MECHANICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
June 1986

ABSTRACT

The search for improved performance of U.S. Navy ships has led to more complex propulsion systems consisting of multiple, interacting inputs. Classical control theory does not effectively exploit these interactions. Modern control theory provides a systematic method of dealing with multiple interacting inputs to achieve improved system performance. One of the the most highly developed modern control techniques is the linear quadratic regulator (LQR) method. Essential to the application of this method is the formulation of a state space description of the plant. In this paper a nonlinear dynamic propulsion system model is developed from experimental data and used to formulate a state space model.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SYMBOLS AND ABBREVIATIONS

E = Fuel Energy Realized at HP Turbine
JD = Dynamometer Inertia
JG = Gas Generator Inertia
Ma = Air Mass Flowrate
Maf = Combined Fuel and Air Mass Flowrate
Mf = Fuel Mass Flowrate
NG = Gas Generator Speed
NS = Power Turbine/Dynamometer Speed
P2 = Compressor Discharge Pressure
P4 = High Pressure Turbine Discharge Pressure
QC = Compressor Torque
QD = Dynamometer Torque
QF = Free Power Turbine Torque
QH = High Pressure Turbine Torque
t = Fuel Energy Lag Time Constant
T2 = Compressor Discharge Temperature
T4 = High Pressure Turbine Discharge Temperature
V = Volume Flowrate
Ww = Dynamometer Water Weight

I. INTRODUCTION

The search for improved performance of U.S. Navy ships has led to increasingly complex marine propulsion systems. Controllable inputs to this system now include fuel flow rate, engine inlet guide vane and stator vane position, bleed air selection, and propeller pitch angle. Current control strategies applied to these propulsion plants, and classical control techniques in general, do not take into account the interaction between these inputs. In contrast, modern control techniques (MCT) provide a systematic method to achieve improved system performance when dealing with multiple, interacting inputs. Specifically, modern control theory methods provide the following benefits not found in classical control methods applied to multiple input, multiple output (MIMO) systems:

(1) Effective treatment of coupled input interactions to improve performance,

(2) Rigorous treatment of stability questions,

(3) Systematic control design which reduce iteration and the need for extensive intuition and experience in the control design process.

The most extensive application of modern control theory to date is the F100 Turbofan Multivariable Control Synthesis Program, sponsored by the Air Force Aero Propulsion Lab and Nasa Lewis Research Center. [Ref. 1: p.43]. The results of this program demonstrated that modern control theory techniques provide an orderly, effective, systematic approach to controller design for multiple, interacting input systems.

Current work at the Naval Postgraduate School, Monterey, is aimed at investigating the application of modern control techniques to U.S. Navy ship propulsion plants. The initial phase of this effort involves the application of modern control techniques to a low power gas turbine test facility located at the school. In the context of this effort the goals of this thesis were:

(1) Development of an accurate digital computer model of the propulsion test facility which includes all significant plant nonlinearities and dynamic effects.

(2) Develop a linear (state-space) model of the propulsion plant.

The nonlinear model is a valuable tool in controller design. First, it provides a means to test control strategies without risking damage to the actual plant. Second, it provides a cost effective alternative to extensive controller tests. Finally, in this work the nonlinear model provided the basis from which a linear (state space) model was derived.

The state space model of the propulsion plant is essential for future controller design work at NPS using the linear quadratic regulator (LQR) technique, which is the most highly developed modern control method. [Ref. 2: p.653].

II. OVERVIEW

This thesis is organized into eight chapters. In the following chapter a description of the test facility at the Naval Postgraduate School (NPS) is given. Also in that chapter, a conceptual model of the plant is developed. The plant is first divided into functional components. The significant plant dynamics are identified, and the number of plant components is reduced so that only the degree of complexity necessary to represent the significant plant processes is retained. Finally, the component interactions are defined by identifying component inputs and outputs. The component interactions link the components together and form a basis for quantitative modeling of the plant.

In Chapter 4 quantitative (versus conceptual) component modeling is described. Using experimental data, the input/output relations for each component (as defined in the previous chapter) are constructed in equation form.

In Chapter 5 the individual component equations are joined together to form a steady state plant model.

In Chapter 6 the differential equations governing the plant dynamics (identified in Chapter 3)are introduced into the steady state model. In this way a nonlinear dynamic plant model is developed.

In Chapter 7 the state space model is derived from the nonlinear dynamic model.

Chapter 8 contains conclusions and recommendations for further work.

III. PLANT DESCRIPTION/CONCEPTUAL MODELING

A. PLANT DESCRIPTION

The test facility (hereafter referred to as 'the plant') consists of a Boeing model 502-6A 175 horsepower gas turbine engine and a Clayton 17-300 water dynamometer. Figure 3.1 is a schematic of the test facility.

The gas turbine can be subdivided into the gas generator section and the power take-off section. The gas generator section consists of a single stage centrifugal compressor, a dual can combustor, an accessory drive section, and a single-stage axial flow high pressure turbine (HPT). The power take-off section consists of a single-stage axial flow free power turbine (FPT). Fuel enters the combustor via the fuel control, which consists of a flyball governor and an acceleration limiter. The fuel control setting is adjusted by an electric-motor driven control lever. The hot gases drive the high pressure turbine. The high pressure turbine, in turn, drives the compressor and accessory drive section. The accessory drive section extracts power from the high pressure turbine to drive the fuel pump and governor, the oil pump, and a tachometer generator. The free power turbine extracts energy from the hot gas stream and drives the dynamometer. There is no mechanical connection between the high pressure turbine and the free power turbine.

The water dynamometer acts as a power absorption unit. The power turbine torque and speed are adjusted by varying the amount of water in the dynamometer. This is analogous to changing the pitch on a controllable pitch propeller. Water enters and leaves the dynamometer via electric motor driven load and unload valves.

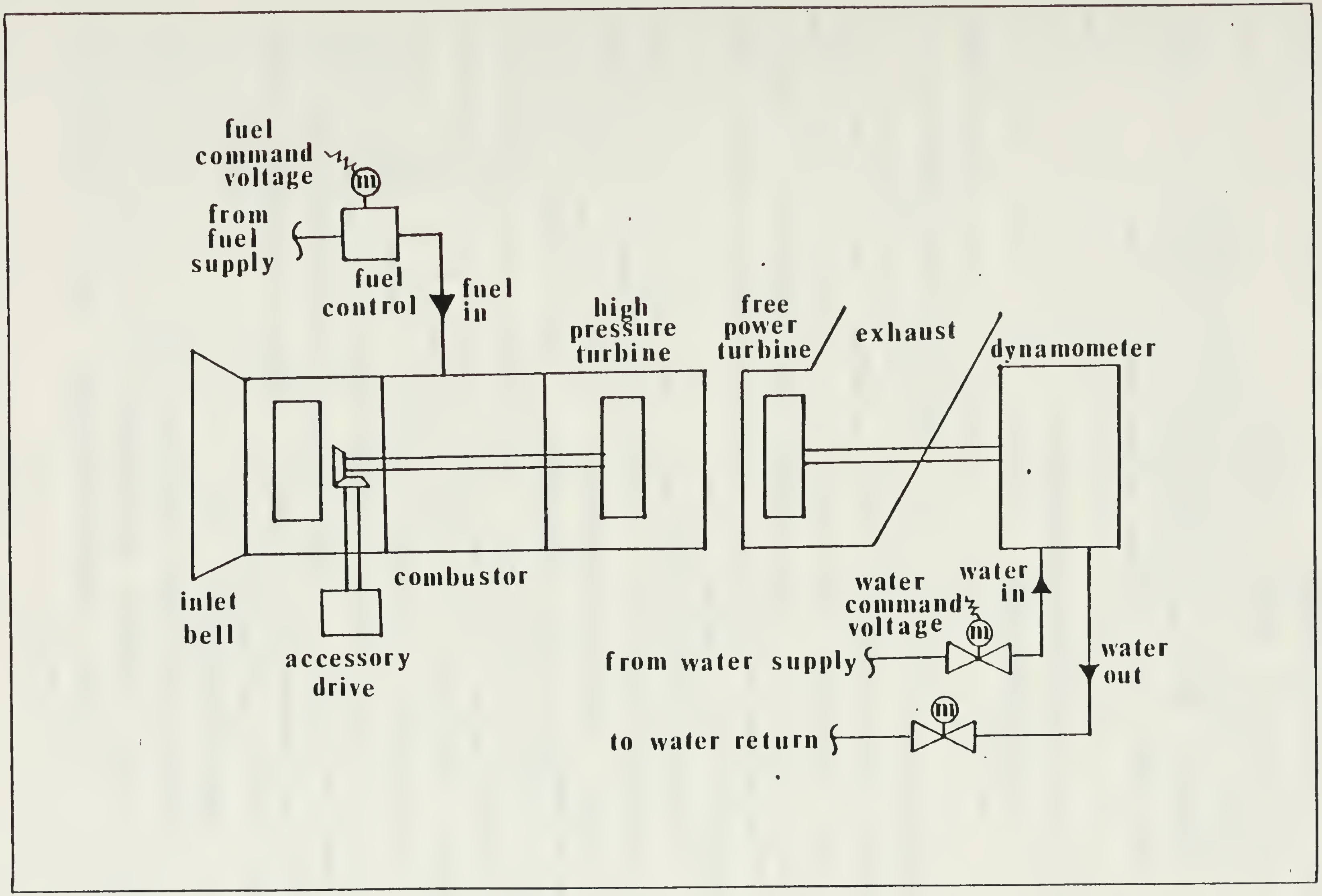

Figure 3.1 NPS Marine Propulsion Test Facility

B. THE CONCEPTUAL PLANT MODEL

In formulating a conceptual plant model the following issues must be addressed:

- definition of plant boundaries,
- identification of significant plant dynamics,
- extent to which the plant must be divided into components and how the components selected, and
- definition of component inputs and outputs.

These issues are discussed below.

1. Plant Boundaries

Selection of plant boundaries is important since this selection determines the plant inputs. Because the long range goal of this project is the implementation of an improved control system, the existing fuel control was excluded from the plant model. The plant boundary was located downstream of the existing fuel control, and actual fuel flow to the combustor (versus fuel command voltage) was established as one plant input.

A model of the dynamometer was developed in earlier work by Johnson [Ref. 3]. This model accurately describes the behavior of the dynamometer during loading conditions (water addition to the dynamometer), but is less accurate during unloading conditions. The source of the dynamometer model inaccuracies is thought to involve the unload valve behavior. In order to avoid introduction of the unload valve inaccuracies into the propulsion model developed in this study, the load and unload valves were placed outside the plant boundaries. Thus, actual water to and from the dynamometer (versus water command voltage) was established as the second plant input.

2. Plant Components

Breaking the plant into components facilitates the identification of causual relationships and plant dynamics. Further, the plant components provide the foundation of plant model development.

The selection of components is a cut and try process. A first cut at component identification is shown in Figure 3.2.
This selection was based on a functional basis and assumptions about the significant processes occuring within the plant. The minimum number of components necessary to account for these processes is sought. If the resulting model is insufficiently accurate, then some significant process has either been overlooked or improperly described. In either case, the selection of components must be reevaluated.

3. Significant Dynamics

Paramount to model accuracy is the identification of significant plant dynamics. Fluid momentum and compressibility, heat transfer, energy storage, rotor inertia, and combustion effects are all possibilities. However, dynamic effects can only be considered significant in the practical sense if their time constants are neither much shorter nor extremely longer than those for the controller actuators and sensors. Previous work by Szuch [Ref. 4:p. 243] in the area of aircraft gas turbine controls indicated that fluid momentum, compressibility, and energy storage dynamics occur too rapidly to be controlled, while heat transfer dynamics occur too slowly to be important in the control problem.

The importance of combustion dynamics deserves special discussion. Szuch [Ref. 4:p. 243] and DeHoff [Ref. 5:p. 274] concluded that combustion dynamics were of too high frequency to be important in controls considerations. In contrast, Rubis [Ref. 6:p. 56] discusses significant transient effects associated with engine torque development in response to fuel flowrate changes. These effects could be due in part to combustion related delays. In the present work combustion effects were initially neglected. The resulting model produced excessive

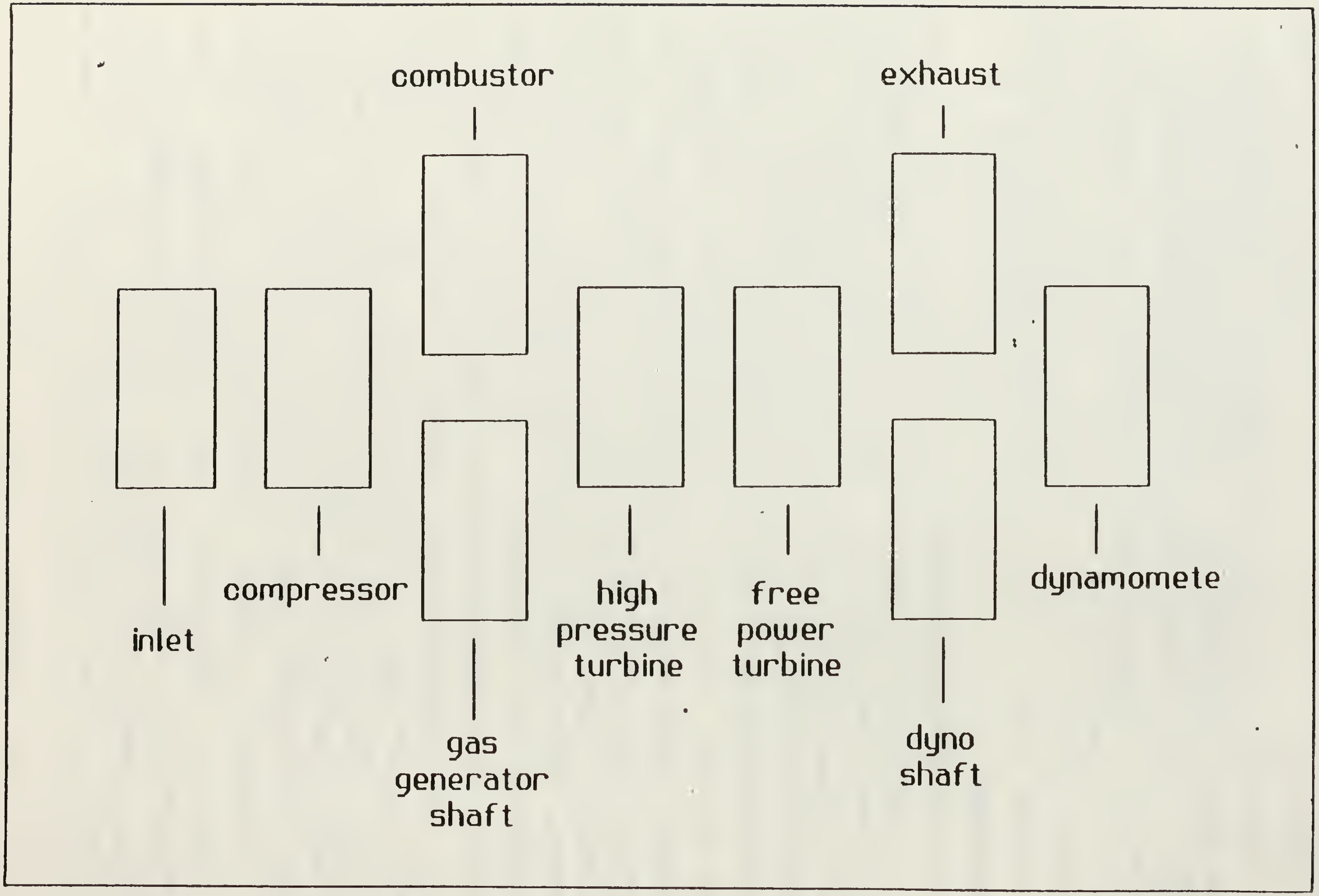

Figure 3.2 Propulsion Plant Components

accelerations when compared to experimental data. When combustion effects were modeled as a fuel energy lag accurate results were achieved. This fuel energy lag represents the delay between the time when the chemical energy in the fuel passes through the fuel nozzles and the time when the mechanical energy is realized at the high pressure turbine.

In addition to the fuel energy lag, the most significant plant dynamics are the rotor inertia effects. Thus, the equations governing the plant dynamic behavior are:

$$\dot{NG} = (QH-QC-QA-QFRG)/JG$$

$$\dot{NS} = (QF-QD-QFRD)/JD$$

$$E/Mf = 1/(tS + 1)$$

Where NG = gas generator acceleration,
NS = free power turbine acceleration,
JG = gas generator inertia,
JD = combined free power turbine and dynamometer inertia,
QH = high pressure turbine torque,
QC = compressor torque,
QA = accessory drive torque,
QFRG = gas generator frictional torque,
QF = free power turbine torque,
QD = dynamometer torque,
QFRD = combined free power turbine and dynamometer frictional torque,
Mf = measured fuel flowrate at the fuel nozzles,
E = mechanical energy applied at the high pressure turbine,
t = time constant associated with the combustion process.

In this study the auxiliary torque (QD) and gas generator frictional torque (QFRG) were lumped into the compressor torque (QD). Also, the dynamometer/power turbine frictional torque (QFRD) was lumped into the dynamometer torque (QD). Having made these simplifications the governing dynamic equations become:

$$\dot{NG} = (QH-QC)/JG \qquad (3.1)$$

$$\dot{NS} = (QF-QD)/JD \qquad (3.2)$$

$$Mf/E = 1/(tS + 1) \qquad (3.3)$$

4. Model Simplification

Once the significant dynamic effects are determined and the governing equations identified, the plant model may be simplified by reducing the number of components. As noted in the overview, it will be necessary during the model development to define the input/output relationships for each component in equation form. If the dynamic effects can be lumped into isolated components, then the input/output equations for these components are already known; they are the governing differential equations for the plant dynamics. Further, if the dynamic effects are lumped into isolated components, then the input/output equations for the remaining components can be obtained from steady state data since these components are assumed to contain no dynamic effects. This approach was applied to the current modeling problem and the reduced component model of Figure 3.3 was produced.

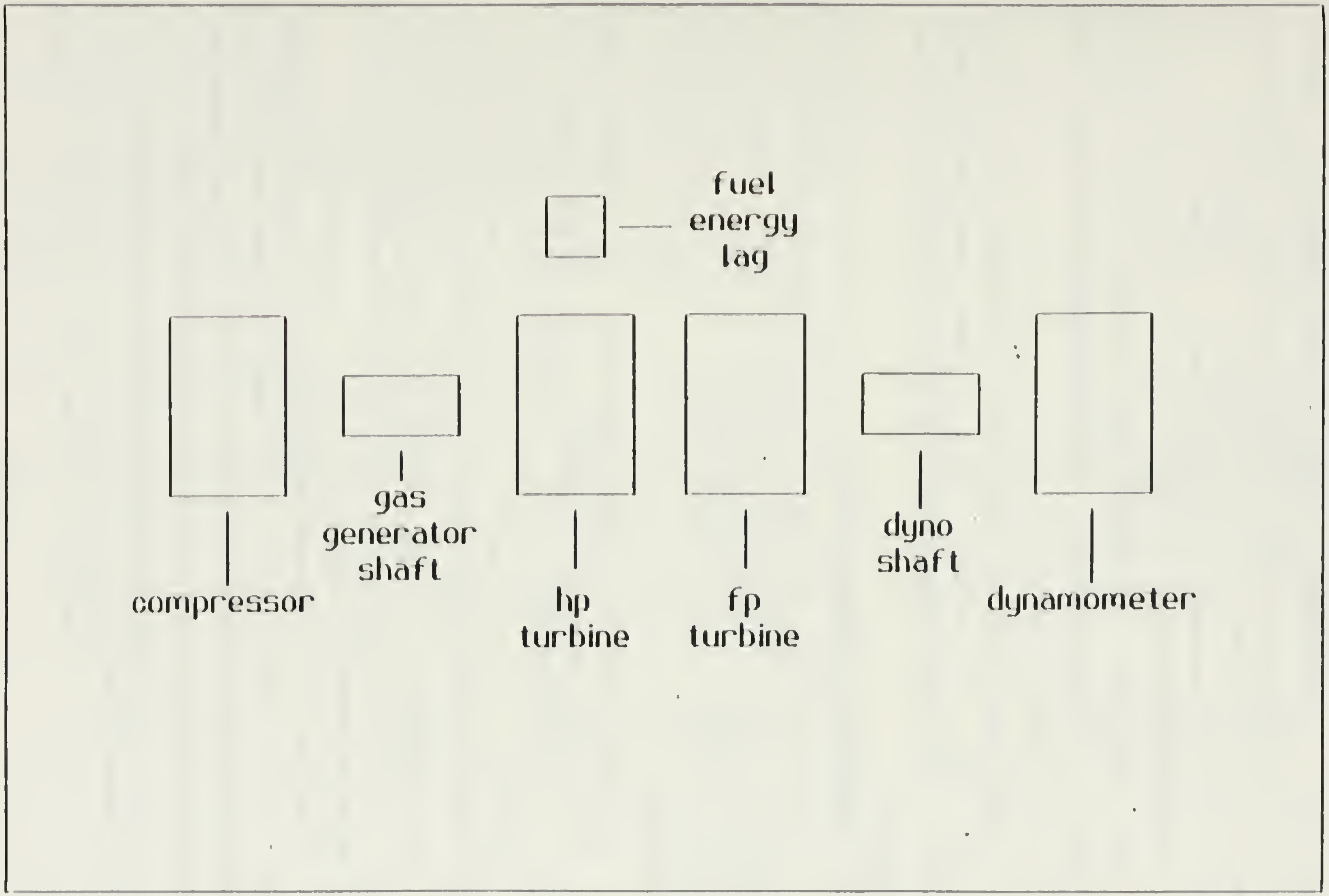

Figure 3.3 Reduced Component Model

In this work two dynamic effects are the accelerations of the gas generator and the power turbine/dynamometer. By lumping the entire gas generator inertia into the gas generator shaft, the gas generator acceleration dynamics were isolated to that component. Similarly, the power turbine/dynamometer dynamics were lumped into the power turbine/dynamometer shaft. The third dynamic effect, the fuel combustion dynamics, was lumped into a single component, a first order lag between the fuel flowrate input and the high pressure turbine.

Since no dynamic effects were considered to occur in the inlet bell or exhaust duct, these components were combined with the compressor and free power turbine, respectively.

5. Component Inputs/Outputs

The next step in formulating the conceptual model is determining the inputs and outputs of each component. Multiport analysis using signal pairs at "ports" of power transfer is one useful method for studying component interaction.

At the mechanical-rotational port (gas generator and power turbine/dynamometer shafts) the signal pair is torque, Q, and rotational speed, N, as shown in Figure 3.4. More difficult to represent is the thermofluid power transfer between the compressor, high pressure turbine, and free power turbine. If the fluid flow were incompressible the port would be represented by a pressure-volume flowrate signal pair as shown in Figure 3.5.

However, this simple representation is not adequate for the case of compressible flow since it does not account for thermal energy transfer via the fluid internal energy. One could represent this port as shown in Figure 3.6 where $U = M * u$, with M the mass flowrate and u the fluid specific internal energy.

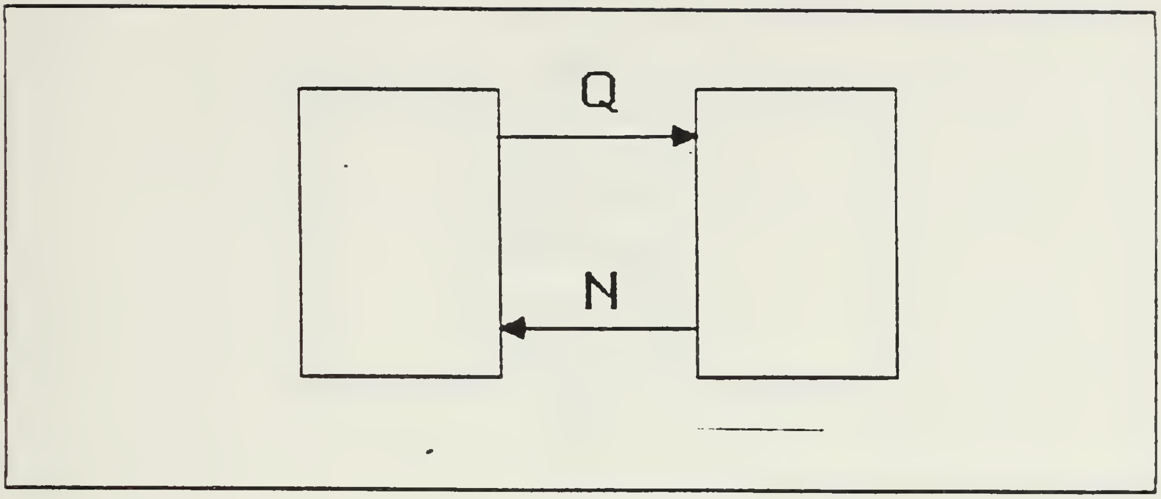

Figure 3.4 Mechanical-Rotational Port

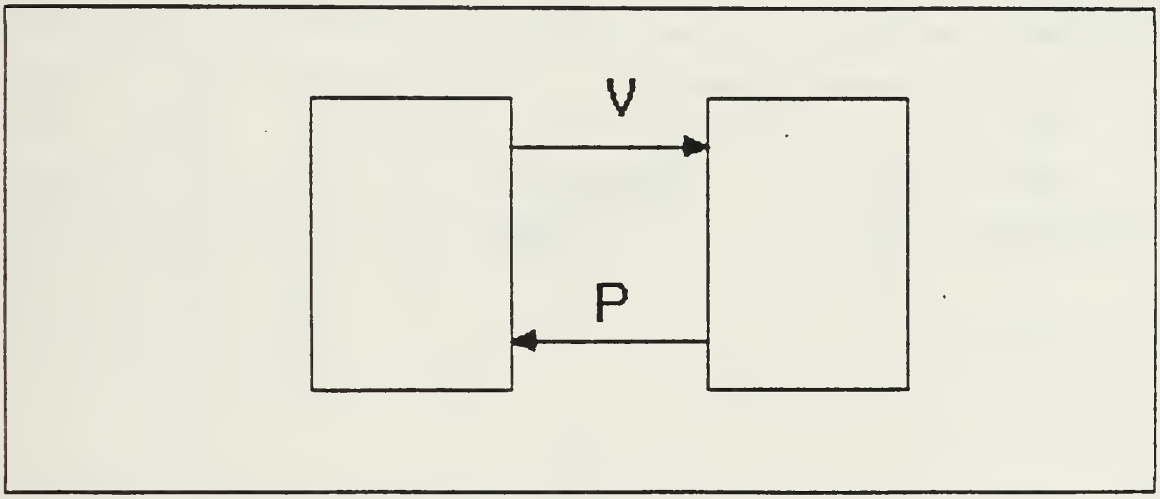

Figure 3.5 Incompressible Fluid Port

Representing internal energy as the product of specific heat (Cv) and temperature, the port could be rewritten as shown in Figure 3.7.

One might now attempt to combine effects and describe the complete thermofluid power transfer as shown in Figure 3.8.

However, since the variables P,V,M, and T are related through the equations of state, it is redundant to measure all four in the modeling process.

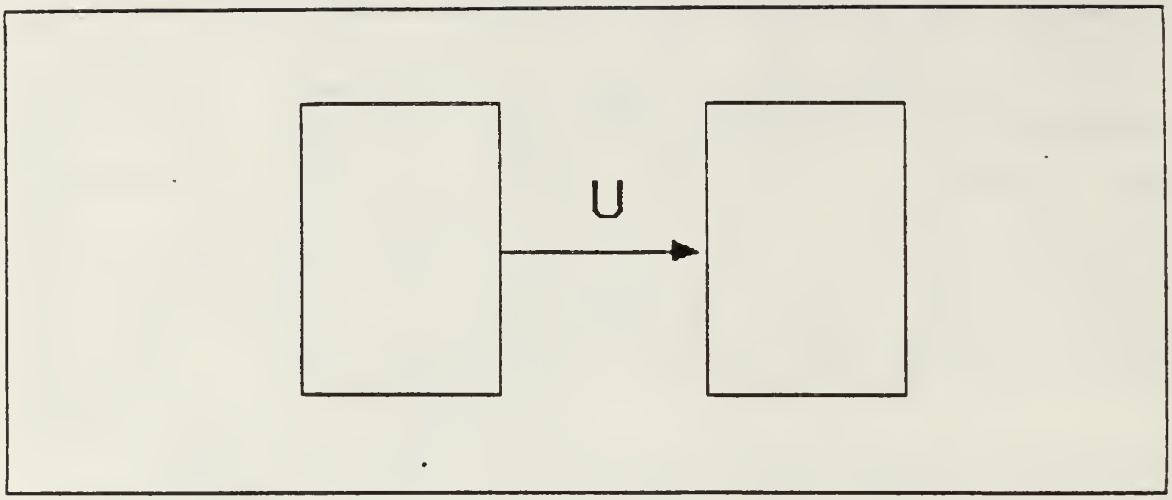

Figure 3.6 Thermal Port

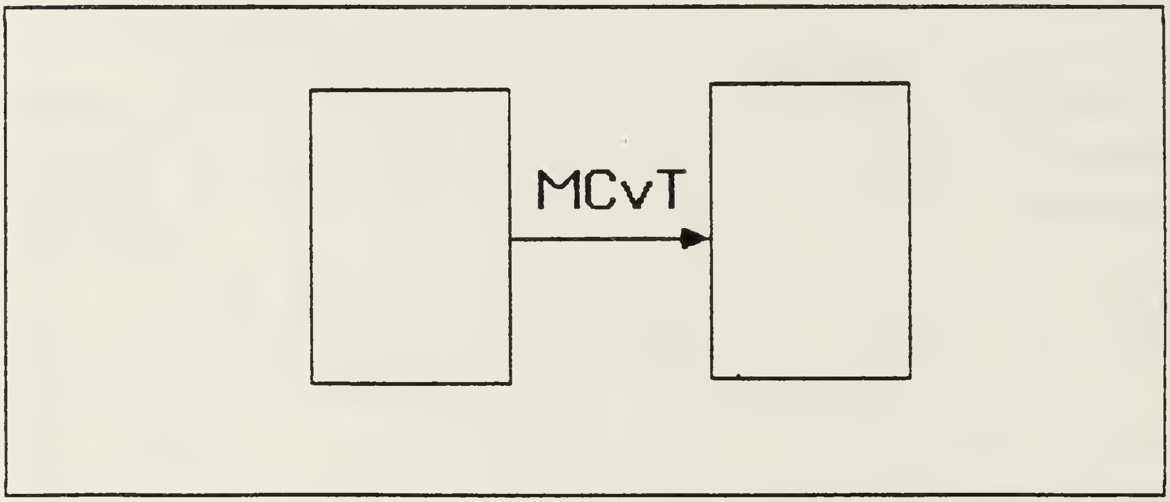

Figure 3.7 Alternate Thermal Port

In considering which variable to eliminate it should be noted that since the direction of mass flow and volume flow must be the same, and since temperature is "carried along" with the mass flow, these three variables have the same signal flow direction. Thus, elimination of the pressure variable was considered unwise since this would essentially eliminate the two-way component interaction. Elimination of temperature from the set of independent

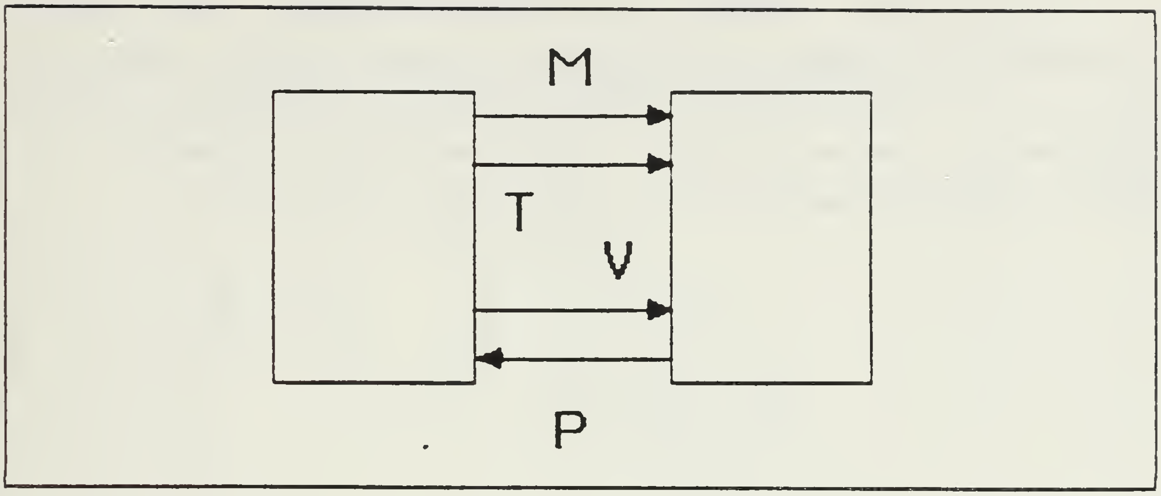

Figure 3.8 Combined Port

variables was considered a poor choice due to its ease of measurement relative to either mass or volume flowrate. Finally, since mass flowrate is conserved while volume flowrate is not, and since this conservation might lead to simplification in future measurements and calculations, it was decided that volume flowrate, V, would be eliminated. The resulting power port is shown in Figure 3.9.

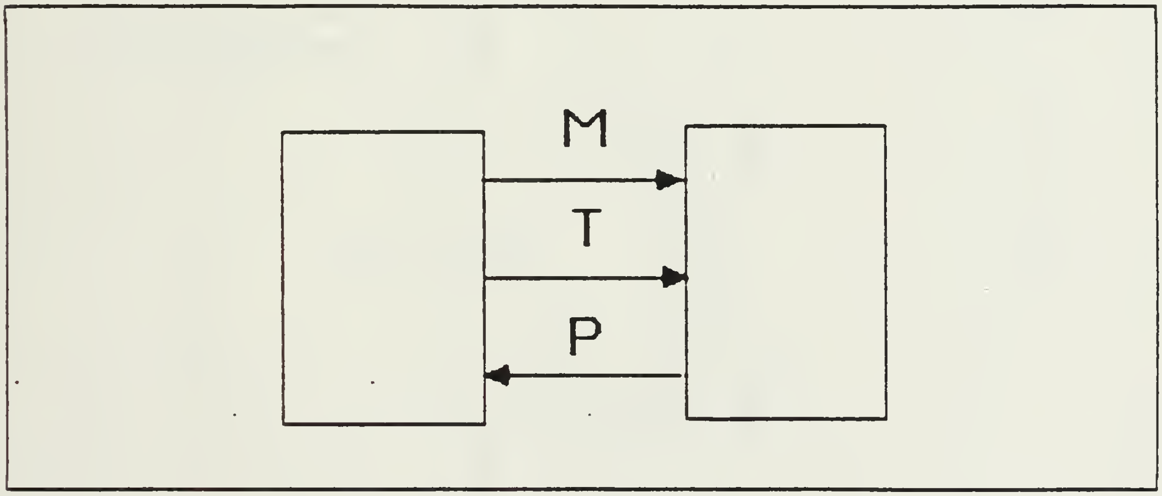

Figure 3.9 Thermodynamic Power Port

Using this concept of the thermodynamic power port the complete multiport diagram was constructed as shown in Figure 3.10. Note that variable ambient conditions were eliminated as system inputs by using corrected variables, a commonly employed technique in gas turbine analysis.

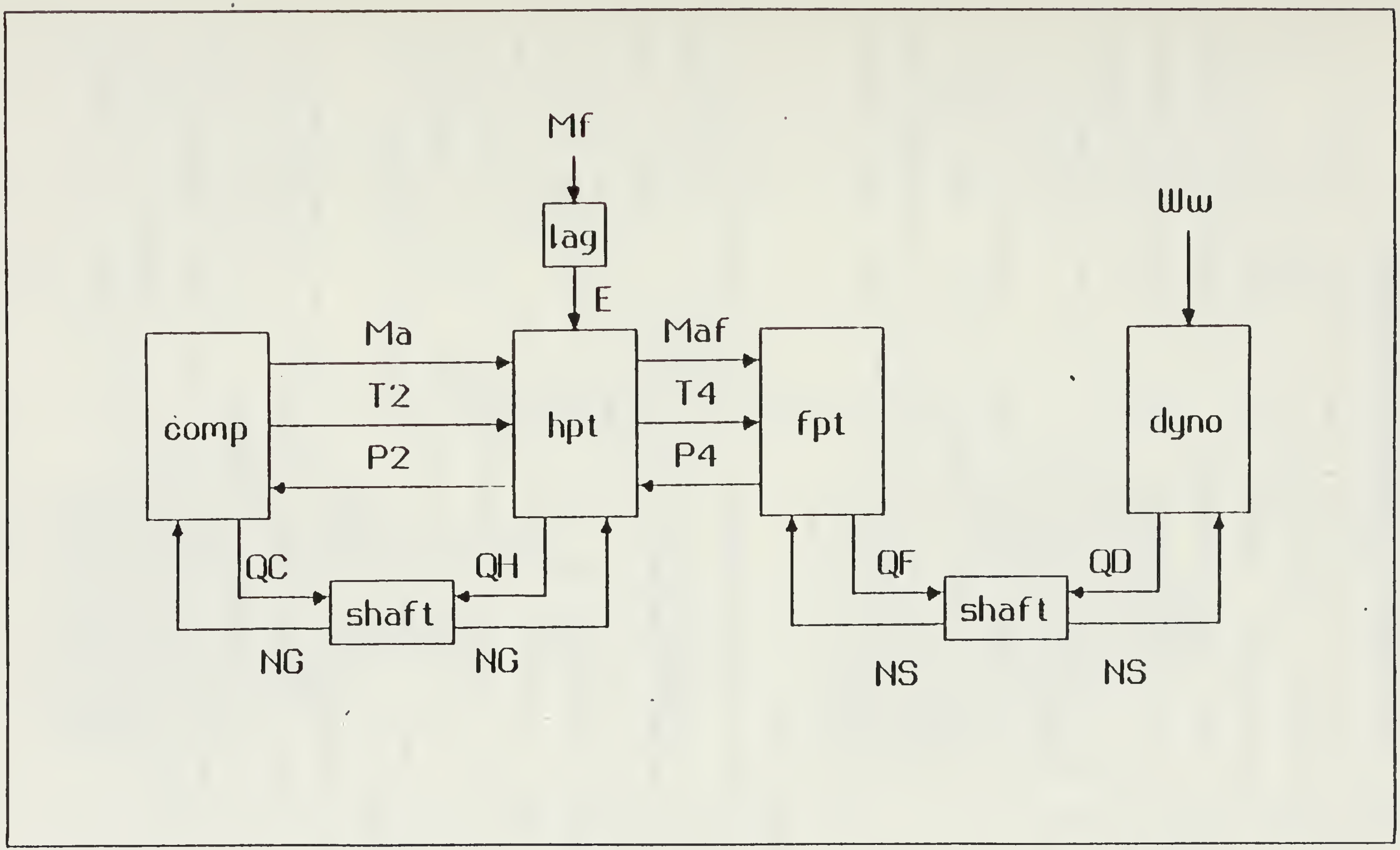

Figure 3.10 Complete Multiport Diagram

IV. QUANTITATIVE COMPONENT MODELING

From the multiport diagram of the previous chapter the component input/output relationships are identified. The next step in the modeling process is to obtain quantitative expressions for these relationships. In this model we have assumed that the gas generator and power turbine/dynamometer shafts, and the fuel energy lag contain the plant dynamic effects. The quantitative relation for these components is the governing differential equations 3.1 , 3.2 and 3.3. Further, since it has been assumed that the remaining components contain no significant dynamic effect, the input/output equations for these components can be obtained from steady state data. The following describes how these equations were obtained.

A. DATA ACQUISITION

The gas turbine and dynamometer is instrumented as indicated in Table 1. The data acquisition system is controlled by an HP-85 personal computer. Temperature, speeds, and torques are taken using an HP-3497A Data Acquisition/Control Unit. Pressure readings are taken using a Pressure Systems DPT-6400. Fuel flowrate is obtained by the operator from two rotometers and entered interactively or from a turbine flowmeter. An HP-82902M flexible disk drive provides program/data storage capability. A Digital DECWRITER IV printer provides hard copy output.

Data was taken at 93 operating points as indicated in Table 2. These points were selected to provide full coverage of the plant operating envelope. At each operating point temperature, speed, and torque values were sampled 30 times and averaged. Pressure values were sampled 8 times and averaged. A copy of the acquisition program is included as Appendix A.

TABLE 1

GAS TURBINE/DYNAMOMETER INSTRUMENTATION

Parameter	Symbol	Instrumentation
Compressor Inlet Temp.	T1	4 type T thermocouples
Compressor Disch. Temp.	T2	4 type T thermocouples
High Pressure Turbine Inlet Temp.	T3	4 type K thermocouples
High Pressure Turbine Disch. Temp.	T4	4 type K thermocouples
Cell Pressure	P0	2 static pressure probes
Inlet Bell Pressure	P1	2 static pressure probes
Compressor Disch. Pressure	P2	2 static pressure probes
High Pressure Turbine Disch. Press.	P4	1 static pressure probes
Gas Generator Speed	NG	1 tachometer generator
Dynamometer Speed	NS	1 RF speed sensor
Dynamometer Torque	QD	1 RF torque sensor
Fuel Flowrate	MF	2 Rotometers
Fuel Flowrate	MF	1 Turbine Flowmeter

B. DATA REDUCTION

Data reduction took place in two phases. In the first phase average values, generator torque, air mass flowrate, and corrected values were computed. These calculations are part of the data acquisition program (Appendix A).

In the second phase, curve fits were obtained for the input/output relations of each turbine component using the method of least squares [Ref. 7: p. 153] The program which performed the least squares fit is included as Appendix B. Using this program, three types of curve fit were obtained for each input/output relation. The first is a "complete quadratic" curve fit shown in equation 4.1,

$$Y = C1*X1^2 + C2*X1*X2 + C3*X2^2 + C4*X1 + C5*X2 + C6 \quad (4.1)$$

where Y = output (dependent) variable,
X1,X2 = input (independent) variables,
C1,C2,C3,C4,C5,C6, = constant coefficients.

The second type of curve fit obtained was a "reduced quadratic", so called because the cross product terms (ie., X1*X2) found in the "complete quadratic" curve fit were excluded. Equation 4.2 is an example of this format.

$$Y = C1*X1^2 + C2*X2^2 + C3*X1 + C4*X2 + C5 \quad (4.2)$$

The third curve fit type was a linear curve fit. Equation 4.3 is an example of this format.

$$Y = C1*X1 + C2*X2 + C3 \quad (4.3)$$

In each case the result of the curve fit program is the coefficients of the curve fit equation. Because the magnitude of some variables is much larger than others (ie., NG = 30,000 rpm, P4 = 17 psia) it was necessary to scale the

TABLE 2

GAS TURBINE DATA ACQUISITION SCHEDULE

GAS GENERATOR SPEED, NG (RPM)	DYNAMOMETER SPEED, NS (RPM)	GAS GENERATOR SPEED, NG (RPM)	DYNAMOMETER SPEED, NS (RPM)
19078.73	549.12	19400.72	563.30
21128.36	557.66	21829.68	557.98
22547.57	562.21	23208.31	563.58
23891.30	566.05	24584.86	569.89
25280.03	565.37	25922.19	573.93
26665.44	567.74	27331.84	569.30
28726.66	576.29	29382.51	580.98
30067.85	580.45	30825.73	587.61
31525.20	583.29	32533.96	584.86
33488.27	593.19	19080.00	935.15
19254.72	946.33	21160.08	958.80
21854.58	954.31	22570.12	952.79
23270.02	957.89	23968.66	953.97
24516.74	954.13	25285.25	959.41
25983.85	968.10	26635.89	965.15
27296.90	964.39	28080.53	969.07
28796.48	971.97	29448.34	976.55
30080.00	989.08	30824.83	979.37
31543.90	977.28	32476.51	978.31
33490.05	981.56	34504.89	983.94
35372.62	996.06	19088.39	1428.13
21112.06	1441.88	21852.49	1431.24
22577.27	1444.77	23202.79	1436.46
23896.71	1445.22	24476.83	1438.68
25259.17	1442.62	25919.51	1443.97
26508.37	1455.67	27302.78	1453.57
27917.37	1452.72	28612.59	1446.09
29370.06	1468.75	30036.55	1458.07
30694.21	1460.35	31568.65	1456.94
32464.30	1459.62	33358.12	1461.41
23925.74	1934.28	24574.83	1953.73
25269.44	1943.66	25845.15	1943.32
26637.27	1944.69	27287.50	1946.16
27807.57	1944.34	28665.43	1947.56
29131.64	1933.95	29985.26	1954.42
30443.17	1939.51	31485.56	1961.59
32379.32	1974.51	33144.56	1938.01
34523.04	1951.02	35366.35	1963.19
27344.96	2432.68	28000.63	2421.92
28696.43	2441.53	29346.11	2433.45
30057.51	2436.86	30736.87	2429.97
31489.43	2431.06	32441.26	2441.16
33358.05	2440.31	29370.30	2922.37
30060.52	2918.94	30653.77	2906.41
31528.78	2927.09	32413.38	2917.06
33291.47	2930.93	34366.77	2917.24

variables to prevent algorithmic singularities during the least squares solution. This was accomplished by dividing each variable by a scaling factor. The scaling factors, shown in table 3, were selected so that each variable had a range of 0.0 to 1.0 when scaled.

TABLE 3

SCALING FACTORS

Variable	Scaling Factor
Gas Generator Speed, NG	36,000 rpm.
Compressor Torque, Qc	130 ft. lb.
Air Mass Flowrate, Ma	13,000 lb/hr.
Compressor Discharge Temperature, T2	800 deg. R.
Compressor Discharge Pressure, P2	43.0 psia.
Fuel Mass Flowrate, Mf	240 lb/hr.
High Pressure Turbine Torque, QH	130 ft. lb.
Combined Air/Fuel Mass Flowrate, Maf	13,000 lb/hr.
High Pressure Turbine Discharge Temp., T4	1800 deg. R.
High Pressure Turbine Discharge Press., P4	20.0 psia.
Dynamometer Speed, NS	3,000 rpm.
Free Power Turbine Torque, QF	480 ft. lb.

V. STEADY STATE PLANT MODEL

A. STEADY STATE MODEL ALGORITHM

With the input/output component relations defined in equation form the next step was to link these relations together to form a steady state plant model. Figure 5.1 presents the flowchart describing the steady plant state model algorithm. In this algorithm the program user inputs the gas generator and dynamometer speeds at which the plant parameters are to be evaluated. The program makes an initial guess at the steady state fuel flowrate, Mf. A guess is also made for the compressor and high pressure turbine discharge pressures, P2 and P4. The program uses these assumed values to calculate the compressor and high pressure turbine outputs. The computed compressor discharge pressure is compared with the assumed value. If the difference between assumed and computed value ceeds the specified tolerance, the value of P2 is updated. Otherwise, the power turbine outputs are calculated and the computed high pressure turbine discharge pressure is compared with the assumed value. Convergence within the specified tolerance is again required. If this check is
met the compressor and high pressure turbine torques are compared.
These torques should be equal in steady state. If they are not the assumed value of fuel flowrate is updated and the entire process is repeated. The steady state computer model is included as Appendix C.

B. STEADY STATE MODEL RESULTS

The steady state computer program was tested at various points in the plant operating envelope. The output of the computer model was compared with the raw data for the same

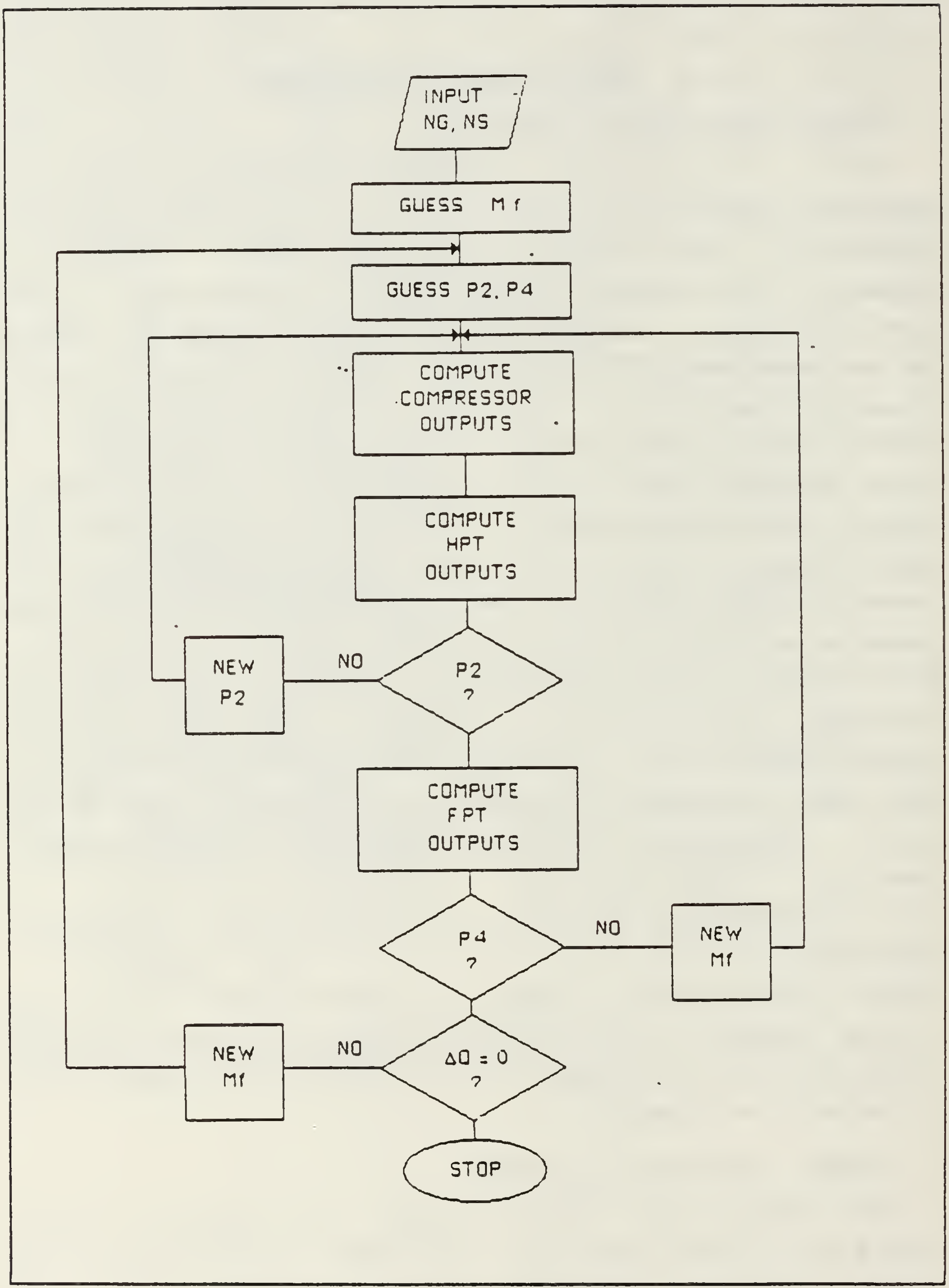

Figure 5.1 Steady State Plant Model Flowchart

operating points. The results showed excellent agreement between the steady state model and the raw data at most operating points. The only exception to this was at extremely high gas generator speeds. A typical comparison is shown in Table 4.

As indicated by the flowchart of Figure 5.1, at any given gas generator speed (NG) and dynamometer speed (NS) combination, the program must hunt for the fuel flow rate that will produce zero torque differential between the compressor and high pressure turbine. An investigation into the nature of the torque differential as a function of fuel flowrate (at fixed NG and NS) provides some interesting insights into the performance characteristics of the engine. As an example, in Figure 5.2 the torque differential is plotted versus fuel flow for NG = 30,000 rpm and NS = 1,100 rpm.

This plot has several interesting features. First, note that there are two values of fuel flowrate that lead to zero torque differential. This suggests that at a given NG/NS combination there are actually two equilibrium conditions. It is thought that the upper fuel flowrate represents a state of inefficient operation. Whatever the source of the higher fuel flowrate condition, comparison with the raw data indicates that the normal operating mode of the turbine is at the lower fuel flowrate.

A second feature of this curve is the peak torque differential. This peak is considered to be a significant characteristic of the gas generator. It is thought to be indicative of the maximum driving torque difference attainable at the specified NG/NS combination. Thus, if one wishes to accelerate the gas generator quickly, this plot indicates the limits of achieveable acceleration as well as the optimal fuel input to achieve the greatest acceleration. Clearly, more fuel does not necessarily lead to greater

TABLE 4

COMPARISON OF STEADY STATE OUTPUT WITH RAW DATA FOR NG = 25,900 RPM, NS = 970 RPM

Parameter	Symbol	Steady State Computer Output	Raw Data
Compressor Torque	QC	69.8 ft. lb.	69.4 ft. lb.
Compressor Disch. Temp.	T2	658 deg. R.	550 deg. R.
High Pressure Turbine Disch. Temp.	T4	1354 deg. R.	1361 deg. R.
Compressor Disch. Pressure	P2	27.41 psia.	27.42 psia.
High Pressure Turbine Disch. Press.	P4	16.36 psia.	16.36 psia.
Free Power Turbine Torque	QF	151.7 ft. lb.	150.5 ft. lb.
Fuel Flowrate	Mf	110.8 lb/hr.	110.3 lb/hr.
Air Mass Flowrate	Ma	8696 lb/hr.	8742 lb/hr.

acceleration. In fact, this plot suggests that too large a fuel increase can lead to deceleration. One should note, however, that this condition may be impossible to achieve practically, since it presumes that this large fuel change can be made instantly, without changing the NG/NS combination (ie., a perfect step). The fuel energy lag dynamics seem to exclude this possibility in the current application. Further, the existing fuel control devices on most gas turbine facilities would likely prevent this condition from being observed.

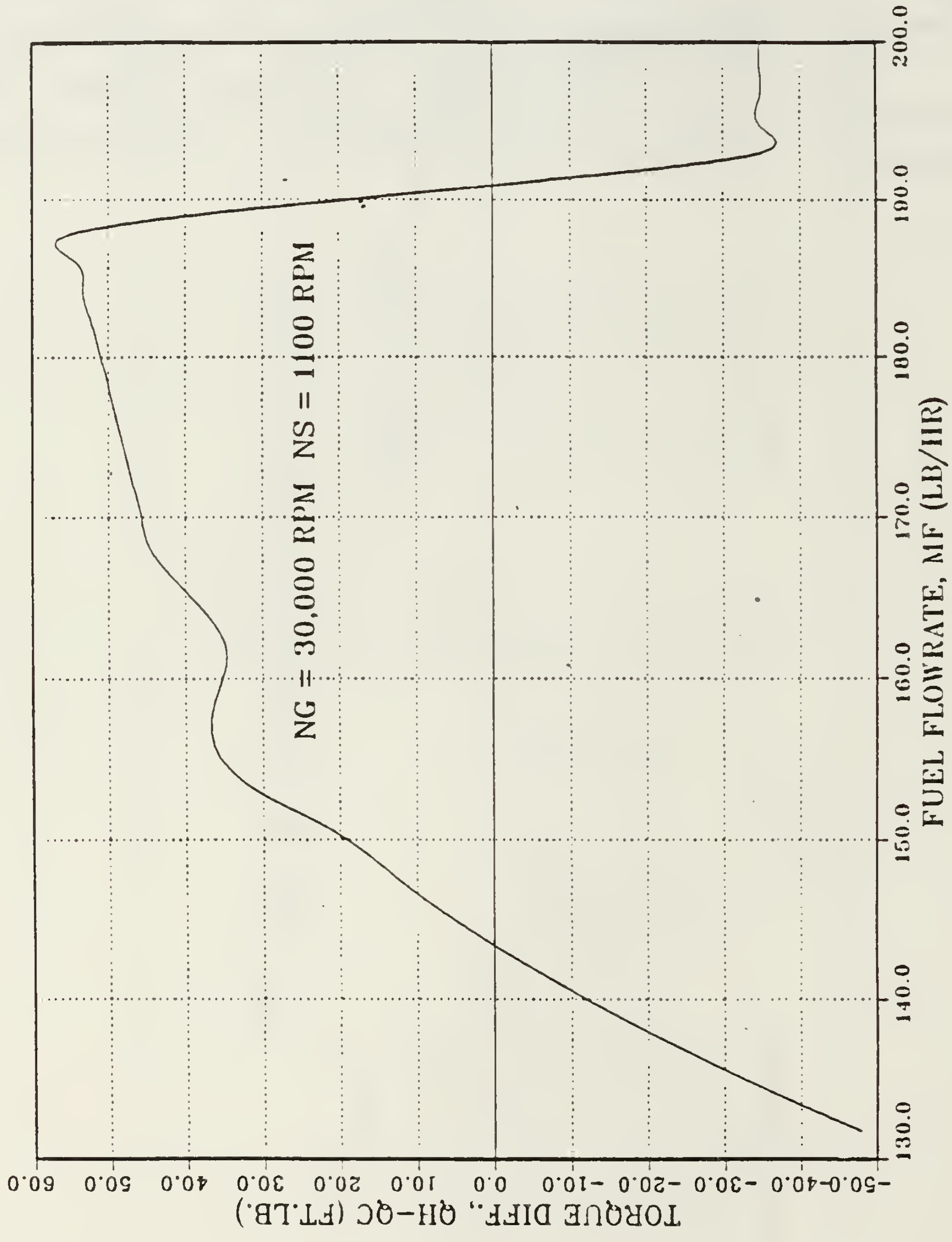

Figure 5.2 Torque Differential .vs. Fuel Flowrate

VI. NONLINEAR DYNAMIC MODEL

The next step was to develop a nonlinear dynamic model by introducing the plant dynamic equations into the steady state model.

In order to implement the governing dynamic equations, the gas generator and power turbine/dynamometer inertias must be known. Johnson [Ref. 3:p. 56] concluded that the combined power turbine/dynamometer inertia (JD) is insensitive to dyno water weight and that a value of JD = 0.6738 lb.ft.s^2 was valid throughout the operating range.

The following section describes the determination of the gas generator inertia. Subsequently, the development and results of the nonlinear dynamic model is described.

A. GAS GENERATOR INERTIA

The technical manual for the Boeing 502-6A gas turbine engine [Ref. 8:p. 6] lists the gas generator inertia (JG) as 0.11 in.lb.s^2. However, it was unclear whether this inertia value included the accessory gearbox inertia. Further, the equipment configuration at the NPS test facility is somewhat different than the standard configuration described in the technical manual. Because the gas generator inertia is crucial to accurate dynamic performance prediction, it was desirable to verify the technical manual value experimentally.

In order to experimentally determine the gas generator inertia, the gas generator inlet bell and nose cone was removed. A lever arm was attached to the compressor impeller using existing bolt holes intended for impeller removal. [Ref. 8:p. VII-10]. A spring was attached at each end of the lever arm and secured to the base of the turbine. The resulting experimental set-up is shown in Figure 6.1.

Finally, a potentiometer was attached to the lever arm and aligned with the impeller shaft centerline to permit measurement of impeller angular position.

The lever was deflected and released from rest. Using a strip chart recorder the oscillatory motion of the gas generator was recorded. This procedure was repeated ten times so that good average values could be obtained.

A simplified diagram of the experimental apparatus is given in Figure 6.2, where K is the effective spring constant, J is the total polar mass moment of inertia about the gas generator axis, θ is the angle of rotation, and c is the system damping due to friction. From this simplified diagram the differential equation for viscously damped free vibration is found to be:

$$J \ddot{\theta} + C \dot{\theta} + K \theta = 0 \qquad (6.1)$$

The solution to this equation [Ref. 9: p. 25-32] for the underdamped case reveals the frequency of damped oscillation to be:

$$Wd = Wn \sqrt{1- \zeta^2} \qquad (6.2)$$

where Wd = frequency of damped oscillation

Wn = natural frequency

ζ = damping ratio.

Using average values, the frequency of damped oscillation was determined from the strip chart readings to be:

$$Wd = 67.544 \text{ rad/sec}$$

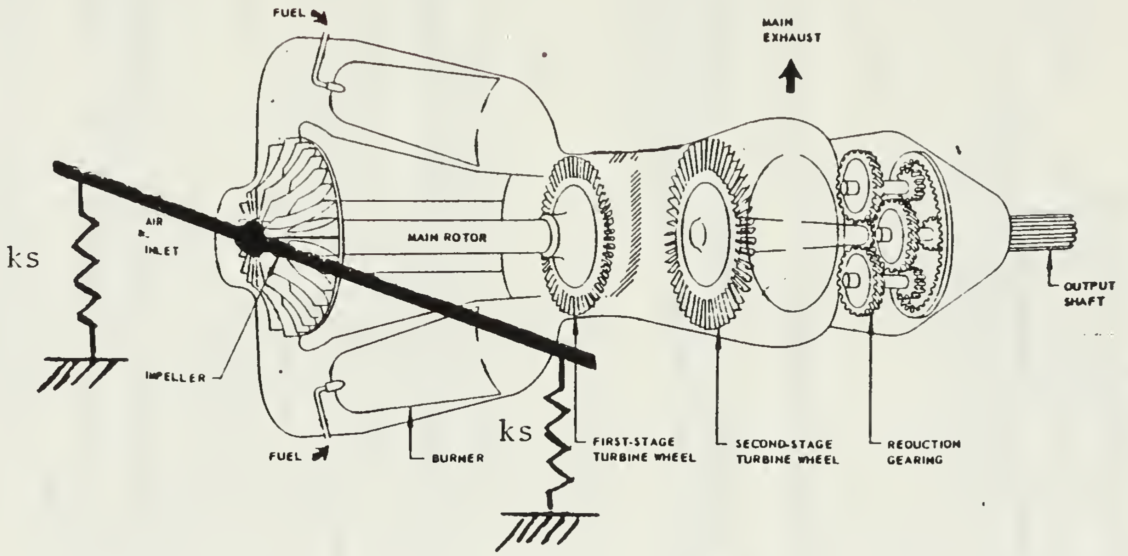

Figure 6.1 Experimental Apparatus for JG Determination
[Ref. 8: fig. 16., p. V-1]

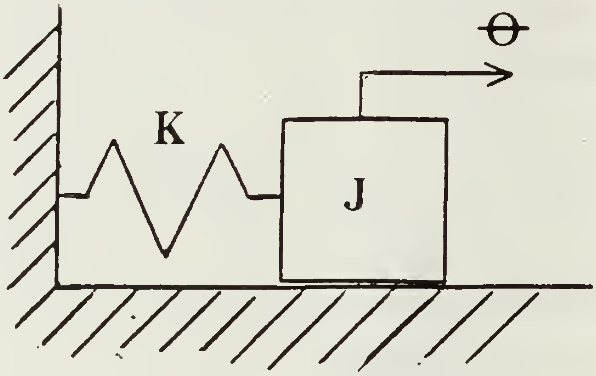

Figure 6.2 Simplified Diagram of Experimental Apparatus

Also from the solution to the free vibration equation, the following relation involving the damping ratio is found:

$$2\pi\zeta/\sqrt{1-\zeta^2} = (1/n)*\ln(Xo/Xn) \qquad (6.3)$$

where n = number of elapsed oscillatiions,
Xo = original oscillation amplitude,
Xn = amplitude after n cycles.

Again using average experimental values, the damping ratio was determined from equation 6.3 to be:

$$\zeta = 0.0325$$

Using the experimentally obtained damping ratio and damped frequency, the natural frequency, Wn, was determined from equation 6.2 to be:

$$Wn = 67.5799 \text{ rad/sec}$$

The natural frequency of the system is also given by:

$$Wn = K/J. \qquad (6.4)$$

For this system the effective spring constant, K, is:

$$K = 2*ks*R^2 \tag{6.5}$$

where ks = individual spring constant,
R = distance from gas generator axis to spring attachment point.

The individual spring constants were experimentally determined to be ks = 6.891 lb./in., with R = 7.0 inches. Solving for the effective spring constant ,K, using equation 6.5 the total system inertia, J, was calculated from equation 6.4 to be:

$$J = 0.14786 \text{ in.lb.s}^2$$

The total inertia is equal to the sum of the individual inertia effects of the gas generator rotor (JG), lever arm (JL), and springs (JS):

$$J = JG + JL + JS. \tag{6.6}$$

The lever arm inertia was calculated as:

$$JL = ml*l^2/12 = 0.014457 \text{ in.lb.s}^2 \tag{6.7}$$

where ml = mass of the lever arm,
l = length of the lever arm.

The combined inertia of both springs was calculated using Rayleigh's method to be:

$$JS = 2*ms*R^2/3 = 0.01914 \text{ in.lb.s}^2 \tag{6.8}$$

where ms = mass of each spring.

With the total (J), spring (JS), and lever (JL) inertias determined, the gas generator inertia was found using equation 6.6 to be:

$$JG = 0.1143 \text{ in. lb. s}^2$$

B. NONLINEAR DYNAMIC PROGRAM

The nonlinear dynamic program was formulated using Discrete Simulation Language (DSL). The flowchart describing the dynamic program algorithm is given in Figure 6.3. The user must enter the fuel flowrate and dynamometer water weight as a function of time by editing the program prior to execution. Upon execution of the program the user interactively enters the initial gas generator and dynamometer speeds. The steady state program is then called to determine the equilibrium value of fuel flowrate, Mfo, and dynamometer water weight, Wwo. A time step is then taken and the new fuel flowrate and water weight is determined. The dynamic effect of the fuel energy lag is then computed. The steady state program is again used to determine the compressor, high pressure turbine, free power turbine, and dynamometer torques (QC, QH, QF, QD respectively). These torques are entered into the dynamic equations describing the gas generator and dynamometer accelerations. accelerations are integrated to obtain speeds. A check is made to determine if the run time is exceeded. If not, time is again incremented and the loop repeats. A copy of the nonlinear dynamic program is included as Appendix D.

In order to validate the nonlinear dynamic program the propulsion plant test facility was subjected to step changes in commanded fuel flowrate voltage. The resulting fuel flowrate, gas generator speed, and dynamometer speed was

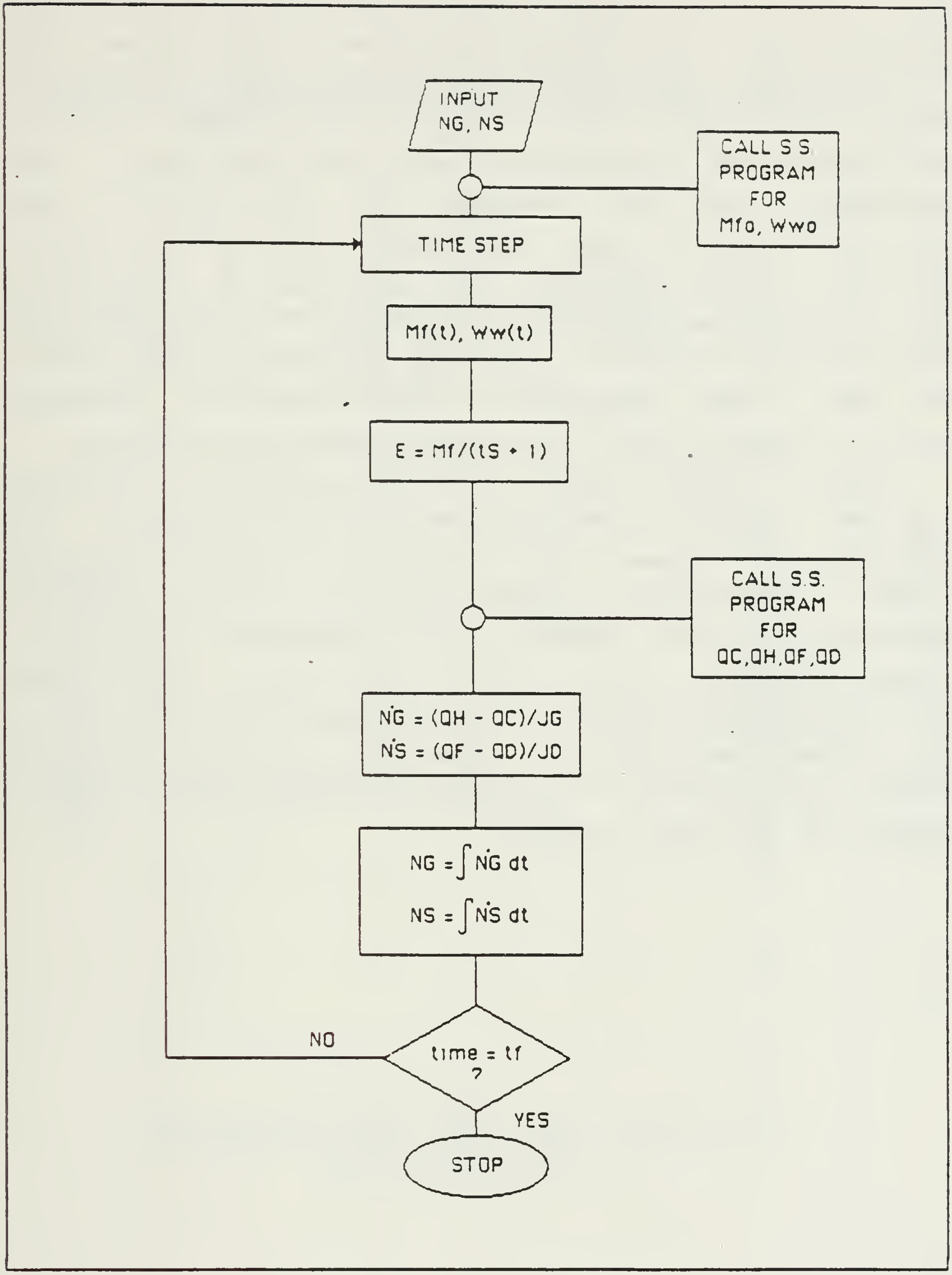

Figure 6.3   Flowchart for Nonlinear Dynamic Program

recorded using a multichannel strip chart. The recorded fuel flowrate versus time was entered into the dynamic program in tabular form and was used to exercise the program. Various acceleration and deceleration tests were conducted. The experimental data were compared with the output of the nonlinear dynamic program. Figures 6.4, 6.5, and 6.6 illustrate the results obtained. In this case an increase in fuel flowrate was applied to accelerate the gas generator from 25,000 to 29,500 rpm and the dynamometer from 960 to 1090 rpm. This represents a large transient, covering nearly one third of the gas generator operating envelope.

Figure 6.4 shows the fuel flowrate transient in response to a step change in commanded fuel flowrate voltage. The discontinuities shown result from the discrete sampling effects of the digital flowmeter, and the unsteady nature of the fuel flow at the location of the flowmeter, just downstream of the fuel control valve. Figures 6.5 and 6.6 show the resulting transients of the gas generator and dynamometer. Experimental data is plotted along with the nonlinear dynamic model results. The results show excellent agreement between the data and the model.

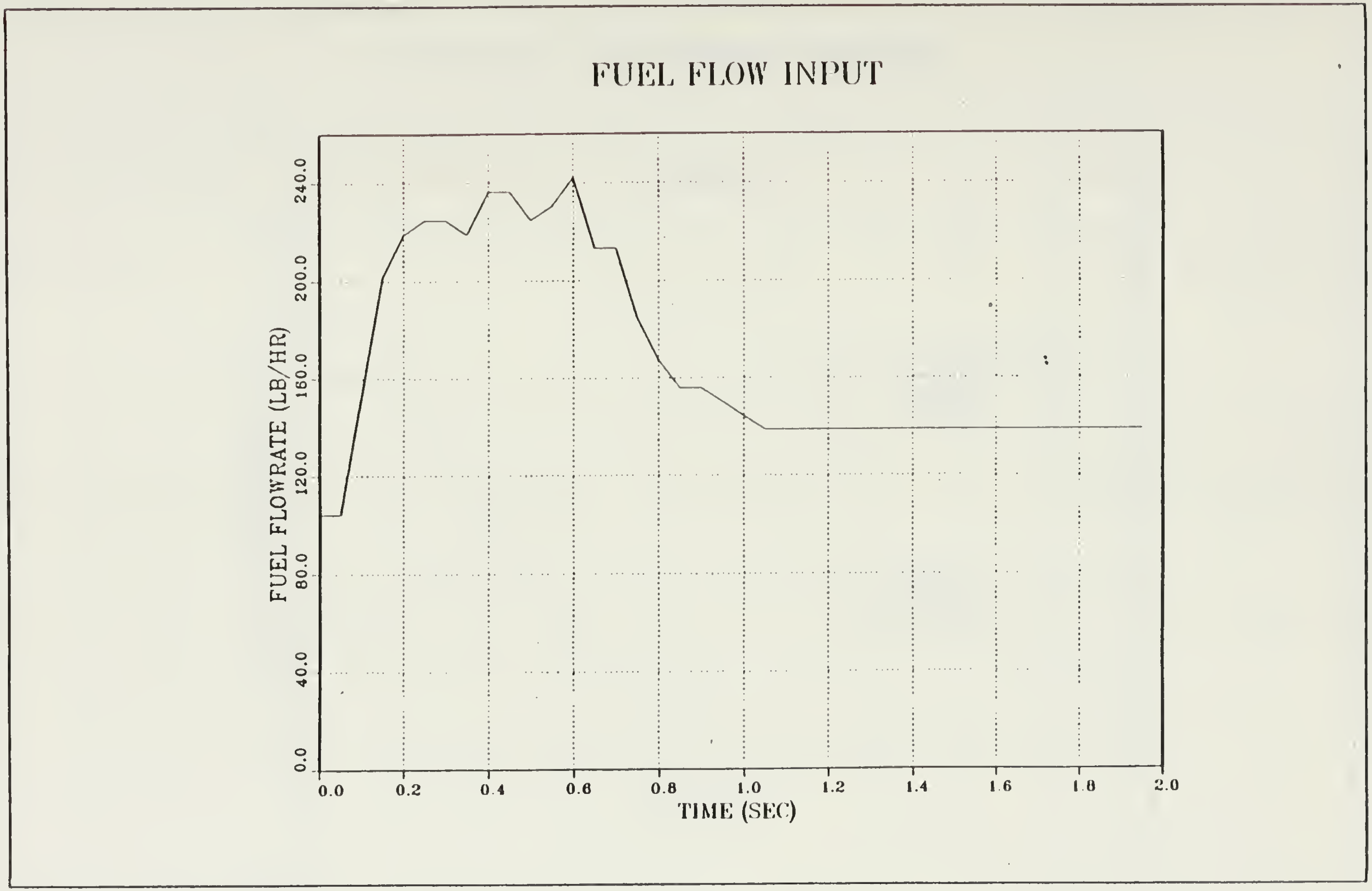

Figure 6.4 Fuel Flowrate Input

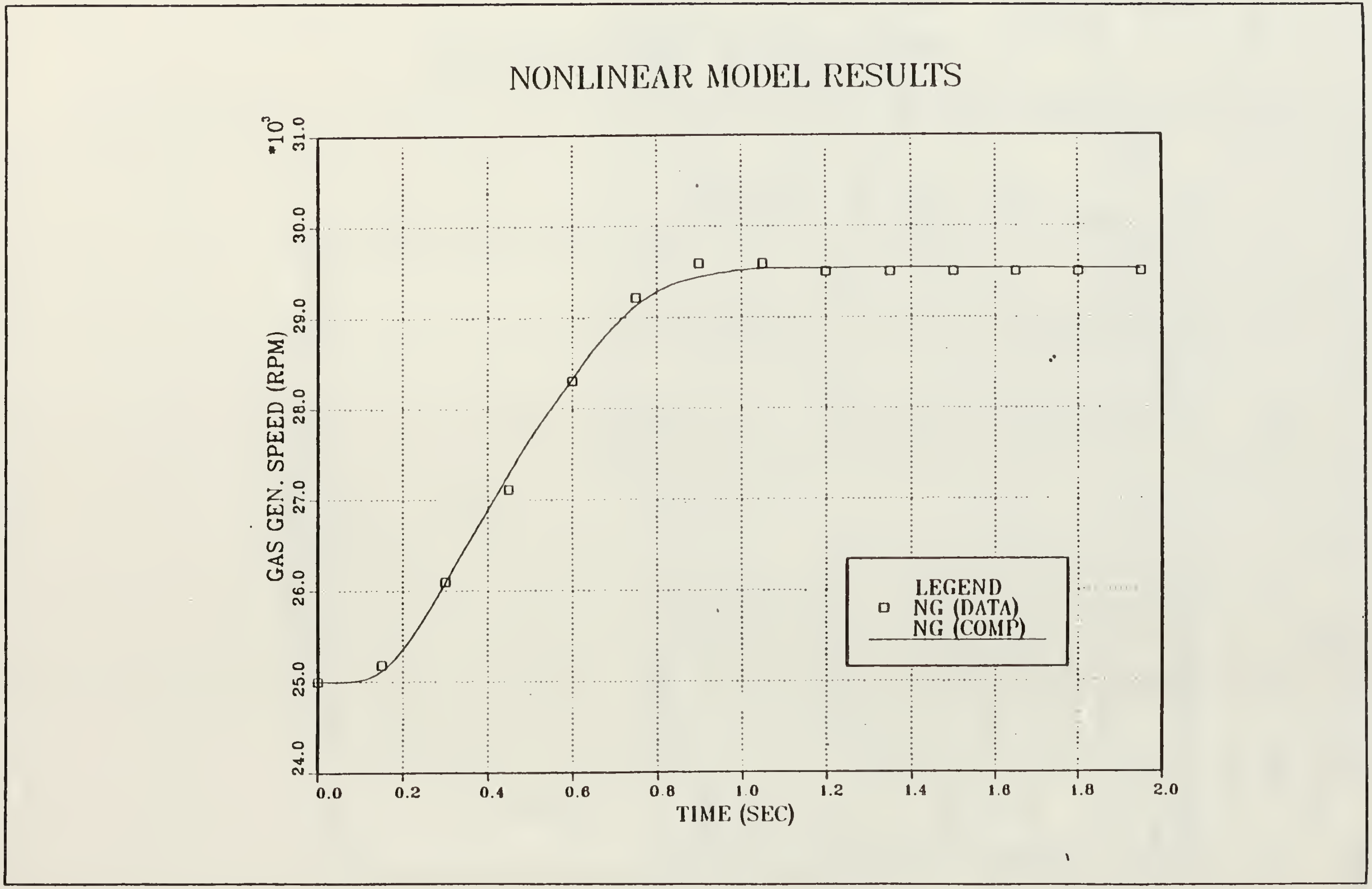

Figure 6.5 Gas Generator Response

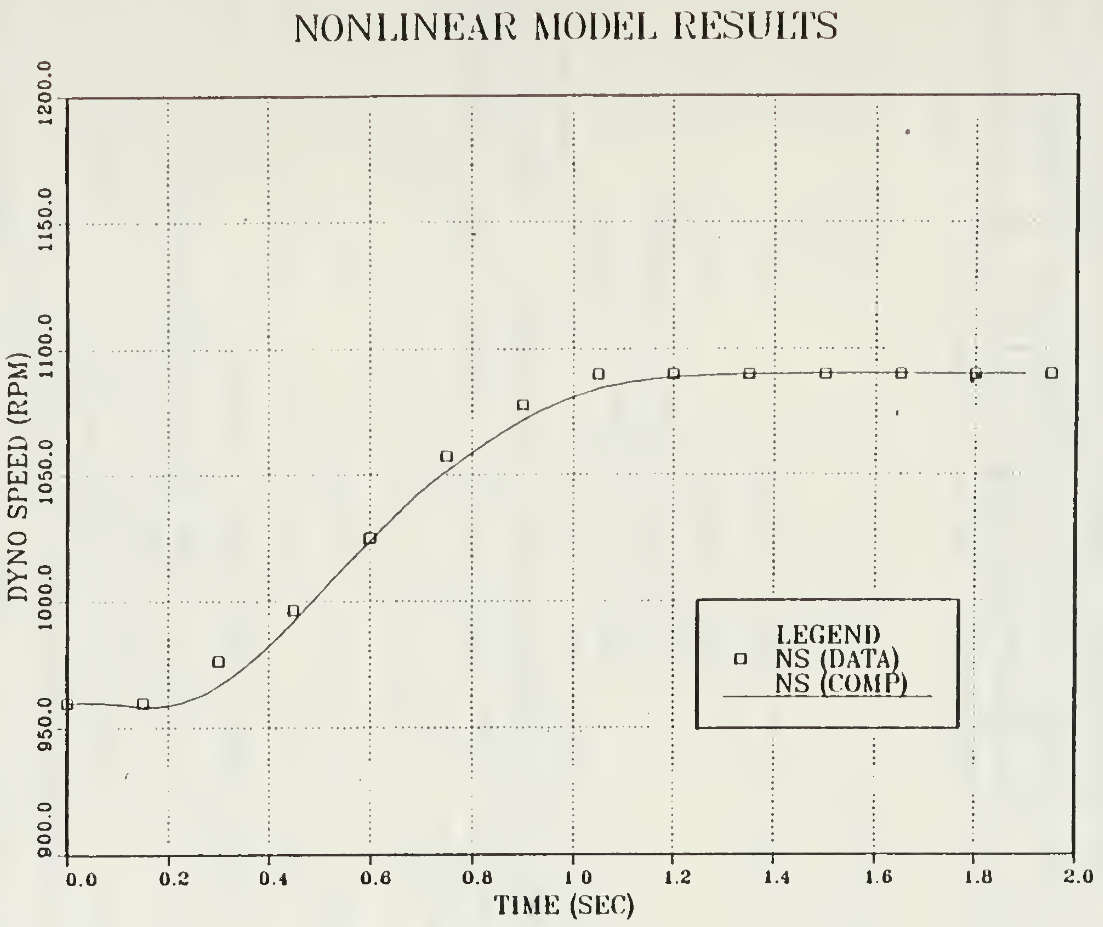

Figure 6.6 Dynamometer Response

VII. STATE SPACE MODEL

Among modern control theory techniques, the linear quadratic regulator (LQR) method is the most highly developed. This method coordinates multiple inputs simultaneously and provides a straightforward manner in which the feedback gain matrix can be manipulated to achieve the desired system performance. The LQR method and calls for the system to be represented in state space form as shown below:

$$\dot{X} = A*X + B*U \tag{7.1}$$

where X = state vector,
U = input vector,
A = state coefficient matrix,
B = input coefficient matrix.

In order to arrive at the state space representation, one must resort to perturbational variables. The excursion of any variable, X, away from its initial condition can be represented by:

$$X = Xo + x \tag{7.2}$$

where Xo = the initial value,
x = dX = the perturbation from the initial value,
X = the current value.

Any variable can be represented in this manner.

In this study the states are the gas generator speed (NG), power turbine/dynamometer speed (NS), and mechanical energy resulting fuel combustion (E). This selection is mandated by the dynamic equations of 3.1, 3.2, and 3.3, one state per derivative term [Ref. 10:p. 665]. The plant inputs are the fuel flowrate (Mf) and the dynamometer water weight (Ww). Using perturbational variables the state space equation becomes:

$$\begin{Bmatrix} \dot{ng} \\ \dot{ns} \\ \dot{e} \end{Bmatrix} = A \begin{Bmatrix} ng \\ ns \\ e \end{Bmatrix} + B \begin{Bmatrix} mf \\ ww \end{Bmatrix} .$$

What remains to be done is to determine the elements of the 'A' and 'B' matrices, which contain the coefficients of the state equation set. We can write these elements symbolically as:

$$\begin{array}{lll} a11 = \partial \dot{ng}/\partial ng & a12 = \partial \dot{ng}/\partial ns & a13 = \partial \dot{ng}/\partial e \\ a21 = \partial \dot{ns}/\partial ng & a22 = \partial \dot{ns}/\partial ns & a23 = \partial \dot{ns}/\partial e \\ a31 = \partial \dot{e}/\partial ng & a32 = \partial \dot{e}/\partial ns & a33 = \partial \dot{e}/\partial e \end{array}$$

$$\begin{array}{ll} b11 = \partial \dot{ng}/\partial mf & b12 = \partial \dot{ng}/\partial ww \\ b21 = \partial \dot{ns}/\partial mf & b22 = \partial \dot{ns}/\partial ww \\ b31 = \partial \dot{e}/\partial mf & b32 = \partial \dot{e}/\partial ww . \end{array}$$

In order to arrive at these coefficients a Taylor series expansion was carried out on each component input/output equation retaining only first order terms. The results is a set of linear equations which can reduced to the state space form given above. A detailed solution for these coefficients is included as Appendix E. It is important to note that these coefficients vary with operating point. A subroutine was added to the steady state program which evaluates these analytic coefficient expressions at user specified operating points (SUBROUTINE PART in Appendix C). Table 5 shows how the 'A' and 'B' matrices vary with operating point.

Comparison between the state space model and nonlinear dynamic model was conducted by subjecting both to the fuel flowrate inputs used in the nonlinear dynamic program validation. Shown in Figure 7.1 and 7.2 is the response of the

TABLE 5

VARIATION OF 'A' AND 'B' MATRICES WITH OPERATING POINT

NG (rpm)	NS (rpm)	'A' matrix			'B' matrix	
21,000	600	-5.800	-0.696	1402.2	0.00	0.00
		0.155	-5.803	7.822	0.00	-166.3
		0.000	0.000	-0.500	0.50	0.00
26,000	1,500	-17.273	-1.649	2479.4	0.00	0.00
		0.317	-3.430	-0.3851	0.00	-708.9
		0.000	0.000	-0.500	0.50	0.00
30,000	2,000	-23.39	-0.950	2485.4	0.00	0.00
		0.5194	-3.706	-5.3616	0.00	-1183.7
		0.000	0.000	-0.500	0.50	0.00

nonlinear and state space models. In this example the 'A' and 'B' matrices were evaluated at the initial condition. As expected for large perturbations such as this, the state space model, with its linear assumptions, does not accurately describe the behavior of this highly nonlinear plant. The limitations of the state space model are important in that they indicate the limits of the LQR controller design and help to define necessary transitions between linear approximations. When an accurate global dynamic model of the plant is needed for control testing, the nonlinear model will be used.

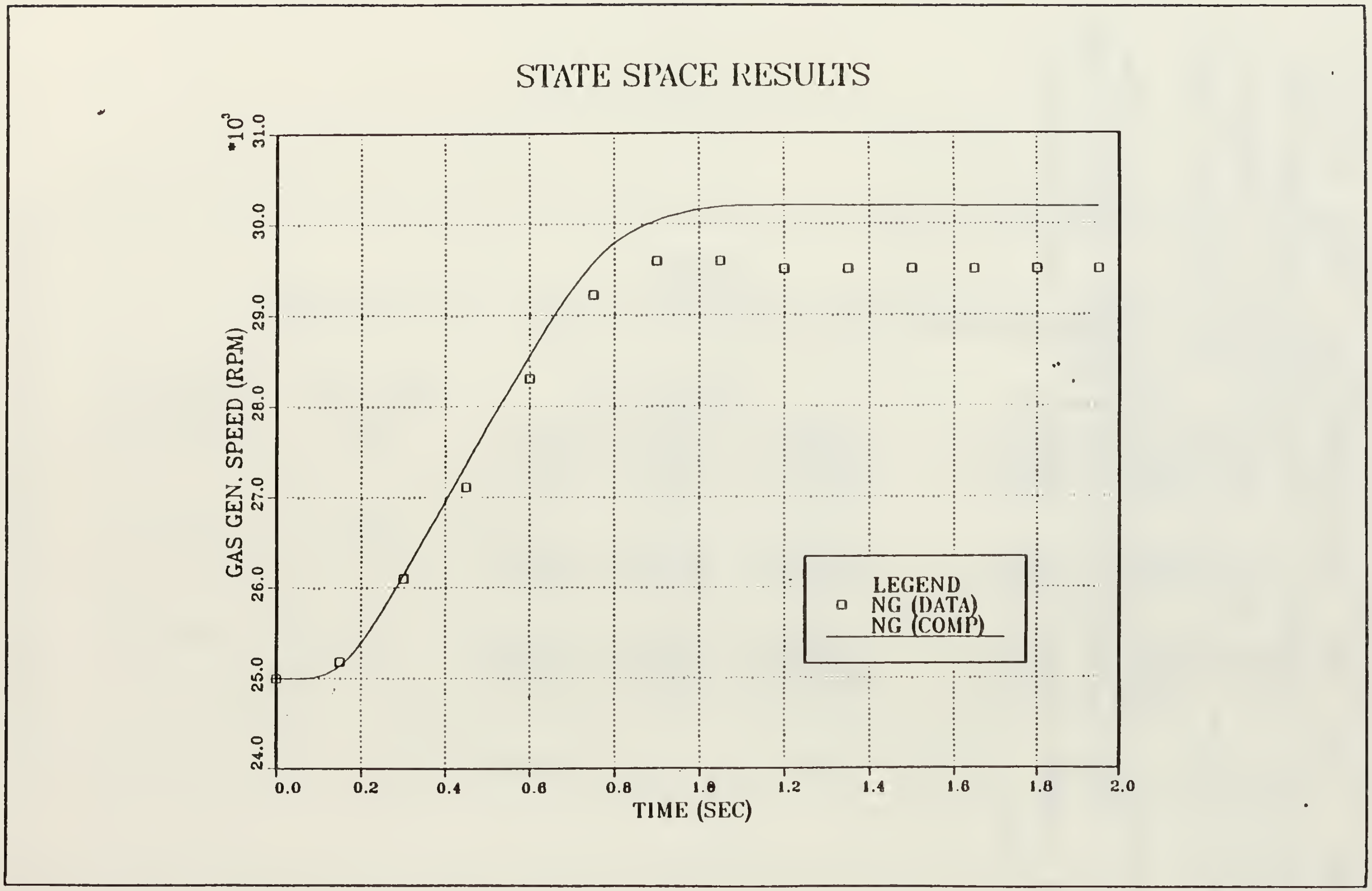

Figure 7.1 Comparison of State Space vs. Nonlinear Model Gas Generator Response

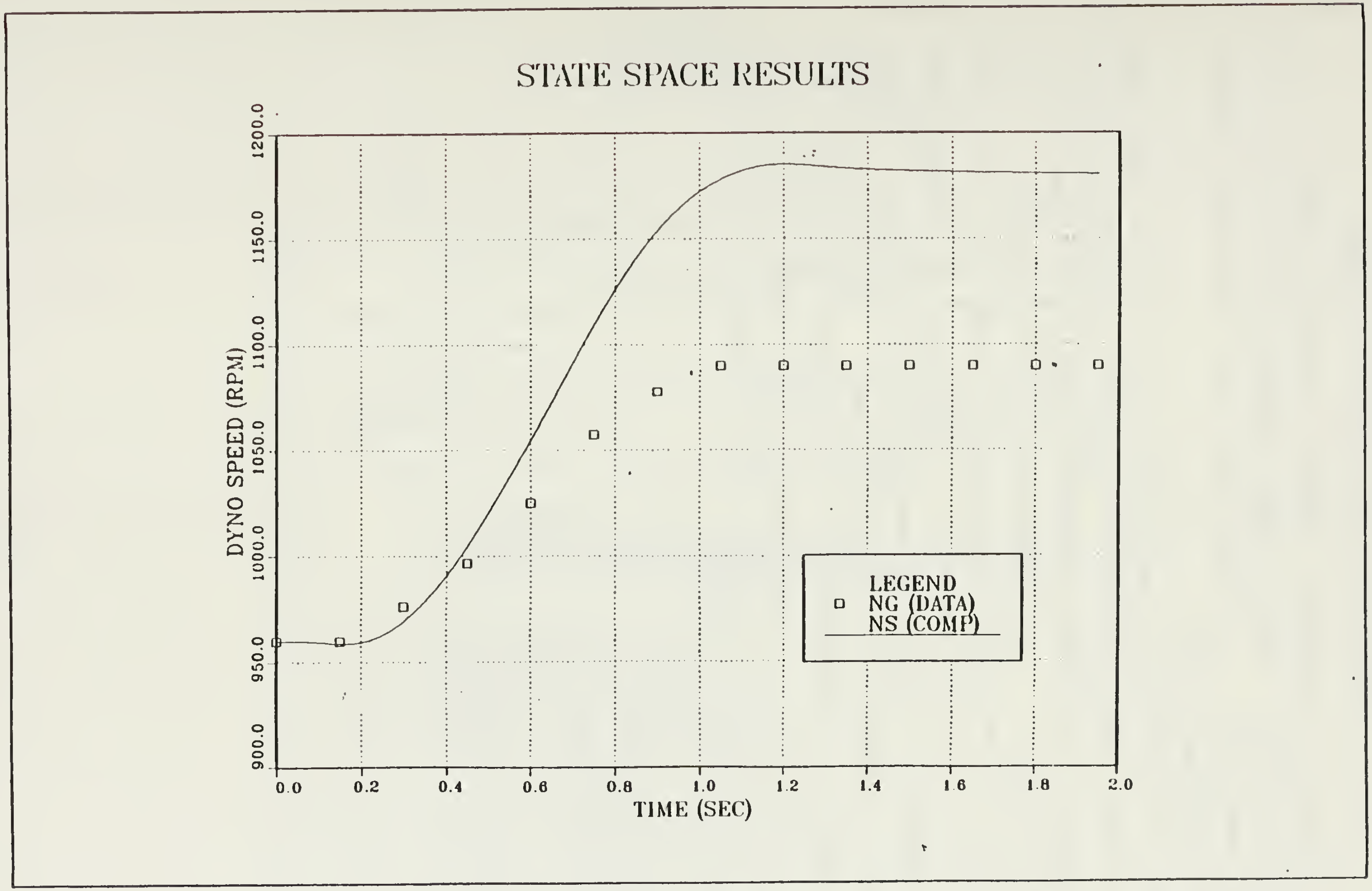

Figure 7.2 Comparison of State Space vs. Nonlinear Model Dynamometer Response

VIII. CONCLUSIONS AND RECOMMENDATIONS

An accurate nonlinear dynamic computer model of the Naval Postgraduate School marine propulsion test facility has been developed. A state space model has been derived from the nonlinear model.

Prior to controller design the fuel flowrate and dynamometer water weight actuators must be accurately modeled and included in both the nonlinear dynamic program and the state space model. In conjunction with this effort, a new fuel control valve should be installed which will allow more direct control of fuel flowrate. When this is accomplished a new controller should be designed using modern control techniques, specifically the LQR method. Performance tests should then be conducted on the turbine using both the present (classical) controller and the modern controller. These tests should include power level transients as well as simulated sea state conditions.

APPENDIX A

DATA ACQUISITION PROGRAM

```
10 ! ##############################################################!
20 ! #                                                            #!
30 ! #        GAS TURBINE DATA AQUISITION/REDUCTION PROGRAM       #!
40 ! #                                                            #!
50 ! ##############################################################!
60 !
70 ! USER INSTRUCTIONS:
80 !
90 ! THIS PROGRAM IS USED TO TAKE GAS TURBINE DATA AT THE
100 ! NAVAL POSTGRADUATE SCHOOL GAS TURBINE MARINE PROPULSION TEST
110 ! FACILITY.
120 !
130 ! THE USER MUST INPUT THE FOLLOWING DATA INTRACTIVELY:
140 !
150 !       (1) TIME ,DATE
160 !       (2) BAROMETERIC PRESSURE
170 !       (1) BAROMETERIC TEMPERATURE CORRECTION
180 !       (4) UPPER AND LOWER FUEL FLOWRATE ROTOMETER READINGS.
190 ! THE FOLLOWING IS A LIST OF VARIABLE NAMES USED IN THIS PROGRAM :
200 !
210 ! A1 = AIR FUEL RATIO.
220 ! D0 = THETA CORRECTION FACTOR
230 ! D1 = DELTA CORRECTION FACTOR
240 ! D2 = DELTA PRESSURE IN THROAT.
250 ! D5 = BAROMETERIC LATITUDE CORRECTION.
260 ! D6 = BAROMETERIC TEMP CORRECTION.
270 ! K(1) = HPT INLET TEMP RIGHT A.
280 ! K(2) = HPT INLET TEMP LEFT A.
290 ! K(3) = HPT INLET TEMP RIGHT A.
300 ! K(4) = HPT INLET TEMP LEFT A.
310 ! K(5) = FPT INLET TEMP RIGHT A.
320 ! K(6) = FPT INLET TEMP LEFT A.
330 ! K(7) = FPT INLET TEMP RIGHT B.
340 ! K(8) = FPT INLET TEMP LEFT B.
350 ! M1 = MASS FLOW RATE LOWER FLOAT
360 ! M2 = MASS FLOW RATE UPPER FLOAT
370 ! M3 = CORRECTED MASS FLOW RATE UPPER FLOAT
```

```
380 ! M4 = CORRECTED MASS FLOW RATE LOWER FLOAT
390 ! M5 = CORRECTED MASS FLOWRATE
400   M6 = MASS FLOW RATE OF AIR.
410 ! M7 = CORRECTED MASS FLOW RATE OF AIR
420 ! M8 = COMBINED AIR AND FUEL FLOW RATE.
430 ! N(1) = COMPRESSOR SPEED, VOLTS
440 ! N(2) = DYNAMOMETER SPEED, VOLTS
450 ! N(4) = DYNAMOMETER  TORQUE, VOLTS
460 ! N1 = COMPRESSOR SPEED
470 ! N2 = DYNAMOMETER SPEED
480 ! N3 = CORRECTED GAS GENERATOR SPEED.
490 ! N4 = CORRECTED DYNAMOMETER SPEED.
500 ! P0 = ATM. PRESSURE
510 ! P1 = AVERAGE INLET BELL PRESS.
520 ! P2 = AVE COMPRESSOR DISCHARGE PRESS.
530 ! P4 = AVE FPT INLET PRESS.
540 ! P8 = AVE CELL PRESS.
550 ! P9 = CORRECTED CELL PRESSURE.
560 ! P(2) = COMPRESSOR DISCHARGE PRESSURE,RIGHT.
570 ! P(4) = FPT INLET PRESSURE.
580 ! P(5) = COMPRESSOR DISCHARGE PRESSURE,LEFT.
590 ! P(9) = INLET BELL PRESSURE, LEFT.
600 ! P(14) = CELL PRESSURE,FRONT.
610 ! P(15) = CELL PRESSURE,REAR.
620 ! P(16) = INLET BELL PRESSURE, RIGHT.
630 ! Q1 = HPT TORQUE.
640 ! Q2 = DYNO TORQUE.
650 ! Q3 = CORRECTED HPT TORQUE.
660 ! Q4 = CORRECTED DYNO TORQUE.
670 ! R1 = LOWER ROTOMETER READING.
680 ! R2 = UPPER ROTOMETER READING.
690 ! R5 = RUN COUNER
700 ! T0 = AVERAGE INLET TEMP.
710 ! T2 = AVE COMPRESSOR DISCHARGE TEMP.
720 ! T3 = AVE HPT INLET TEMP.
730 ! T4 = AVE FPT INLET TEMP
740 ! T(0) = COMPRESSOR INLET TEMP A.
750 ! T(1) = COMPRESSOR INLET TEMP B.
760 ! T(2) = COMPRESSOR INLET TEMP C.
770 ! T(3) = COMPRESSOR INLET TEMP D.
780 ! T(4) = COMPRESSOR DISCHARGE TEMP ,RIGHT A.
790 ! T(5) = COMPRESSOR DISCHARGE TEMP ,LEFT A.
800 ! T(6) = COMPRESSOR DISCHARGE TEMP ,RIGHT B.
810 ! T(7) = COMPRESSOR DISCHARGE TEMP ,LEFT B.
820 !
830 OPTION BASE 0
840 DIM B$[10],C$[11],P(16),T(10),K(10),N(10),K1$[30]
850 !
```

```
860 !   DECLARE TYPEWRITER PRINTER AS OUTPUT DEVICE.
870 !
880 PRINTER IS 10,120
890 !
900 ! ZERO RUN COUNTER
910 !
920 R5=0
930 !
940 CLEAR
950 DISP "ENTER TIME,DATE:";
960 INPUT K1$
970 DISP " "
980 DISP "ENTER BAROMETRIC PRESS (IN.HG.)";
990 INPUT PO
1000 ! DISP " "
1010 ! DISP "FOR NPS THE LAT. CORRECTION IS:  -0.025 (IN.HG.)";
1020 D5=-.025
1030 DISP " "
1040 DISP "ENTER TEMP. CORRECTION (IN.HG.)";
1050 INPUT D6
1060 !
1070 ! ATMOSPHERIC PRESSURE CORRECTED FOR LATITUDE AND TEMP (IN.HG) , PO.
1080 !
1090 PO=PO+D5+D6
1100 CLEAR
1110 R5=R5+1
1120 DISP " "
1130 DISP "ENTER UPPER ROTOMETER READING";
1140 INPUT R2
1150 DISP " "
1160 DISP "ENTER LOWER ROTOMETER READING";
1170 INPUT R1
1180 !
1190 SETTIME 0,0
1200 !
1210 ! FORMAT THE OUTPUT STATEMENTS
1220 !
1230 IMAGE K,5D.2D,K
1240 IMAGE K,4D.2D,K
1250 IMAGE K,1D.4D,K
1260 IMAGE K,5D.D,K
1270 !
1280 ! ZERO INPUT ARRAYS .
1290 !
1300 FOR I=0 TO 16
1310 P(I)=0
1320 NEXT I
1330 FOR I=0 TO 10
```

```
1340 T(I)=0
1350 K(I)=0
1360 N(I)=0
1370 NEXT I
1380 !
1390 ! THE FOLLOWING LOOP CAUSES 'K0' READINGS TO BE TAKEN
1400 ! FOR EACH PRESSURE DATA POINT.
1410 ! THIS ENSURES THAT A GOOD AVERAGE VALUE IS OBTAINED.
1420 !
1430 K0=8
1440 K1=30
1450 FOR K5=1 TO K0
1460 !
1470 !
1480 ! TAKE PRESSURE READINGS.
1490 !
1500 ! NOTE: CHANNELS 1-8 ARE CALIBRATED IN IN.HG.
1510 !
1520 SET TIMEOUT 7;1000
1530 OUTPUT 705 "SCO 1-8 ",CHR$(13)
1540 FOR I=1 TO 8
1550 ENTER 705 USING "%,K" ; B$
1560 P(I)=P(I)+VAL(B$[3])
1570 NEXT I
1580 ABORTIO 7
1590 !
1600 ! NOTE: CHANNELS 1-8 ARE CALIBRATED IN IN.H2O.
1610 !
1620 SET TIMEOUT 7;1000
1630 OUTPUT 705 "SCO 9-16 ",CHR$(13)
1640 FOR J=9 TO 16
1650 ENTER 705 USING "%,K" ; C$
1660 P(J)=P(J)+VAL(C$[4])
1670 NEXT J
1680 ABORTIO 7
1690 !
1700 NEXT K5
1710 !
1720 !
1730 ! TAKE THERMOCOUPLE VOLTS.
1740 !
1750 ! TYPE 'T' THERMOCOUPLES.
1760 !
1770 FOR I=40 TO 49
1780 OUTPUT 709 ;"AC",I
1790 FOR J=1 TO K1
1800 ENTER 709 ; V
1810 T(I-40)=V+T(I-40)
```

```
1820 ! PRINT "V(";I-40;")=";V
1830 NEXT J
1840 NEXT I
1850 !
1860 ! TYPE 'K' THERMOCOUPLES.
1870 !
1880 FOR I=50 TO 59
1890 OUTPUT 709 ;"AC",I
1900 FOR J=1 TO K1
1910 ENTER 709 ; V
1920 K(I-49)=K(I-49)+V
1930 ! PRINT "V(";I-49;")=";V
1940 ! PRINT "K(";I-49;")=";K(I-49)
1950 NEXT J
1960 NEXT I
1970 !
1980 !
1990   TAKE SPEED,TORQUE VOLTS.
2000 !
2010 FOR I=20 TO 29
2020 OUTPUT 709 ;"AC",I
2030 FOR J=1 TO K1
2040 ENTER 709 ; V
2050 N(I-19)=N(I-19)+V
2060 NEXT J
2070 NEXT I
2080 !
2090 !
2100 ! DATA COLLECTION COMPLETE. NOW COMPUTE AVERAGE VALUE OF EACH DATA POINT.
2110 !
2120 ! DATA POINT.
2130 BEEP
2140 BEEP
2150 BEEP
2160 !
2170 DISP "ENTER CELL PRESSURE  (REMINDER, NEG NUMBER!) ";
2180 INPUT P8
2190 !
2200 !
2210 FOR I=0 TO 16
2220 P(I)=P(I)/K0
2230 NEXT I
2240 !
2250 ! CONVERT FROM VOLTS TO DEG.F FOR TYPE 'T' THERMOCOUPLE.
2260 !
2270 FOR I=0 TO 9
2280 T(I)=T(I)/K1
2290 M0=.1008609
```

```
2300 M1=25727.94369
2310 M2=-767345.8295
2320 M3=78025595.81
2330 M4=-9247486589
2340 M5=697688000000
2350 M6=-2.66192E13
2360 M7=3.94078E14
2370 T(I)=M0+T(I)*(M1+T(I)*(M2+T(I)*(M3+T(I)*(M4+T(I)*(M5+T(I)*(M6+T(I)*M7))))))
2380 T(I)=9/5*T(I)+32
2390 NEXT I
2400 !
2410 ! CONVERT FROM VOLTS TO DEG.F FOR TYPE 'K' THERMOCOUPLE.
2420 !
2430 FOR I=1 TO 10
2440 K(I)=K(I)/K1
2450 !
2460 L0=.226584602
2470 L1=24152.109
2480 L2=67233.4248
2490 L3=2210340.682
2500 L4=-860963914.9
2510 L5=48350600000
2520 L6=-1.18452E12
2530 L7=1.3869E13
2540 L8=-6.33708E13
2550 K(I)=L0+K(I)*(L1+K(I)*(L2+K(I)*(L3+K(I)*(L4+K(I)*(L5+K(I)*(L6+K(I)*(L7+K(I)*L8)))))))
2560 K(I)=9/5*K(I)+32
2570 NEXT I
2580 !
2590 ! CONVERT SPEED,TORQUE READINGS.
2600 !
2610 FOR I=1 TO 10
2620 N(I)=N(I)/K1
2630 NEXT I
2640 !
2650 ! DYNO SPEED (RPM),N2.
2660 !
2670 N2=N(1)*652.728191
2680 !
2690 ! GAS GENERATOR SPEED (RPM),N1.
2700 !
2710 N1=16143.31+N(2)*3778
2720 !
2730 ! DYNO TORQUE (FT-LBS.),Q2.
2740 !
2750 Q2=N(4)*59.254138
2760 !
2770 ! COMPUTE AVERAGE VALUE FOR REDUNDANT DATA POINTS.
```

```
2780 !
2790 !
2800 ! INLET BELL PRESSURE (IN.H2O),P1
2810 !
2820 P1=(P(9)+P(16))/2
2830 !
2840 ! COMPRESSOR DISCHARGE PRESSURE (IN.HG.),P2.
2850 !
2860 P2=(P(2)+P(5))/2
2870 !
2880 ! FPT INLET PRESSURE (IN.HG.),P4.
2890 !
2900 P4=P(4)
2910 !
2920 ! COMPRESSOR INLET TEMPERATURE (DEG.F),TO.
2930 !
2940 TO=(T(0)+T(1)+T(2)+T(3))/4
2950 !
2960 ! COMPRESSOR DISCHARGE TEMPERATURE (DEG.F),T2.
2970 !
2980 T2=(T(4)+T(5)+T(6)+T(7))/4
2990 !
3000 ! HPT INLET TEMPERATURE (DEG.F),T3.
3010 !
3020 T3=(K(1)+K(2)+K(3)+K(4))/4
3030 !
3040 ! FPT INLET TEMPERATURE (DEG.F),T3.
3050 !
3060 T4=(K(5)+K(6)+K(7)+K(8))/4
3070 !
3080 PRINT "DATE: ";K1$;"     RUN: ";R5
3090 PRINT
3100 !
3110 CLEAR
3120 DISP "DETAILED REPORT ENTER'1'."
3130 DISP "SUMMARY REPORT ENTER '2'.";
3140 DISP "ANALYSIS ENTER '3'.";
3150 INPUT Z1
3160 IF Z1=3 THEN GOTO 3950
3170 !
3180 PRINT "*****************************************************************"
3190 PRINT "                          RAW DATA                              "
```

```
3170 !
3180 PRINT "****************************************************************"
3190 PRINT "                           RAW DATA                             "
3200 PRINT "****************************************************************"
3210 PRINT
3220 PRINT
3230 IF Z1=2 THEN GOTO 3710
3240 PRINT
3250 PRINT "PRESSURES"
3260 PRINT " "
3270 PRINT USING 1230 ; "BAROMETERIC PRESS.,                 PO     = ";PO;" IN.HG."
3280 PRINT USING 1230 ; "CELL PRESS. FRONT,                  P(13) = ",P(13)," IN.H2O"
3290 PRINT USING 1230 ; "CELL PRESS. REAR,                   P(14) = ",P(14)," IN.H2O"
3300 PRINT USING 1230 ; "CELL PRESS. AVE,                    P8     = ",P8," IN.H2O"
3310 PRINT USING 1230 ; "INLET BELL PRESS. RIGHT,            P(16) = ",P(16)," IN.H2O"
3320 PRINT USING 1230 ; "INLET BELL PRESS. LEFT,             P(9)   = ",P(9)," IN.H2O"
3330 PRINT USING 1230 ; "INLET BELL PRESS. AVE,              P1     = ",P1," IN.H2O"
3340 PRINT USING 1230 ; "COMPRESSOR DISCH. PRESS. RIGHT,     P(2)   = ",P(2)," IN.HG"
3350 PRINT USING 1230 ; "COMPRESSOR DISCH. PRESS. LEFT,      P(5)   = ",P(5)," IN.HG"
3360 PRINT USING 1230 ; "COMPRESSOR DISCH. PRESS. AVE,       P2     = ",P2," IN.HG"
3370 PRINT USING 1230 ; "FPT INLET PRESS.,                   P4     = ",P4," IN.HG"
3380 PRINT " "
3390 PRINT "TEMPERATURE"
3400 PRINT " "
3410 PRINT USING 1240 ; "COMPRESSOR INLET TEMP. A,           T(0)   =  ",T(0)," DEG. F"
3420 PRINT USING 1240 ; "COMPRESSOR INLET TEMP. B,           T(1)   =  ",T(1)," DEG. F"
3430 PRINT USING 1240 ; "COMPRESSOR INLET TEMP. C,           T(2)   =  ",T(2)," DEG. F"
3440 PRINT USING 1240 ; "COMPRESSOR INLET TEMP. D,           T(3)   =  ",T(3)," DEG. F"
3450 PRINT USING 1240 ; "COMPRESSOR INLET TEMP. AVE,         TO     =  ",TO," DEG. F"
3460 PRINT USING 1240 ; "COMPRESSOR DISCH. TEMP. RIGHT A,    T(4)   =  ",T(4)," DEG. F"
3470 PRINT USING 1240 ; "COMPRESSOR DISCH. TEMP. LEFT A,     T(5)   =  ",T(5)," DEG. F"
3480 PRINT USING 1240 ; "COMPRESSOR DISCH. TEMP. RIGHT B,    T(6)   =  ",T(6)," DEG. F"
3490 PRINT USING 1240 ; "COMPRESSOR DISCH TEMP. LEFT B,      T(7)   =  ",T(7)," DEG. F"
3500 PRINT USING 1240 ; "COMPRESSOR DISCH. TEMP. AVE,        T2     =  ",T2," DEG. F"
3510 PRINT USING 1240 ; "HPT INLET TEMP. RIGHT A,            K(1)   =  ",K(1)," DEG. F"
3520 PRINT USING 1240 ; "HPT INLET TEMP. LEFT A,             K(2)   =  ",K(2)," DEG. F"
3530 PRINT USING 1240 ; "HPT INLET TEMP. RIGHT B,            K(3)   =  ",K(3)," DEG. F"
3540 PRINT USING 1240 ; "HPT INLET TEMP. LEFT B,             K(4)   =  ",K(4)," DEG. F"
3550 PRINT USING 1240 ; "HPT INLET TEMP. AVE,                T3     =  ",T3," DEG. F"
3560 PRINT USING 1240 ; "FPT INLET-TEMP. RIGHT A,            K(5)   =  ",K(5)," DEG. F"
3570 PRINT USING 1240 ; "FPT INLET TEMP. LEFT A,             K(6)   =  ",K(6)," DEG. F"
3580 PRINT USING 1240 ; "FPT INLET TEMP. RIGHT B,            K(7)   =  ",K(7)," DEG. F"
3590 PRINT USING 1240 ; "FPT INLET TEMP. LEFT B,             K(8)   =  ",K(8)," DEG. F"
3600 PRINT USING 1240 ; "FPT INLET TEMP. AVE,                T4     =  ",T4," DEG. F"
3610 PRINT " "
3620 PRINT "SPEEDS ETC."
3630 PRINT " "
3640 PRINT USING 1240 ; "UPPER ROTOMETER,                    R2     =  ",R2
```

```
3650 PRINT USING 1240 ; "LOWER ROTOMETER,                   R1    =  ",R1
3660 PRINT USING 1260 ; "GAS GENERATOR SPEED,               N1    =  ",N1," RPM"
3670 PRINT USING 1260 ; "DYNAMOMETER SPEED,                 N2    =  ",N2," RPM"
3680 PRINT USING 1260 ; "DYNAMOMETER TORQUE,                Q2    =  ",Q2," FT-LB"
3690 !
3700 GOTO 3900
3710 !
3720 !
3730 PRINT USING 1260 ; "GAS GENERATOR SPEED,               N1    =  ",N1," RPM"
3740 PRINT USING 1260 ; "DYNAMOMETER SPEED,                 N2    =  ",N2," RPM"
3750 PRINT " "
3760 PRINT USING 1230 ; "BAROMETERIC PRESS.,                P0    = ";P0;" IN.HG."
3770 PRINT USING 1230 ; "CELL PRESS. AVE,                   P8    = ",P8," IN.H2O"
3780 PRINT USING 1230 ; "INLET BELL PRESS. AVE,             P1    = ",P1," IN.H2O"
3790 PRINT USING 1230 ; "COMPRESSOR DISCH. PRESS. AVE,      P2    = ",P2," IN.HG"
3800 PRINT USING 1230 ; "FPT INLET PRESS.,                  P4    = ",P4," IN.HG"
3810 PRINT USING 1240 ; "COMPRESSOR INLET TEMP. AVE,        T0    =  ",T0," DEG. F"
3820 PRINT USING 1240 ; "COMPRESSOR DISCH. TEMP. AVE,       T2    =  ",T2," DEG. F"
3830 PRINT USING 1240 ; "HPT INLET TEMP. AVE,               T3    =  ",T3," DEG. F"
3840 PRINT USING 1240 ; "FPT INLET TEMP. AVE,               T4    =  ",T4," DEG. F"
3850 PRINT USING 1240 ; "UPPER ROTOMETER,                   R2    =  ",R2
3860 PRINT USING 1240 ; "LOWER ROTOMETER,                   R1    =  ",R1
3870 PRINT USING 1260 ; "DYNAMOMETER TORQUE,                Q2    =  ",Q2," FT-LB"
3880 !
3890 !
3900 !
3910 ! DATA REDUCTION CALCULATIONS
3920 !
3930 ! ABS. COMPRESSOR INLET TEMP.(DEG.R)
3940 !
3950 T0=T0+460
3960 !
3970 ! ABS. COMPRESSOR DISCH. TEMP.(DEG.R)
3980 !
3990 T2=T2+460
4000 !
4010 ! ABS. HPT INLET TEMP.(DEG.R)
4020 !
4030 T3=T3+460
4040 !
4050 ! ABS. FPT INLET TEMP.(DEG.R)
4060 !
4070 T4=T4+460
4080 !
4090 ! TEMP. CORRECTION FACTOR
4100 !
4110 D0=T0/520
4120 !
```

```
4130 ! ABS.CELL PRESS.(PSIA)
4140 !
4150 P9=P0*(14.696/29.92)+P8*(14.696/406.92)
4160 !
4170 ! PRESSURE CORRECTION FACTOR.
4180 !
4190 D1=P9/14.696
4200 !
4210 ! CORRECTED DYNO TORQUE.(FT.LBS)
4220 !
4230 Q4=Q2/D1
4240 !
4250 ! CORRECTED GAS GENERATOR SPEED.(RPM)
4260 !
4270 N3=N1/SQR(D0)
4280 !
4290 ! CORRECTED DYNAMOMETER SPEED.(RPM)
4300 !
4310 N4=N2/SQR(D0)
4320 !
4330 ! CALCULATE MASS FLOW RATE OF FUEL.(LB/HR)
4340 !
4350 !
4360 M2=.607954*R2-4.627907
4370 M1=1.17442*R1-8.556818
4380 M5=(M1+M2)/2
4390 !
4400 ! CALCULATE MASS FLOW RATE OF AIR
4410 !
4420 D2=(P8-P1)*14.696/406.92
4430 M6=8.02*.98*P9*21.73/SQR(53.34*T0)*SQR(D2/P9-1.5/1.4*(D2/P9)^2)
4440 M7=3600*M6
4450 !
4460 ! CALCULATE AIR/FUEL RATIO
4470 !
4480 A1=M7/M5
4490 !
4500 ! CALCULATE THE COMBINED AIR + FUEL FLOW RATE (LB/HR).
4510 !
4520 M8=M5+M7
4530 !
4540 ! CALCULATE THE PERCENT THEORETIC AIR.
4550 !
4560 W=A1*6.7847
4570 !
4580 ! CALCULATE THE CORRECTED HPT TORQUE (FT-LBS.).
4590 !
4600 BQ=5.94
```

```
4610 B1=.001951
4620 B2=-.000000281
4630 !
4640 C0=-.0003657
4650 C1=-.0000004536
4660 C2=3.571E-11
4670 !
4680 X=T3-T4
4690 Y=T3*T3-T4*T4
4700 Z=T3^3-T4^3
4710 !
4720 B3=B0*X+B1*Y/2+B2*Z/3
4730 C3=C0*X+C1*Y/2+C2*Z/3
4740 H1=(B3+W*C3)/28.954
4750 Q1=M8*H1*778/(2*3.14159*N1*60)
4760 Q3=Q1/D1
4770 !
4780 ! ABS. COMPRESSOR  DISCH. PRESS. (PSIA)
4790 !
4800 P2=P0*(14.696/29.92)+P2*(14.696/29.92)
4810 !
4820 ! ABS. FPT INLET PRESS. (PSIA)
4830 P4=P0*(14.696/29.92)+P4*(14.696/29.92)
4840 !
4850 !
4860 ! CORRECTED VALUES.
4870 !
4880 T2=T2/D0
4890 T4=T4/D0
4900 M5=M5/(D1*SQR(D0))
4910 M7=M7*SQR(D0)/D1
4920 A1=M7/M5
4930 M8=M5+M7
4940 !
4950 P2=P2/D1
4960 P4=P4/D1
4970 !
4980 !
4990 PRINT " "
5000 PRINT "****************************************************************"
5010 PRINT "                 ANALYSIS (CORRECTED VALUES)                    "
5020 PRINT "****************************************************************"
5030 PRINT " "
5040 !
5050 PRINT USING 1240 ; "COMPRESSOR DISCH. TEMP.,           T2    =  ",T2," DEG. R."
5060 PRINT USING 1240 ; "FPT INLET TEMP.,                   T4    =  ",T4," DEG. R."
5070 PRINT USING 1230 ; "COMPRESSOR DISCH. PRESS.,          P2    = ",P2," PSIA"
5080 PRINT USING 1230 ; "FPT INLET PRESSURE,                P4    = ",P4," PSIA"
```

```
5090 PRINT USING 1240 ; "CORRECTED DYNO TORQUE,            Q4     =  ",Q4," FT-LB"
5100 PRINT USING 1240 ; "CORRECTED HPT TORQUE,             Q3     =  ",Q3," FT-LB"
5110 PRINT USING 1260 ; "CORRECTED GAS GENERAOR SPEED,     N3     =  ",N3," RPM"
5120 PRINT USING 1260 ; "CORRECTED DYNAMOMETER SPEED,      N4     =  ",N4," RPM"
5130 PRINT USING 1230 ; "CORRECTED FUEL FLOW RATE,         M5     = ",M5," LBM/HR"
5140 PRINT USING 1230 ; "CORRECTED AIR FLOW RATE,          M7     = ",M7," LBM/HR"
5150 PRINT USING 1230 ; "COMBINED AIR+FUEL FLOW RATE,      M8     = ",M8," LBM/HR"
5160 PRINT "AIR FUEL RATIO,                A1     = ";A1
5170 PRINT "THETA,                         DO     = ";DO
5180 PRINT "DELTA,                         D1     = ";D1
5190 !
5200 FOR L=1 TO 8
5210 PRINT " "
5220 NEXT L
5230 PRINT N3;P2;M7;T2;Q3;M5;P4;M8;T4;N4;Q4
5240 CLEAR
5250 DISP "ANOTHER RUN ENTER '1'."
5260 DISP "STOP ENTER '2'.";
5270 INPUT Z2
5280 IF Z2<1.5 THEN GOTO 1110
5290 END
```

APPENDIX B

LEAST SQUARES CURVE FIT PROGRAM

```
C
C
C   ******************************************************************
C   ******************************************************************
C   **                                                              **
C   **                LEAST SQUARES CURVE FIT PROGRAM               **
C   **                                                              **
C   **                                                              **
C   ******************************************************************
C   ******************************************************************
C
C
C        USER INSTRUCTIONS:
C
C
C                                                                      FIT00010
C   THIS PROGRAM OBTAINS A PARABOLIC EQUATION TO FIT THE INDEPENDENT   FIT00020
C   VARIABLES (X,Y,Z) TO THE DEPENDENT VARIABLE(B). THE METHOD OF      FIT00030
C   LEAST SQUARES IS USED. THE FORM OF THE PARABOLIC EQUATION IS :     FIT00040
C                                                                      FIT00050
C   C1(XX) + C2(XY) + C3(XZ) + C4(YY) + C5(YZ) + C6(ZZ)  ....          FIT00060
C             + C7(X) + C8(Y) + C9(Z) + C10  =  B                      FIT00070
C                                                                      FIT00080
C   THE OUTPUT OF THIS PROGRAM IS 'C', THE VECTOR OF COEFFICIENTS.     FIT00090
C        !!!! LOOK CLOSELY AT THE EQUATION FORMAT SO YOU               FIT00100
C                USE THE COEFFICIENTS CORRECTLY !!!!!!                 FIT00110
C                                                                      FIT00120
C   BEFORE EXECUTING, FILE 02 MUST BE DEFINED AND HAVE THE DATA .      FIT00130
C   THIS IS ACCOMPLISHED BY TYPING :
C           FI 2 DISK FILENAME DATA (PERM)
C   WHERE 'FILENAME' IS THE NAME OF YOUR DATA FILE WHICH HAS
C   FILETYPE 'DATA'.
C
C   CHECK THE READ STATEMENTS TO ENSURE YOU ARE GETTING THE RIGHT      FIT00140
C   DATA IN THE RIGHT PLACES.                                          FIT00150
C   THE INPUT DATA FORMAT  SHOULD BE :                                 FIT00160
C                                                                      FIT00170
C                      X1  Y1  Z1 ..... B1                             FIT00180
C                      X2  Y2  Z2 ..... B2                             FIT00190
C                      X3  ....                                        FIT00200
C                      .                                               FIT00210
C                      .                                               FIT00220
C                                                                      FIT00230
```

```
C   WHERE X,Y,Z ARE THE INDEPENDENT VARIABLES , B IS THE DEPENDENT       FIT00240
C   VARIABLE, AND SUBSCRIPTS 1,2,3,ETC., INDICATE THE RUN NUMBER.        FIT00250
C                                                                        FIT00260
C                                                                        FIT00270
C   BEFORE EXECUTING , TYPE :                                            FIT00280
C           FORTVS FIT (AUTODBL(DBLPAD)
C   TO RUN THIS PROGRAM IN DOUBLE PRECISION.                             FIT00290
C
C   ALSO BEFORE EXECUTING , TYPE :                                       FIT00280
C                   GLOBAL MACLIB IMSLSP NONIMSL                         FIT00280
C   TO GET ACCESS TO IMSL ROUTINE LEQT2F.                                FIT00290
C
C             GOOD LUCK!!!!!!!
C
C  ########################################################################
C
C                                                                        FIT00300
         DIMENSION X(90,5),BB(90),F(90,21),B(21),                        FIT00310
     1             A(21,21),WKAREA(10000)                                FIT00320
C                                                                        FIT00330
         DATA B/21*0.0/,A/441*0.0/,X/450*0.0/,BB/90*0.0/,F/1890*0.0/,    FIT00340
     1        WKAREA/10000*0.0/                                          FIT00350
C                                                                        FIT00360
         WRITE(6,44)                                                     FIT00370
   44    FORMAT(//,3X,'HOW MANY DATA RUNS ?')                            FIT00380
         READ(5,*) NDATA                                                 FIT00390
         WRITE(6,*) NDATA                                                FIT00400
C                                                                        FIT00410
         WRITE(6,55)                                                     FIT00420
   55    FORMAT(//,3X,'HOW MANY INDEPENDENT VARIABLES ?')                FIT00430
         READ(5,*) NIND                                                  FIT00440
         WRITE(6,*) NIND                                                 FIT00450
C                                                                        FIT00460
         WRITE(6,111)                                                    FIT00470
 111     FORMAT(2X,'SELECT :  1 = COMPLETE QUADRATIC',/                  FIT00480
     1          12X,'2 = REDUCED QUADRATIC',/                            FIT00490
     1          12X,'3 = LINEAR APPROX.')                                FIT00500
C                                                                        FIT00510
         READ(5,*) YY                                                    FIT00520
         WRITE(6,*) YY                                                   FIT00530
C                                                                        FIT00540
         IF(YY-2.0) 112,113,114                                          FIT00550
C                                                                        FIT00560
C                                                                        FIT00570
C                                                                        FIT00580
C     THIS IS FOR THE COMPLETE QUADRATIC                                 FIT00590
C                                                                        FIT00600
 112     NCOEFF = (NIND)*(1 + NIND)/2 + NIND + 1                         FIT00610
         WRITE(6,*) 'NCOEFF=',NCOEFF                                     FIT00620
         GO TO 115                                                       FIT00630
```

```
C
C
C     THIS IS FOR THE REDUCED QUADRATIC
C
 113      NCOEFF = NIND*2 + 1
          WRITE(6,*) 'NCOEFF=',NCOEFF
          GO TO 115
C
C     THIS IS FOR THE LINEAR CURVE FIT
C
 114      NCOEFF = NIND + 1
          WRITE(6,*) 'NCOEFF=',NCOEFF
C
 115      CALL COEFF(X,BB,F,B,A,WKAREA,NCOEFF,NIND,NDATA,YY)
C
          END
C
C         ***************************************************************
          SUBROUTINE COEFF(X,BB,F,B,A,WKAREA,NCOEFF,NIND,NDATA,YY)
C         ***************************************************************
C
          DIMENSION X(NDATA,NIND),BB(NDATA),F(NDATA,NCOEFF),B(NCOEFF),
     1              A(NCOEFF,NCOEFF),WKAREA(10000)
C
C
C   TEST CASE
C
C         X(1,1) = -3.0
C         X(2,1) = -2.0
C         X(3,1) =  0.0
C         X(4,1) =  3.0
C         X(5,1) =  4.0
C
C         BB(1) = 18.0
C         BB(2) = 10.0
C         BB(3) =  2.0
C         BB(4) =  2.0
C         BB(5) =  5.0
C
C
C   READ IN DATA FROM FILE 02
C
          DO 12 I = 1,NDATA
          READ(2,88) X(I,1),BB(I)
          READ(2,89) X(I,2)
  12     CONTINUE
  88      FORMAT(F10.0,40X,F10.0)
  89      FORMAT(20X,F10.0)
C
C
```

```
C   PRINT OUT THE DATA AS A CHECK                                         FIT01140
C                                                                         FIT01150
         WRITE(6,377)                                                     FIT01160
 377     FORMAT(/,2X,'DO YOU WANT TO CHECK THE INPUT ? 1=YES, 2=NO')      FIT01170
         READ(5,*) ZZ                                                     FIT01180
         WRITE(6,*) ZZ                                                    FIT01190
         IF(ZZ.GT.1.5) GO TO  378                                         FIT01200
C                                                                         FIT01210
         DO 201 I = 1,NDATA                                               FIT01220
         DO 191 J = 1,NIND                                                FIT01230
         WRITE(8,211) I,J,X(I,J)                                          FIT01240
  211    FORMAT(3X,'X(',I2,',',I2,') = ',G15.7)                           FIT01250
  191    CONTINUE                                                         FIT01260
         WRITE(8,221) I,BB(I)                                             FIT01270
  221    FORMAT(3X,'BB(',I2,') = ',G15.7)                                 FIT01280
  201    CONTINUE                                                         FIT01290
C                                                                         FIT01300
C                                                                         FIT01310
C   CONSTRUCT THE SYSTEM OF EQUATIONS USING THE METHOD OF                 FIT01320
C   LEAST SQUARES.                                                        FIT01330
C                                                                         FIT01340
  378    DO 100 K = 1,NDATA                                               FIT01350
              L = 0.0                                                     FIT01360
C                                                                         FIT01370
         IF(YY-2.0) 112,113,114                                           FIT01380
C                                                                         FIT01390
C     THIS SET GETS THE COMPLETE QUADRATIC                                FIT01400
C                                                                         FIT01410
  112    DO 110 I = 1,NIND                                                FIT01420
         DO 120 J = I,NIND                                                FIT01430
              L = L + 1                                                   FIT01440
         F(K,L) = X(K,I) * X(K,J)                                         FIT01450
  120    CONTINUE                                                         FIT01460
  110    CONTINUE                                                         FIT01470
         DO 130 II = 1,NIND                                               FIT01480
              L = L + 1                                                   FIT01490
         F(K,L) = X(K,II)                                                 FIT01500
  130    CONTINUE                                                         FIT01510
              L = L + 1                                                   FIT01520
         F(K,L) = 1.0                                                     FIT01530
C                                                                         FIT01540
         GO TO 100                                                        FIT01550
C                                                                         FIT01560
C     THIS GETS THE REDUCED QUADRATIC                                     FIT01570
C                                                                         FIT01580
  113    DO 810 I = 1,NIND                                                FIT01590
              L = L + 1                                                   FIT01600
         F(K,L) = X(K,I) * X(K,I)                                         FIT01610
  810    CONTINUE                                                         FIT01620
         DO 830 II = 1,NIND                                               FIT01630
```

```
            L = L + 1
         F(K,L) = X(K,II)                                                FIT01640
  830    CONTINUE                                                        FIT01650
            L = L + 1                                                    FIT01660
         F(K,L) = 1.0                                                    FIT01670
C                                                                        FIT01680
         GO TO 100                                                       FIT01690
C                                                                        FIT01700
C                                                                        FIT01710
C     THIS GETS THE LINEAR CURVE FIT                                     FIT01720
C                                                                        FIT01730
 114     DO 930 II = 1,NIND                                              FIT01740
            L = L + 1                                                    FIT01750
         F(K,L) = X(K,II)                                                FIT01760
  930    CONTINUE                                                        FIT01770
            L = L + 1                                                    FIT01780
         F(K,L) = 1.0                                                    FIT01790
  100    CONTINUE                                                        FIT01800
C                                                                        FIT01810
         DO 140 M = 1,NCOEFF                                             FIT01820
         DO 150 J = 1,NCOEFF                                             FIT01830
         DO 160 I = 1,NDATA                                              FIT01840
C                                                                        FIT01850
            A(M,J) = (F(I,J) * F(I,M)) + A(M,J)                          FIT01860
C                                                                        FIT01870
  160    CONTINUE                                                        FIT01880
  150    CONTINUE                                                        FIT01890
  140    CONTINUE                                                        FIT01900
         DO 170 J = 1,NCOEFF                                             FIT01910
         DO 180 I = 1,NDATA                                              FIT01920
C                                                                        FIT01930
            B(J) = (F(I,J) * BB(I)) + B(J)                               FIT01940
C                                                                        FIT01950
  180    CONTINUE                                                        FIT01960
  170    CONTINUE                                                        FIT01970
C                                                                        FIT01980
C   PRINT OUT THE 'A' AND 'B' MATRICES AS A CHECK                        FIT01990
C                                                                      F IT02000
C        DO 190 J = 1,NCOEFF                                             FIT02010
C        DO 200 I = 1,NCOEFF                                             FIT02020
C        WRITE(8,210) J,I,A(J,I)                                         FIT02030
C 210    FORMAT(3X,'A(',I2,',',I2,') = ',G15.7)                          FIT02040
C 200    CONTINUE                                                        FIT02050
C        WRITE(8,220) J,B(J)                                             FIT02060
C 220    FORMAT(3X,'B(',I2,') = ',G15.7)                                 FIT02070
C 190    CONTINUE                                                        FIT02080
C                                                                        FIT02090
C                                                                        FIT02100
C                                                                        FIT02110
C                                                                        FIT02120
C                                                                        FIT02130
```

```
C   CALL THE IMSL ROUTINE 'LEQT2F' TO SOLVE THE  LINEAR SYSTEM           FIT02140
C                                                                       FIT02150
C                         A * C = B                                     FIT02160
C                                                                       FIT02170
C   THE SOLUTION,C,IS RETURNED TO THE B VECTOR,AND IS OUTPUT BELOW.     FIT02180
C   'C' IS THE VECTOR OF POLYNOMIAL COEFFICIENTS.                       FIT02190
C                                                                       FIT02200
C                                                                       FIT02210
      NC = 1                                                            FIT02220
      IER = 0                                                           FIT02230
      IDGT = 1                                                          FIT02240
C                                                                       FIT02250
      CALL LEQT2F(A,NC,NCOEFF,NCOEFF,B,IDGT,WKAREA,IER)                 FIT02260
C                                                                       FIT02270
      DO 10 IN = 1,NCOEFF                                               FIT02280
      WRITE(8,11) IN,B(IN)                                              FIT02290
 11   FORMAT(10X,'C(',I2,')=',G15.7)                                    FIT02300
 10   CONTINUE                                                          FIT02310
          END                                                           FIT02320
```

APPENDIX C

STEADY STATE COMPUTER PROGRAM

```
$JOB
C
C
C   ******************************************************************
C   ******************************************************************
C   **                                                              **
C   **                BOEING MODEL 502-6A GAS TURBINE               **
C   **                                                              **
C   **               STEADY STATE COMPUTER SIMULATION               **
C   **                                                              **
C   **                                                              **
C   ******************************************************************
C   ******************************************************************
C
C
C       USER INSTRUCTIONS:
C
C
C   THIS PROGRAM PROVIDES THE INITIALIZATION PROCESS FOR THE DYNAMIC
C   PROGRAM. SPECIFICALLY, THE USER INPUTS GAS GENERATOR AND DYNO
C   SPEEDS, AND THE PROGRAM USES STEADY STATE MAPS (IN EQUATION FORM)
C   OF SYSTEM INPUTS/OUTPUTS TO FIND STEADY STATE VALUES.
C
C   THIS PROGRAM ALSO EVALUATES THE COEFFICIENTS OF THE STATE SPACE
C   MATRICES 'A' AND 'B' AT THE USER SPECIFIED STEADY STATE OPERATING
C   CONDITION.
C
C       THE DEVELOPMENT OF THIS COMPUTER SIMULATION IS DESCRIBED IN:
C
C           HERDA, V.J., 'MARINE GAS TURBINE MODELING
C           FOR MODERN CONTROL DESIGN',(M.S. THESIS,
C           NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA.,
C           JUNE 1986).
C
C           GOOD LUCK!!!!!!!
C
C  #######################################################################
C
C
C      THE FOLLOWING IS A INDEX OF THE PARAMETERS USED IN THIS PROGRAM
C
C           A          STATE SPACE COEFFICIENT MATRIX.
C
```

```
C           B         STATE SPACE COEFFICIENT MATRIX.
C
C           C         VECTOR CONTAINING THE COEFFICIENTS FOR THE
C                     QUADRATIC CURVE FIT.
C
C           MA        AIR FLOW RATE, LBS/HR.
C
C           MAF       COMBINED FUEL AND AIR FLOW RATE, LBS/HR.
C
C           MERR      PERCENTAGE ERROR IN FUEL FLOW RATE
C                     ALLOWED DURING FUEL FLOW RATE CONVERGENCE.
C
C           MF        FUEL FLOW RATE, LBS/HR.
C
C           MFDEL     INCREMENTAL CHANGE IN FUEL FLOW RATE
C                     USED WHEN SEARCHING FOR UPPER AND
C                     LOWER BOUNDS ON FUEL FLOW RATE.
C
C           MFL       LOWER LIMIT ON FUEL FLOW RATE.
C
C           MFMIN     FUEL FLOW RATE WHICH LEADS TO THE MINIMUM
C                     GAS GENERATOR TORQUE DIFFERENTIAL.
C
C           MFU       UPPER LIMIT ON FUEL FLOW RATE.
C
C           MF1       INTERMEDIATE FUEL FLOW RATE USED IN
C                     FUEL FLOW RATE CONVERGENCE.
C
C           MF2       INTERMEDIATE FUEL FLOW RATE USED IN
C                     FUEL FLOW RATE CONVERGENCE.
C
C           NG        GAS GENERATOR SPEED, RPM.
C
C           NS        DYNAMOMETER SPEED, RPM.
C
C           PERR      PERCENTAGE ERROR IN PRESSURES (P2,P4)
C                     ALLOWED DURING PRESSURE CONVERGENCE.
C
C           P2        COMPRESSOR DISCHARGE PRESSURE, PSIA.
C
C           P2G       GUESS FOR THE COMPRESSOR DISCHARGE
C                     PRESSURE, PSIA.
C
C           P4        HPT DISCHARGE PRESSURE, PSIA.
C
C           P4G       GUESS FOR THE HPT DISCHARGE PRESSURE, PSIA.
C
C           QC        COMPRESSOR TORQUE, FT-LBS.
C
C           QD        DYNAMOMETER TORQUE, FT-LBS.
```

```
C
C            QERR      PERCENTAGE GAS GENERATOR TORQUE DIFFERENTIAL
C                      ALLOWED DURING CONVERGENCE.
C
C            QFPT      FREE POWER TURBINE TORQUE, FT-LBS.
C
C            QHPT      HIGH PRESSURE TURBINE TORQUE, FT-LBS.
C
C            QL        QPERC EVALUATED AT MFL.
C
C            QPERC     PERCENT DIFFERENCE BETWEEN COMPRESSOR
C                      AND HIGH PRESSURE TURBINE (HPT) TORQUES.
C
C            QU        QPERC EVALUATED AT MFU.
C
C            Q1        QPERC EVALUATED AT MF1.
C
C            Q2        QPERC EVALUATED AT MF2.
C
C            T2        COMPRESSOR DISCHARGE TEMPERATURE, DEG.R.
C
C            T4        HPT DISCHARGE TEMPERATURE, DEG.R.
C
C
C            WW        DYNAMOMETER WATER WEIGHT, LBS.
C
C            WWERR     PERCENTAGE ERROR IN DYNO WATER WEIGHT
C                      ALLOWED DURING WATER WEIGHT CONVERGENCE.
C
C            X         VECTOR CONTAINING THE SCALED INPUTS AND OUTPUTS.
C
C            XCOUNT    COUNTER USED DURING ESTABLISHMENT OF
C                      FUEL FLOW RATE BOUNDS.
C
C            XK        COUNTER ON THE NUMBER OF FUNCTION EVALUATIONS
C                      DURING THE GOLDEN SECTION METHOD.
C            XN        NUMBER OF FUNCTION EVALUATIONS NEEDED FOR
C                      ME TO CONVERGE WITHIN MERR DURING GOLDEN
C                      SECTION METHOD ITERATION.
C
C            XR        VECTOR CONTAINING THE UNSCALED INPUTS AND
C                      OUTPUTS.
C
C            Z         VECTOR CONTAINING THE SCALING FACTORS FOR THE
C                      VARIOUS INPUTS AND OUTPUTS.
C
C
C
C        $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
```

```
C
C
       COMMON QC,NG,P2G,QHPT,MA,T2,MF,P4G,QFPT,MAF,T4,NS,QD,WW
       REAL NG,NS,MF,MAF,MA,NSO,NGO,MFO,MAFO,MAO,MFDEL,MFU,MFL,
    1      MF1,MF2,MFMIN,MERR
       DIMENSION A(2,2),B(2,2)
C
C
C  INPUT THE INITIAL GAS GENERATOR SPEED AND DYNO SPEED.
C
       WRITE(6,1)
   1   FORMAT(/,3X,'INPUT INITIAL GAS GENERATOR SPEED,"NG".')
C
       READ(5,*) NG
       WRITE(6,*) NG
C
C
       WRITE(6,3)
   3   FORMAT(/,3X,'INPUT INITIAL DYNO SPEED,"NS".')
C
       READ(5,*) NS
       WRITE(6,*) NS
C
C  ESTABLISH THE CONVERGENCE TOLERANCES.
C
       WWERR = 0.05
       MERR = 0.001
       PERR = 0.05
       QERR = 0.01
C
C
C  FIND AN INITIAL "GOOD GUESS" FOR 'MF' GIVEN 'NG' AND 'NS'.
C
       CALL NGNSMF(NG,NS,MF)
C
       MF1 = MF
       XCOUNT = 0.0
C
C
C  CONVERGE ON 'P2' AND 'P4' FOR THE GIVEN 'MF','NG' AND 'NS'.
C
  5    CALL P2P4(NG,NS,MF,PERR,QPERC,P2G,P4G)
C
C
C
       IF(ABS(QPERC).LT.QERR ) GO TO 300
C
C
C  ESTABLISH UPPER AND LOWER BOUNDS ON 'MF'.
C
```

```
         XCOUNT = XCOUNT + 1.0
         X1 = 1.0
         XSIGN = -1.0 * SIGN(X1,QPERC)
C        WRITE(6,*) 'QPERC = ',QPERC, 'XSIGN = ',XSIGN
         MFDEL = 2.0
         IF(XCOUNT.GT.1.5) GO TO 33
C
         XSIGN1 = XSIGN
         MF2 = MF1 + XSIGN * MFDEL
         MF = MF2
         QPERC1 = QPERC
         GO TO 5
C
   33    IF((XSIGN1*XSIGN).LT.0.5) GO TO 298
C
         MF2 = MF1 + XSIGN1 * MFDEL * XCOUNT
         MF = MF2
C
         GO TO 5
C
C   USE THE GOLDEN SECTION METHOD TO FIND THE VALUE OF 'MF' THAT
C   WILL LEAD TO ZERO GAS GENERATOR TORQUE MISMATCH.
C
  298    QPERC2 = QPERC
C
         IF(MF2.LT.MF1) GO TO 34
C
         MFU = MF2
         MFL = MF1
         QU = ABS(QPERC2)
         QL = ABS(QPERC1)
C
C
  34     MFU = MF1
         MFL = MF2
         QU = ABS(QPERC1)
         QL = ABS(QPERC2)
C
C
C        WRITE(6,170)
C 170    FORMAT(/,9X,'XMFL',14X,'XMF1',14X,'XMF2',14X,'XMFU',/)
C
         TAU = 0.381966
         XN = -2.078 * ALOG(MERR/100.0) + 3
C        WRITE(6,*) 'XN = ',XN
         MF1 = (1.0-TAU)*MFL + TAU*(MFU)
         CALL P2P4(NG,NS,MF1,PERR,QPERC,P2G,P4G)
         Q1 = ABS(QPERC)
         MF2 = (1.0-TAU)*MFU + TAU*(MFL)
         CALL P2P4(NG,NS,MF2,PERR,QPERC,P2G,P4G)
```

```
      Q2 = ABS(QPERC)
      XK = 3.0
   90 XK = XK + 1.0
C     WRITE(6,168)  MFL,MF1,MF2,MFU
C168  FORMAT(2X,4(F15.7,3X))
      MFMIN = (MF1 + MF2)/2
C     QDIFF = (Q1 + Q2)/2.0
C     WRITE(6,*) 'QDIFF = ',QDIFF
C     IF(QDIFF.LT.QERR) GO TO 92
      IF(XK.GT.XN) GO TO 92
      IF(Q1.GT.Q2) GO TO 91
      MFU = MF2
      QU = Q2
      MF2 = MF1
      Q2 = Q1
      MF1 = (1.0-TAU)*MFL + TAU*(MFU)
      CALL P2P4(NG,NS,MF1,PERR,QPERC,P2G,P4G)
      Q1 = ABS(QPERC)
      GO TO 90
  91  MFL = MF1
      QL = Q1
      MF1 = MF2
      Q1 = Q2
      MF2 = (1.0-TAU)*MFU + TAU*(MFL)
      CALL P2P4(NG,NS,MF2,PERR,QPERC,P2G,P4G)
      Q2 = ABS(QPERC)
      GO TO 90
   92 WRITE(6,169) MFMIN
  169 FORMAT(2X,'G.S.  MFMIN = ',F15.7)
      MF = MFMIN
C
C
      CALL SUBMA(NG,P2G,MA)
      CALL SUBT2(NG,P2G,T2)
      CALL SUBQC(NG,P2G,QC)
      CALL SUBQHT(NG,MA,T2,MF,P4G,QHPT)
      MAF = MA + MF
      CALL SUBT4(NG,MA,T2,MF,P4G,T4)
      CALL SUBQFT(MAF,T4,NS,QFPT)
      QPERC = 100*(QHPT-QC)/QHPT
C
C
C
C   EQUATE QD = QFPT FOR STEADY STATE AND SOLVE FOR 'WW' USING
C   NEWTON'S METHOD. NOTE THAT 'WWERR' HALTS THE ITERATION.
C
C
C   "WW1" IS THE INITIAL GUESS FOR DYNO WATER WEIGHT.  IT IS
C   NEEDED TO START NEWTON'S SCHEME.  THE VALUE IS FAIRLY
C   ARBITRARY BUT DO NOT USE 'WW1 = 0.0'!
```

```
C
      WW1 = 5.00
C
C   GG =  QD - QFPT, WHERE QD = FCN(WW1)
C
      C5 = 1.19294E-5
      C3 = 4.0E-6
      C4 = -20.0 + C3*NS*NS
 37   QD = C4 + C5*NS*NS*(WW1**1.3)
      GG = QD - QFPT
      GGP = 1.3*C5*NS*NS*(WW1**0.3)
      WW  = WW1 - GG/GGP
      WWDIFF = 100.0 * ABS((WW  - WW1)/WW1)
      IF(WWDIFF.LT.WWERR) GO TO 300
      WW1 = WW
      GO TO 37
C
C
C   SET VALUES TO INITIAL VALUES AND PRINT OUT.
C
 300  NGO = NG
      NSO = NS
      QCO = QC
      MAO = MA
      T2O = T2
      P2O = P2G
      QHPTO = QHPT
      MFO = MF
      P4O = P4G
      T4O = T4
      MAFO = MAF
      QFPTO = QFPT
      QDO = QD
      WWO = WW
C
C   CALL SUBROUTINE 'PART' TO GET THE COEFFICIENTS OF THE STATE
C   SPACE MATRICES, 'A' AND 'B'. THE RESULTS ARE SENT TO FILE 02.
C
      CALL PART(A,B)
C
C
C   PRINT OUTPUT TO THE SCREEN.
C
      WRITE(6,*) 'NGO = ',NGO
      WRITE(6,*) 'NSO = ',NSO
      WRITE(6,*) 'QCO = ',QCO
      WRITE(6,*) 'MAO = ',MAO
      WRITE(6,*) 'T2O = ',T2O
      WRITE(6,*) 'P2O = ',P2O
      WRITE(6,*) 'QHPTO = ',QHPTO
```

```
      WRITE(6,*) 'MFO = ',MFO
      WRITE(6,*) 'P4O = ',P4O
      WRITE(6,*) 'T4O = ',T4O
      WRITE(6,*) 'MAFO = ',MAFO
      WRITE(6,*) 'QFPTO = ',QFPTO
      WRITE(6,*) 'QDO = ',QDO
      WRITE(6,*) 'WWO = ',WWO
      WRITE(6,*)
      WRITE(6,*) 'P2ERR = ',PERR
      WRITE(6,*) 'P4ERR = ',PERR
      WRITE(6,*) 'MERR = ',MERR
      WRITE(6,*) 'WWERR = ',WWERR
C
C
C   PRINT OUTPUT TO FILE 02.
C
C     WRITE(2,*)
C     WRITE(2,*) 'NGO = ',NGO
C     WRITE(2,*)
C     WRITE(2,*) 'NSO = ',NSO
C     WRITE(2,*)
C     WRITE(2,*) 'QCO = ',QCO
C     WRITE(2,*)
C     WRITE(2,*) 'MAO = ',MAO
C     WRITE(2,*)
C     WRITE(2,*) 'T2O = ',T2O
C     WRITE(2,*)
C     WRITE(2,*) 'P2O = ',P2O
C     WRITE(2,*)
C     WRITE(2,*) 'QHPTO = ',QHPTO
C     WRITE(2,*)
C     WRITE(2,*) 'MFO = ',MFO
C     WRITE(2,*)
C     WRITE(2,*) 'P4O = ',P4O
C     WRITE(2,*)
C     WRITE(2,*) 'T4O = ',T4O
C     WRITE(2,*)
C     WRITE(2,*) 'MAFO = ',MAFO
C     WRITE(2,*)
C     WRITE(2,*) 'QFPTO = ',QFPTO
C     WRITE(2,*)
C     WRITE(2,*) 'QDO = ',QDO
C     WRITE(2,*)
C     WRITE(2,*) 'WWO = ',WWO
C     WRITE(2,*)
C     WRITE(2,*) 'QPERC = ',QPERC
C     WRITE(2,*)
C     WRITE(2,*) 'P2ERR = ',PERR
C     WRITE(2,*)
C     WRITE(2,*) 'P4ERR = ',PERR
```

```
C        WRITE(2,*)
C        WRITE(2,*) 'MERR = ',MERR
C        WRITE(2,*)
C        WRITE(2,*) 'WWERR = ',WWERR
C        WRITE(2,*)
C
C
C
         STOP
         END
C
C
C
C************************************************************
         SUBROUTINE NGNSMF(X1,X2,BR)
C************************************************************
C
C    THIS SUBROUTINE PRODUCES AN INITIAL "GOOD GUESS" FOR 'MF'
C    BASED ON THE SPECIFIED INPUTS, 'NG' AND 'NS'.
C
C
         DIMENSION X(5),C(21),Z(5),XR(5)
C
         XR(1) = X1
         XR(2) = X2
C
C
C    COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
          C( 1)=   1.982237
          C( 2)=  0.2461511
          C( 3)= -5.147902E-02
          C( 4)=  -1.884269
          C( 5)= -9.572456E-02
          C( 6)=  0.7639713
C
C    SCALING FACTORS.
C
         Z(1) =    36000.0
         Z(2) =    3000.0
         Z(3) =    240.0
C
         NIND = 2
C
         DO 686 I = 1,NIND
              X(I) = XR(I)/Z(I)
 686     CONTINUE
C
C    CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
```

```
      B = 0
      K = 0
      DO 70 J = 1,NIND
      DO 71 I = J,NIND
            K = K+1
      B = B+C(K)*X(J)*X(I)
   71 CONTINUE
   70 CONTINUE
C
      DO 72 J = 1,NIND
            K = K+1
      B = B+C(K)*X(J)
   72 CONTINUE
C
            K = K+1
      B = B+C(K)
C
C     WRITE(6,84) B
C  84 FORMAT(/,2X,'THE SCALED MF IS :',2X,G15.7)
C
C
      BR = B * Z(NIND + 1)
C
C
C  THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
      XHI =  240.0
      XLO =   70.0
C
      BR = AMAX1(XLO,BR)
      BR = AMIN1(XHI,BR)
C
C     WRITE(6,85) BR
C  85 FORMAT(/,2X,'MF IS :',2X,G15.7)
C
C
      RETURN
      END
C
C****************************************************************
      SUBROUTINE P2P4(NG,NS,MF,PERR,QPERC,P2G,P4G)
C****************************************************************
C
C  THE INPUTS TO THIS SUBROUTINE ARE 'NG','NS', AND 'MF'.
C  FOR THESE INPUTS THE SUBROUTINE CONVERGES ON 'P2'AND 'P4'
C  AND FINDS THE "STEADY STATE" TORQUE DIFFERENTIAL, WHICH
C  IS REPRESENTED IN PERCENTAGE FORM AS 'QPERC'.
C
      REAL NG,NS,MF,MAF,MA,NSO,NGO,MFO,MAFO,MAO,MFDEL,MFU,MFL
C
```

```
C   P2G AND P4G ARE NOMINAL VALUES OF COMPRESSOR DISCHARGE AND FPT
C   INLET PRESSURES. THEY PROVIDE AN INITIAL GUESS FOR THE CONVERGENCE
C   ROUTINE. P2ERR AND P4ERR ARE THE MAXIMUM ALLOWABLE DIFFERENCES
C   BETWEEN P2G AND P2 , AND BETWEEN P4G AND P4.
C
C
C
          P2G = 31.5
          P4G = 17.0
          P2ERR = PERR
          P4ERR = PERR
          ZS =  50.0
          ZX = 0.0
C
C
C   COMPUTE COMPRESSOR OUTPUTS.
C
  10      ZX = ZX + 1.0
C
          CALL SUBMA(NG,P2G,MA)
          CALL SUBT2(NG,P2G,T2)
          CALL SUBQC(NG,P2G,QC)
C
C
C   COMPUTE P2 AND CHECK AGAINST P2G.
C
 20       CALL SUBP2(NG,MA,T2,MF,P4G,P2)
C
          P2DIFF = 100.0 * ABS(P2 - P2G)/P2
          P2G = P2
C         P2G = P2G + 0.5*(P2-P2G)
          IF(ZX.GT.ZS) GO TO 511
          IF(P2DIFF.GT.P2ERR) GO TO 10
C
511        ZX = 0.0
C
C         COMPUTE REMAINING HPT OUTPUTS.
C
          MAF = MA + MF
C         WRITE(6,*) 'MAF = ',MAF
C
 47       CALL SUBT4(NG,MA,T2,MF,P4G,T4)
C
C   COMPUTE P4 AND CHECK AGAINST P4G.
C
          CALL SUBP4(MAF,T4,NS,P4)
C
          P4DIFF = 100.0 * ABS(P4 - P4G)/P4
C
          P4G = P4G + 0.5*(P4-P4G)
```

```
C        P4G = P4
         IF(P4DIFF.GT.P4ERR) GO TO 20
C
C   COMPUTE THE TORQUE MISMATCH (QHPT-QC).
C
C
         CALL SUBMA(NG,P2G,MA)
         CALL SUBT2(NG,P2G,T2)
         CALL SUBQC(NG,P2G,QC)
         CALL SUBQHT(NG,MA,T2,MF,P4G,QHPT)
         CALL SUBQFT(MAF,T4,NS,QFPT)
         QPERC = 100*(QHPT-QC)/QHPT
C
C
  919    RETURN
         END
C***********************************************************
         SUBROUTINE SUBMA(X1,X2,BR)
C***********************************************************
C
C THIS SUBROUTINE PRODUCES OUTPUT 'MA' FOR THE GIVEN INPUTS.
C
         DIMENSION X(5),C(21),Z(5),XR(5)
C
         XR(1) = X1
         XR(2) = X2
C
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
          C( 1)=   1.570198
          C( 2)=  -0.7270151
          C( 3)=   0.2529498
          C( 4)=   0.1880112
          C( 5)=  -0.6588774
          C( 6)=   0.3668176
C
C   SCALING FACTORS.
C
         Z(1) =    36000.0
         Z(2) =    43.0
         Z(3) =    13000.0
C
         NIND = 2
C
         DO 686 I = 1,NIND
              X(I) = XR(I)/Z(I)
 686     CONTINUE
C
C   CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
```

```
C
      B = 0
      K = 0
      DO 70 J = 1,NIND
      DO 71 I = J,NIND
            K = K+1
      B = B+C(K)*X(J)*X(I)
 71   CONTINUE
 70   CONTINUE
C
      DO 72 J = 1,NIND
            K = K+1
      B = B+C(K)*X(J)
 72   CONTINUE
C
            K = K+1
      B = B+C(K)
C
C
      WRITE(6,84) B
 84   FORMAT(/,2X,'THE SCALED MA IS :',2X,G15.7)
C
C
C
      BR = B * Z(NIND + 1)
C
C
C THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
      XHI =  13500.0
      XLO =  5500.0
C
      BR = AMAX1(XLO,BR)
      BR = AMIN1(XHI,BR)
C
C
      WRITE(6,85) BR
 85   FORMAT(/,2X,'MA IS :',2X,G15.7)
C
C
C
      RETURN
      END
C
C**************************************************************
      SUBROUTINE SUBT2(X1,X2,BR)
C**************************************************************
C
C THIS SUBROUTINE PRODUCES OUTPUT 'T2' FOR THE GIVEN INPUTS.
C
C
      DIMENSION X(5),C(21),Z(5),XR(5)
C
C
```

```
         XR(1) = X1
         XR(2) = X2
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
          C( 1)=  -0.5771397
          C( 2)=  2.203628
          C( 3)=  -1.040498
          C( 4)=  0.1354878
          C( 5)=  -0.4898891
          C( 6)=  0.7473461
C
C   SCALING FACTORS.
C
         Z(1) =    36000.0
         Z(2) =    43.0
         Z(3) =    800.0
C
         NIND = 2
C
 711     DO 500 I = 1,NIND
         X(I) = XR(I)/Z(I)
  500    CONTINUE
C
C
C   CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
         B = 0
         K = 0
         DO 70 J = 1,NIND
         DO 71 I = J,NIND
              K = K+1
         B = B+C(K)*X(J)*X(I)
   71    CONTINUE
   70    CONTINUE
C
         DO 72 J = 1,NIND
              K = K+1
         B = B+C(K)*X(J)
   72    CONTINUE
C
              K = K+1
         B = B+C(K)
C
C        WRITE(6,84) B
C  84    FORMAT(/,2X,'THE SCALED T2 IS :',2X,G15.7)
C
C
         BR = B * Z(NIND + 1)
C
```

```
C
C   THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
         XHI =  850.0
         XLO =  500.0
C
         BR = AMAX1(XLO,BR)
         BR = AMIN1(XHI,BR)
C
C
C  85    WRITE(6,85) BR
         FORMAT(/,2X,'T2 IS :',2X,G15.7)
C
C
         RETURN
         END
C
C************************************************************
         SUBROUTINE SUBQC(X1,X2,BR)
C************************************************************
C
C THIS SUBROUTINE PRODUCES OUTPUT 'QC' FOR THE GIVEN INPUTS.
C
         DIMENSION X(5),C(21),Z(5),XR(5)
C
         XR(1) = X1
         XR(2) = X2
C
C
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
          C( 1)=  -9.796132
          C( 2)=   20.03512
          C( 3)=  -10.70980
          C( 4)=  0.1464243
          C( 5)=   1.657819
          C( 6)= -0.3884839
C
C   SCALING FACTORS.
C
         Z(1) =     36000.0
         Z(2) =     43.0
         Z(3) =     130.0
C
         NIND = 2
C
 711     DO 500 I = 1,NIND
         X(I) = XR(I)/Z(I)
  500    CONTINUE
C
```

```
C
C   CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
        B = 0
        K = 0
        DO 70 J = 1,NIND
        DO 71 I = J,NIND
            K = K+1
        B = B+C(K)*X(J)*X(I)
   71   CONTINUE
   70   CONTINUE
C
        DO 72 J = 1,NIND
            K = K+1
        B = B+C(K)*X(J)
   72   CONTINUE
C
            K = K+1
        B = B+C(K)
C
C
        WRITE(6,84) B
   84   FORMAT(/,2X,'THE SCALED QC IS :',2X,G15.7)
C
C
C
        BR = B * Z(NIND + 1)
C
C
C   THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
        XHI =  130.0
        XLO =  40.0
C
        BR = AMAX1(XLO,BR)
        BR = AMIN1(XHI,BR)
C
C
        WRITE(6,85) BR
   85   FORMAT(/,2X,'QC IS :',2X,G15.7)
C
C
C
        RETURN
        END
C
C
C
C*****************************************************************
        SUBROUTINE SUBP2(X1,X2,X3,X4,X5,BR)
C*****************************************************************
C
C THIS SUBROUTINE PRODUCES OUTPUT 'P2' FOR THE GIVEN INPUTS.
C
```

```
C
         DIMENSION X(5),C(21),Z(6),XR(5)
C
C
         XR(1) = X1
         XR(2) = X2
         XR(3) = X3
         XR(4) = X4
         XR(5) = X5
C
C    COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
          C( 1)=  4.17287
          C( 2)=  -2.839741
          C( 3)=  1.223014
          C( 4)=  -1.854803
          C( 5)=  -10.26167
          C( 6)=  -0.2169524
          C( 7)=  -1.156939
          C( 8)=  2.860795
          C( 9)=  5.767990
          C(10)=  -0.101891
          C(11)=  0.2918
          C(12)=  -0.441640
          C(13)=  0.7359644
          C(14)=  -9.559825
          C(15)=  22.88968
          C(16)=  4.7794
          C(17)=  -2.558953
          C(18)=  0.272224
          C(19)=  6.295503
          C(20)=  -28.57775
          C(21)=  9.380198
C
C
C    SCALING FACTORS.
C
         Z(1) =     36000.0
         Z(2) =     13000.0
         Z(3) =     800.0
         Z(4) =     240.0
         Z(5) =     20.0
         Z(6) =     43.0
C
         NIND = 5
C
 711     DO 500 I = 1,NIND
         X(I) = XR(I)/Z(I)
  500    CONTINUE
C
```

```
C
C
C
C   CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
         B = 0
         K = 0
         DO 70 J = 1,NIND
         DO 71 I = J,NIND
              K = K+1
         B = B+C(K)*X(J)*X(I)
   71    CONTINUE
   70    CONTINUE
C
         DO 72 J = 1,NIND
              K = K+1
         B = B+C(K)*X(J)
   72    CONTINUE
C
              K = K+1
         B = B+C(K)
C
C        WRITE(6,84) B
C  84    FORMAT(/,2X,'THE SCALED P2 IS :',2X,G15.7)
C
C
         BR = B * Z(NIND + 1)
C
C
C   THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
         XHI =  43.0
         XLO =  20.0
C
         BR = AMAX1(XLO,BR)
         BR = AMIN1(XHI,BR)
C
C        WRITE(6,85) BR
C  85    FORMAT(/,2X,'P2 IS :',2X,G15.7)
C
C
         RETURN
         END
C
C
C************************************************************
         SUBROUTINE SUBT4(X1,X2,X3,X4,X5,BR)
C************************************************************
C
C THIS SUBROUTINE PRODUCES OUTPUT 'T4' FOR THE GIVEN INPUTS.
```

```
C
C
      DIMENSION X(5),C(21),Z(6),XR(5)
C
C
      XR( 1 ) = X1
      XR( 2 ) = X2
      XR( 3 ) = X3
      XR( 4 ) = X4
      XR( 5 ) = X5
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
C
       C(  1)=  -22.20944
       C(  2)=  10.79398
       C(  3)=  21.99301
       C(  4)=  86.64350
       C(  5)=  -208.0447
       C(  6)=  1.232848
       C(  7)=  -12.46899
       C(  8)=  -64.69914
       C(  9)=  180.0014
       C( 10)=  -0.6479730
       C( 11)=  -11.01693
       C( 12)=  20.21592
       C( 13)=  16.70037
       C( 14)=  -121.1824
       C( 15)=  183.1548
       C( 16)=  138.3667
       C( 17)=  -117.2714
       C( 18)=  -18.03533
       C( 19)=  72.75989
       C( 20)=  -229.4335
       C( 21)=  73.97864
C
C
C   SCALING FACTORS.
C
      Z( 1 ) =    36000.0
      Z( 2 ) =    13000.0
      Z( 3 ) =    800.0
      Z( 4 ) =    240.0
      Z( 5 ) =    20.0
      Z( 6 ) =    1800.0
C
C
      NIND = 5
C
 711  DO 500 I = 1,NIND
```

```
        X(I) = XR(I)/Z(I)
  500   CONTINUE
C
C
C
C
C  CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
        B = 0
        K = 0
        DO 70 J = 1,NIND
        DO 71 I = J,NIND
              K = K+1
        B = B+C(K)*X(J)*X(I)
   71   CONTINUE
   70   CONTINUE
C
        DO 72 J = 1,NIND
              K = K+1
        B = B+C(K)*X(J)
   72   CONTINUE
C
              K = K+1
        B = B+C(K)
C
C
        WRITE(6,84) B
   84   FORMAT(/,2X,'THE SCALED T4 IS :',2X,G15.7)
C
C
C
        BR = B * Z(NIND + 1)
C
C
C  THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
        XHI =  1800.0
        XLO =  1300.0
C
        BR = AMAX1(XLO,BR)
        BR = AMIN1(XHI,BR)
C
C
        WRITE(6,85) BR
   85   FORMAT(/,2X,'T4 IS :',2X,G15.7)
C
C
C
        RETURN
        END
C
C
C*************************************************************
        SUBROUTINE SUBQHT(X1,X2,X3,X4,X5,BR)
```

```
C*************************************************************
C
C THIS SUBROUTINE PRODUCES OUTPUT 'QHPT' FOR THE GIVEN INPUTS.
C
C
      DIMENSION X(5),C(21),Z(6),XR(5)
C
C
      XR(1) = X1
      XR(2) = X2
      XR(3) = X3
      XR(4) = X4
      XR(5) = X5
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
       C( 1)=  343.8178
       C( 2)=  -562.3596
       C( 3)=  -23.65895
       C( 4)=  -54.97896
       C( 5)=  98.09515
       C( 6)=  217.8508
       C( 7)=  3.591497
       C( 8)=  119.9962
       C( 9)=  -248.3938
       C(10)=  -0.1507291
       C(11)=  17.95723
       C(12)=  -17.87346
       C(13)=  -40.67739
       C(14)=  28.27711
       C(15)=  190.2205
       C(16)=  -160.9423
       C(17)=  260.8458
       C(18)=  21.40023
       C(19)=  -34.85067
       C(20)=  -219.0661
       C(21)=  62.16870
C
C
C   SCALING FACTORS.
C
      Z(1) =    36000.0
      Z(2) =    13000.0
      Z(3) =    800.0
      Z(4) =    240.0
      Z(5) =    20.0
      Z(6) =    130.0
C
C
C
```

```
       NIND = 5
C
 711   DO 500 I = 1,NIND
       X(I) = XR(I)/Z(I)
  500  CONTINUE
C
C
C
C
C  CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
       B = 0
       K = 0
       DO 70 J = 1,NIND
       DO 71 I = J,NIND
           K = K+1
       B = B+C(K)*X(J)*X(I)
   71  CONTINUE
   70  CONTINUE
C
       DO 72 J = 1,NIND
           K = K+1
       B = B+C(K)*X(J)
   72  CONTINUE
C
           K = K+1
       B = B+C(K)
C
C      WRITE(6,84) B
C  84  FORMAT(/,2X,'THE SCALED QHPT IS :',2X,G15.7)
C
C
       BR = B * Z(NIND + 1)
C
C
C  THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
       XHI =  130.0
       XLO =   40.0
C
       BR = AMAX1(XLO,BR)
       BR = AMIN1(XHI,BR)
C
C      WRITE(6,85) BR
C  85  FORMAT(/,2X,'QHPT IS :',2X,G15.7)
C
C
       RETURN
       END
C
```

```
C***********************************************************
         SUBROUTINE SUBP4(X1,X2,X3,BR)
C***********************************************************
C
C THIS SUBROUTINE PRODUCES OUTPUT 'P4' FOR THE GIVEN INPUTS.
C
C
         DIMENSION X(5),C(21),Z(6),XR(5)
C
C
         XR(1) = X1
         XR(2) = X2
         XR(3) = X3
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
          C( 1)=  0.1926178
          C( 2)=  1.158328
          C( 3)=  0.1008366
          C( 4)=  6.138049E-02
          C( 5)=  8.429369E-02
          C( 6)=  -5.136141E-02
          C( 7)=  -0.8789043
          C( 8)=  -1.171511
          C( 9)=  -4.834537E-02
          C(10)=  1.559548
C
C
C   SCALING FACTORS.
C
         Z(1) =    13000.0
         Z(2) =    1800.0
         Z(3) =   3000.0
         Z(4) =    20.0
C
C
C
C
         NIND = 3
C
 711     DO 500 I = 1,NIND
         X(I) = XR(I)/Z(I)
  500    CONTINUE
C
C
C
C
C   CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
         B = 0
```

```
         K = 0
         DO 70 J = 1,NIND
         DO 71 I = J,NIND
              K = K+1
         B = B+C(K)*X(J)*X(I)
   71    CONTINUE
   70    CONTINUE
C
         DO 72 J = 1,NIND
              K = K+1
         B = B+C(K)*X(J)
   72    CONTINUE
C
              K = K+1
         B = B+C(K)
C
C        WRITE(6,84) B
C  84    FORMAT(/,2X,'THE SCALED P4 IS :',2X,G15.7)
C
C
         BR = B * Z(NIND + 1)
C
C
C    THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
         XHI =  20.0
         XLO =  15.2
C
         BR = AMAX1(XLO,BR)
         BR = AMIN1(XHI,BR)
C
C        WRITE(6,85) BR
C  85    FORMAT(/,2X,' P4 IS :',2X,G15.7)
C
C
         RETURN
         END
C
C***********************************************************
         SUBROUTINE SUBQFT(X1,X2,X3,BR)
C***********************************************************
C
C  THIS SUBROUTINE PRODUCES OUTPUT 'QFT' FOR THE GIVEN INPUTS.
C
C
         DIMENSION X(5),C(21),Z(6),XR(5)
C
C
         XR(1) = X1
         XR(2) = X2
```

```
        XR(3) = X3
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
C
C
         C( 1)=  2.192477
         C( 2)=  0.8755642
         C( 3)=  -0.6626919
         C( 4)=  3.892829
         C( 5)=  -0.1769417
         C( 6)=  1.446682E-02
         C( 7)=  -1.83825
         C( 8)=  -7.607660
         C( 9)=  0.2095135
         C(10)=  3.747696
C
C   SCALING FACTORS.
C
        Z(1) =    13000.0
        Z(2) =    1800.0
        Z(3) =   3000.0
        Z(4) =   480.00
C
C
C
        NIND = 3
C
 711    DO 500 I = 1,NIND
        X(I) = XR(I)/Z(I)
  500   CONTINUE
C
C
C   CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
        B = 0
        K = 0
        DO 70 J = 1,NIND
        DO 71 I = J,NIND
             K = K+1
        B = B+C(K)*X(J)*X(I)
   71   CONTINUE
   70   CONTINUE
C
        DO 72 J = 1,NIND
             K = K+1
        B = B+C(K)*X(J)
   72   CONTINUE
C
             K = K+1
```

```
         B = B+C(K)
C
C        WRITE(6,84) B
C  84    FORMAT(/,2X,'THE SCALED QFPT IS :',2X,G15.7)
C
C
         BR = B * Z(NIND + 1)
C
C
C   THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
         XHI = 480.0
         XLO =  25.0
C
         BR = AMAX1(XLO,BR)
         BR = AMIN1(XHI,BR)
C
C        WRITE(6,85) BR
C  85    FORMAT(/,2X,'QFPT IS :',2X,G15.7)
C
C
         RETURN
         END
C
C************************************************************
         SUBROUTINE PART(A,B)
C************************************************************
C
C   THIS SUBROUTINE CALCULATES THE ELEMENTS OF THE 'A' AND 'B' MATRICES
C   IN THE STATE SPACE EQUATION:
C
C                        XDOT = A*X + B*U .
C
C
         COMMON QC,NG,P2,QH,MA,T2,MF,P4,QF,MAF,T4,NS,QD,WW
         DIMENSION A(2,2),B(2,2)
         REAL NG,NS,MF,MA,MAF,JG,JD
C
C
C
         JG = 0.009525 * 2 * 3.14159 / 60.0
         JD = 0.6738 * 2 * 3.14159 / 60.0
C
C   CALL SUBROUTINES TO GET PARTIAL DERIVATIVES.
C
         CALL DMA(NG,P2,DMADNG,DMADP2)
         CALL DT2(NG,P2,DT2DNG,DT2DP2)
         CALL DQC(NG,P2,DQCDNG,DQCDP2)
         CALL DP2(NG,MA,T2,MF,P4,DP2DNG,DP2DMF,DP2DMA,DP2DT2,DP2DP4)
         CALL DT4(NG,MA,T2,MF,P4,DT4DNG,DT4DMF,DT4DMA,DT4DT2,DT4DP4)
```

```
         CALL DQHT(NG,MA,T2,MF,P4,DQHDNG,DQHDMF,DQHDMA,DQHDT2,DQHDP4)
         CALL DP4(MAF,T4,NS,DP4DNS,DP4MAF,DP4DT4)
         CALL DQFT(MAF,T4,NS,DQFDNS,DQFMAF,DQFDT4)
         CALL DQD(NS,WW,DQDDNS,DQDDWW)
C
C    COMPUTE THE COEFFICIENTS OF THE STATE SPACE EQUATIONS (I.E. THE
C    ELEMENTS OF THE 'A' AND 'B' MATRICES).
C
         A1 = DQHDMA
         A2 = DQHDT2
         A3 = DQHDMF
         A4 = DQHDP4
         A5 = DQHDNG
         A6 = DQCDP2
         A7 = DQCDNG
         B1 = DMADP2
         B2 = DMADNG
         C1 = DT2DP2
         C2 = DT2DNG
         D1 = DP4MAF
         D2 = DP4DT4
         D3 = DP4DNS
         E1 = DP2DMA
         E2 = DP2DT2
         E3 = DP2DMF
         E4 = DP2DP4
         E5 = DP2DNG
         F1 = DT4DMA
         F2 = DT4DT2
         F3 = DT4DMF
         F4 = DT4DP4
         F5 = DT4DNG
         Z1 = DQFDNS
         Z2 = DQFMAF
         Z3 = DQFDT4
         Z4 = DQDDNS
         Z5 = DQDDWW
         G1 = 1-F4*D2
         G2 = (F1+F4*D1)/G1
         G3 = F2/G1
         G4 =  (F3+F4*D1)/G1
         G5 =  F4*D3/G1
         G6 =  F5/G1
         G7 =  (1-E1*B1-E2*C1)
         G8 =  (E1*B2+E2*C2+E5)/G7
         G9 =  E3/G7
         G10 = E4/G7
         G11 = (A1*B1+A2*C1-A6)
         G12 = (A1*B2+A2*C2+A5-A7)
         G13 = A3
```

```
         G14 = A4
         G15 = G11*G8+G12
         G16 = G11*G9+G13
         G17 = G11*G10+G14
         G18 = (D1+D2*G2)
         G19 = (D1+D2*G4)
         G20 = (D2*G3)
         G21 = (D2*G5+D3)
         G22 = D2*G6
         G23 = (G18*B1+G20*C1)
         G24 = (G18*B2+G20*C2+G22)
         G25 = (1-G23*G10)
         G26 = (G23*G8+G24)/G25
         G27 = (G23*G9+G19)/G25
         G28 = G21/G25
         G29 = (G15+G17*G26)
         G30 = (G16+G17*G27)
         G31 = G17*G28
         G32 = Z1+Z3*G5-Z4
         G33 = Z2+Z3*G2
         G34 = Z2+Z3*G4
         G35 = Z3*G3
         G36 = Z3*G6
         G37 = G33*B1+G35*C1
         G38 = G33*B2+G35*C2+G36
         G39 = (G37*G8+G38)
         G40 = (G37*G9+G34)
         G41 = G37*G10
         G42 = G32+G41*G28
         G43 = G39+G41*G26
         G44 = G40+G41*G27
C
C   FINAL FORM OF THE 'A' AND 'B' MATRICES.
C   ! NOTE ! ELEMENTS A33 AND B31 ARE NOT COMPUTED HERE BUT WERE
C   DETERMINED EXPERIMENTALLY FROM GAS TURBINE TEST DATA.
C
C            FOR ACCELERATIONS USE:
C
C                      A33 = -0.5
C                      B31 = 0.5
C
C            FOR DECELERATIONS USE:
C
C                      A33 = -0.87
C                      B31 = 0.87
C
         A11 = G29/JG
         A12 = G31/JG
         A13 = G30/JG
         A21 = G43/JD
```

```
      A22 = G42/JD
      A23 = G44/JD
      A31 = 0.0
      A32 = 0.0
      A33 = -0.5
      B11 = 0.0
      B12 = 0.0
      B21 = 0.0
      B22 = -Z5/JD
      B31 = 0.5
      B32 = 0.0
C
      WRITE(2,*) '"A" AND "B" MATRICES FOR NG = ',NG
      WRITE(2,*) '                         AND NS = ',NS
      WRITE(2,*) ' '
      WRITE(2,*) ' '
      WRITE(2,*) 'A11 = ', A11
      WRITE(2,*) 'A12 = ', A12
      WRITE(2,*) 'A13 = ', A13
      WRITE(2,*) 'A21 = ', A21
      WRITE(2,*) 'A22 = ', A22
      WRITE(2,*) 'A23 = ', A23
      WRITE(2,*) 'A31 = ', A31
      WRITE(2,*) 'A32 = ', A32
      WRITE(2,*) 'A33 = ', A33
      WRITE(2,*) 'B11 = ', B11
      WRITE(2,*) 'B12 = ', B12
      WRITE(2,*) 'B21 = ', B21
      WRITE(2,*) 'B22 = ', B22
      WRITE(2,*) 'B31 = ', B31
      WRITE(2,*) 'B32 = ', B32
C
      RETURN
      END
C
C***********************************************************
      SUBROUTINE DMA(X1,X2,DMADNG,DMADP2)
C***********************************************************
C
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C               DMA/DNG, DMA/DP2
C
C
      DIMENSION X(5),C(21),Z(5),XR(5)
C
      XR(1) = X1
      XR(2) = X2
C
```

```
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
          C( 1)=   1.570198
          C( 2)=  -0.7270151
          C( 3)=   0.2529498
          C( 4)=   0.1880112
          C( 5)=  -0.6588774
          C( 6)=   0.3668176
C
C   SCALING FACTORS.
C
      Z(1) =     36000.0
      Z(2) =     43.0
      Z(3) =     13000.0
C
      NIND = 2
C
      DO 686 I = 1,NIND
         X(I) = XR(I)/Z(I)
 686  CONTINUE
C
C
      DMADNG = 2.0*C(1)*X(1) + C(2)*X(2) + C(4)
      DMADNG = DMADNG*Z(3)/Z(1)
C
      DMADP2 = C(2)*X(1) + 2.0*C(3)*X(2) + C(5)
      DMADP2 = DMADP2*Z(3)/Z(2)
C
C
C
      RETURN
      END
C
C************************************************************
      SUBROUTINE DT2(X1,X2,DT2DNG,DT2DP2)
C************************************************************
C
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C                   DT2/DNG, DT2/DP2
C
C
C
C
      DIMENSION X(5),C(21),Z(5),XR(5)
C
C
      XR(1) = X1
      XR(2) = X2
```

```
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
         C( 1)=  -9.796132
         C( 2)=   20.03512
         C( 3)=  -10.70980
         C( 4)=  0.1464243
         C( 5)=   1.657819
         C( 6)= -0.3884839
C
C   SCALING FACTORS.
C
        Z( 1) =    36000.0
        Z( 2) =    43.0
        Z( 3) =    130.0
C
        NIND = 2
C
 711    DO 500 I = 1,NIND
        X(I) = XR(I)/Z(I)
  500   CONTINUE
C
C
        DQCDNG = 2.0*C(1)*X(1) + C(2)*X(2) + C(4)
        DQCDNG = DQCDNG*Z(3)/Z(1)
C
        DQCDP2 = C(2)*X(1) + 2.0*C(3)*X(2) + C(5)
        DQCDP2 = DQCDP2*Z(3)/Z(2)
C
C
        RETURN
        END
C
C
C
C**********************************************************************
      SUBROUTINE DP2(X1,X2,X3,X4,X5,DP2DNG,DP2DMF,DP2DMA,DP2DT2,DP2DP4)
C**********************************************************************
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C                  DP2/DNG, DP2/DMF
C
C
        DIMENSION X(5),C(21),Z(6),XR(5)
C
C
        XR(1) = X1
        XR(2) = X2
        XR(3) = X3
```

```
      XR( 4) = X4
      XR( 5) = X5
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
       C(  1)=  4.17287
       C(  2)=  -2.839741
       C(  3)=  1.223014
       C(  4)=  -1.854803
       C(  5)=  -10.26167
       C(  6)=  -0.2169524
       C(  7)=  -1.156939
       C(  8)=  2.860795
       C(  9)=  5.767990
       C( 10)=  -0.101891
       C( 11)=  0.2918
       C( 12)=  -0.441640
       C( 13)=  0.7359644
       C( 14)=  -9.559825
       C( 15)=  22.88968
       C( 16)=  4.7794
       C( 17)=  -2.558953
       C( 18)=  0.272224
       C( 19)=  6.295503
       C( 20)=  -28.57775
       C( 21)=  9.380198
C
C
C   SCALING FACTORS.
C
      Z( 1) =    36000.0
      Z( 2) =    13000.0
      Z( 3) =    800.0
      Z( 4) =    240.0
      Z( 5) =    20.0
      Z( 6) =    43.0
C
      NIND = 5
C
 711  DO 500 I = 1,NIND
      X(I) = XR(I)/Z(I)
  500 CONTINUE
C
C
      DP2DNG = 2*C(1)*X(1) + C(2)*X(2) + C(3)*X(3) + C(4)*X(4)
     1         + C(5)*X(5) + C(16)
      DP2DNG = DP2DNG*Z(6)/Z(1)
C
      DP2DMF = C(4)*X(1) + C(8)*X(2) + C(11)*X(3) + 2*C(13)*X(4)
     1         + C(14)*X(5) + C(19)
```

```
         DP2DMF = DP2DMF*Z(6)/Z(4)
C
         DP2DMA = C(2)*X(1) + C(7)*X(3) + C(8)*X(4) + 2*C(6)*X(2)
     1            + C(9)*X(5) + C(17)
         DP2DMA = DP2DMA*Z(6)/Z(2)
C
         DP2DT2 = C(3)*X(1) + C(7)*X(2) + C(11)*X(4) + 2*C(10)*X(3)
     1            + C(12)*X(5) + C(18)
         DP2DT2 = DP2DT2*Z(6)/Z(3)
C
         DP2DP4 = C(5)*X(1) + C(9)*X(2) + C(12)*X(3) + 2*C(15)*X(5)
     1            + C(14)*X(4) + C(20)
         DP2DP4 = DP2DP4*Z(6)/Z(5)
C
C
C
C
         RETURN
         END
C
C
C***********************************************************************
      SUBROUTINE DT4(X1,X2,X3,X4,X5,DT4DNG,DT4DMF,DT4DMA,DT4DT2,DT4DP4)
C***********************************************************************
C
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C                  DT4/DNG, DT4/DMF
C
C
C
         DIMENSION X(5),C(21),Z(6),XR(5)
C
C
         XR(1) = X1
         XR(2) = X2
         XR(3) = X3
         XR(4) = X4
         XR(5) = X5
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
C
          C( 1)=  -22.20944
          C( 2)=  10.79398
          C( 3)=  21.99301
          C( 4)=  86.64350
          C( 5)=  -208.0447
          C( 6)=  1.232848
```

```
           C(  7)=   -12.46899
           C(  8)=   -64.69914
           C(  9)=   180.0014
           C( 10)=   -0.6479730
           C( 11)=   -11.01693
           C( 12)=   20.21592
           C( 13)=   16.70037
           C( 14)=   -121.1824
           C( 15)=   183.1548
           C( 16)=   138.3667
           C( 17)=   -117.2714
           C( 18)=   -18.03533
           C( 19)=   72.75989
           C( 20)=   -229.4335
           C( 21)=   73.97864
C
C
C   SCALING FACTORS.
C
          Z( 1) =    36000.0
          Z( 2) =    13000.0
          Z( 3) =    800.0
          Z( 4) =    240.0
          Z( 5) =    20.0
          Z( 6) =    1800.0
C
C
          NIND = 5
C
 711      DO 500 I = 1,NIND
          X(I) = XR(I)/Z(I)
  500     CONTINUE
C
C
          DT4DNG = 2*C(1)*X(1) + C(2)*X(2) + C(3)*X(3) + C(4)*X(4)
     1             + C(5)*X(5) + C(16)
          DT4DNG = DT4DNG*Z(6)/Z(1)
C
          DT4DMF = C(4)*X(1) + C(8)*X(2) + C(11)*X(3) + 2*C(13)*X(4)
     1             + C(14)*X(5) + C(19)
          DT4DMF = DT4DMF*Z(6)/Z(4)
C
          DT4DMA = C(2)*X(1) + C(7)*X(3) + C(8)*X(4) + 2*C(6)*X(2)
     1             + C(9)*X(5) + C(17)
          DT4DMA = DT4DMA*Z(6)/Z(2)
C
          DT4DT2 = C(3)*X(1) + C(7)*X(2) + C(11)*X(4) + 2*C(10)*X(3)
     1             + C(12)*X(5) + C(18)
          DT4DT2 = DT4DT2*Z(6)/Z(3)
C
```

```
         DT4DP4 = C(5)*X(1) + C(9)*X(2) + C(12)*X(3) + 2*C(15)*X(5)
     1            + C(14)*X(4) + C(20)
         DT4DP4 = DT4DP4*Z(6)/Z(5)
C
C
C
C
         RETURN
         END
C
C
C***********************************************************************
      SUBROUTINE DQHT(X1,X2,X3,X4,X5,DQHDNG,DQHDMF,DQHDMA,DQHDT2,DQHDP4)
C***********************************************************************
C
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C           DQH/DNG, DQH/DMF, DQH/DMA, DQH/DT2, DQH/DP4
C
C
C
         DIMENSION X(5),C(21),Z(6),XR(5)
C
C
         XR(1) = X1
         XR(2) = X2
         XR(3) = X3
         XR(4) = X4
         XR(5) = X5
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
          C( 1)=  343.8178
          C( 2)=  -562.3596
          C( 3)=  -23.65895
          C( 4)=  -54.97896
          C( 5)=  98.09515
          C( 6)=  217.8508
          C( 7)=  3.591497
          C( 8)=  119.9962
          C( 9)=  -248.3938
          C(10)=  -0.1507291
          C(11)=  17.95723
          C(12)=  -17.87346
          C(13)=  -40.67739
          C(14)=  28.27711
          C(15)=  190.2205
          C(16)=  -160.9423
          C(17)=  260.8458
```

```
          C( 18)=  21.40023
          C( 19)=  -34.85067
          C( 20)=  -219.0661
          C( 21)=  62.16870
C
C
C   SCALING FACTORS.
C
         Z( 1) =    36000.0
         Z( 2) =    13000.0
         Z( 3) =    800.0
         Z( 4) =    240.0
         Z( 5) =    20.0
         Z( 6) =    130.0
C
C
C
         NIND = 5
C
 711     DO 500 I = 1,NIND
         X( I) = XR( I)/Z( I)
  500    CONTINUE
C
C
C
         DQHDNG = 2*C( 1)*X( 1) + C( 2)*X( 2) + C( 3)*X( 3) + C( 4)*X( 4)
     1            + C( 5)*X( 5) + C( 16)
         DQHDNG = DQHDNG*Z( 6)/Z( 1)
C
         DQHDMF = C( 4)*X( 1) + C( 8)*X( 2) + C( 11)*X( 3) + 2*C( 13)*X( 4)
     1            + C( 14)*X( 5) + C( 19)
         DQHDMF = DQHDMF*Z( 6)/Z( 4)
C
         DQHDMA = C( 2)*X( 1) + C( 7)*X( 3) + C( 8)*X( 4) + 2*C( 6)*X( 2)
     1            + C( 9)*X( 5) + C( 17)
         DQHDMA = DQHDMA*Z( 6)/Z( 2)
C
         DQHDT2 = C( 3)*X( 1) + C( 7)*X( 2) + C( 11)*X( 4) + 2*C( 10)*X( 3)
     1            + C( 12)*X( 5) + C( 18)
         DQHDT2 = DQHDT2*Z( 6)/Z( 3)
C
         DQHDP4 = C( 5)*X( 1) + C( 9)*X( 2) + C( 12)*X( 3) + 2*C( 15)*X( 5)
     1            + C( 14)*X( 4) + C( 20)
         DQHDP4 = DQHDP4*Z( 6)/Z( 5)
C
C
         RETURN
         END
C
C***********************************************************
```

```
         SUBROUTINE DP4(X1,X2,X3,DP4DNS,DP4MAF,DP4DT4)
C************************************************************
C
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C                          DP4/DNS
C
C
C
         DIMENSION X(5),C(21),Z(6),XR(5)
C
C
         XR(1) = X1
         XR(2) = X2
         XR(3) = X3
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
          C( 1)=  0.1926178
          C( 2)=  1.158328
          C( 3)=  0.1008366
          C( 4)=  6.138049E-02
          C( 5)=  8.429369E-02
          C( 6)=  -5.136141E-02
          C( 7)=  -0.8789043
          C( 8)=  -1.171511
          C( 9)=  -4.834537E-02
          C( 10)=  1.559548
C
C
C   SCALING FACTORS.
C
         Z(1) =     13000.0
         Z(2) =     1800.0
         Z(3) =    3000.0
         Z(4) =     20.0
C
C
C
C
         NIND = 3
C
 711     DO 500 I = 1,NIND
         X(I) = XR(I)/Z(I)
  500    CONTINUE
C
C
         DP4DNS = C(3)*X(1) + C(5)*X(2) + 2*C(6)*X(3) + C(9)
         DP4DNS = DP4DNS*Z(4)/Z(3)
```

```
C
         DP4MAF = C(2)*X(2) + C(3)*X(3) + 2*C(1)*X(1) + C(7)
         DP4MAF = DP4MAF*Z(4)/Z(1)
C
         DP4DT4 = C(1)*X(1) + C(5)*X(3) + 2*C(4)*X(2) + C(8)
         DP4DT4 = DP4DT4*Z(4)/Z(2)
C
C
         RETURN
         END
C
C***************************************************************
         SUBROUTINE DQFT(X1,X2,X3,DQFDNS,DQFMAF,DQFDT4)
C***************************************************************
C
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C                DQF/DNS, DQF/DMAF, DQF/DT4
C
C
C
         DIMENSION X(5),C(21),Z(6),XR(5)
C
C
         XR(1) = X1
         XR(2) = X2
         XR(3) = X3
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
C
C
          C( 1)=  2.192477
          C( 2)=  0.8755642
          C( 3)=  -0.6626919
          C( 4)=  3.892829
          C( 5)=  -0.1769417
          C( 6)=  1.446682E-02
          C( 7)=  -1.83825
          C( 8)=  -7.607660
          C( 9)=  0.2095135
          C(10)=  3.747696
C
C   SCALING FACTORS.
C
         Z(1) =    13000.0
         Z(2) =     1800.0
         Z(3) =   3000.0
         Z(4) =   480.00
```

```
C
C
C
         NIND = 3
C
 711     DO 500 I = 1,NIND
         X(I) = XR(I)/Z(I)
  500    CONTINUE
C
C
C
         DQFDNS = C(3)*X(1) + C(5)*X(2) + 2*C(6)*X(3) + C(9)
         DQFDNS = DQFDNS*Z(4)/Z(3)
C
         DQFMAF = C(2)*X(2) + C(3)*X(3) + 2*C(1)*X(1) + C(7)
         DQFMAF = DQFMAF*Z(4)/Z(1)
C
         DQFDT4 = C(1)*X(1) + C(5)*X(3) + 2*C(4)*X(2) + C(8)
         DQFDT4 = DQFDT4*Z(4)/Z(2)
C
C
C
         RETURN
         END
C
C************************************************************
         SUBROUTINE DQD(X1,X2,DQDDNS,DQDDWW)
C************************************************************
C
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C                DQD/DNS, DQD/DWW
C
C
C   SCALING FACTORS
C
        XQD  = 480.
        XNS  = 3000.
        XWW  = 49.
C
         C1 = -20.0
         C2 = 4.0E-6
         C3 = 1.19294E-5
C
         DQDDNS = 2*X1*C2 + 2*C3*(X2**1.3)*X1
C
         DQDDWW = 1.3*C3*X1*X1*(X2**0.3)
C
C        DQDDNS = DQDDNS*XNS/XQD
```

```
C         DQDDWW = DQDDWW*XWW/XQD
C
C
C
          RETURN
          END
$ENTRY
```

APPENDIX D

NONLINEAR DYNAMIC PROGRAM

```
TITLE GT DYNAMIC PROGRAM
*
*
*   ******************************************************************
*   ******************************************************************
*   **                                                              **
*   **                BOEING MODEL 502-6A GAS TURBINE               **
*   **                                                              **
*   **                  DYNAMIC COMPUTER SIMULATION                 **
*   **                                                              **
*   ******************************************************************
*   ******************************************************************
*
*
*
*   THIS PROGRAM SIMULATES THE DYNAMIC RESPONSE OF THE NPS BOEING GAS
*   TURBINE PROPULSION TEST FACILITY USING A NONLINEAR SIMULATION
*   AND A LINEAR (STATE SPACE) SIMULATION.
*
*
*        THE DEVELOPMENT OF THIS COMPUTER SIMULATION IS DESCRIBED IN:
*
*             HERDA, V.J., 'MARINE GAS TURBINE MODELING
*             FOR MODERN CONTROL DESIGN',(M.S. THESIS,
*             NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA.,
*             JUNE 1986).
*
*             GOOD LUCK!!!!!!!
*
*  #######################################################################
*
*
*
PARAM JG=0.009525, JD=0.6738, PI=3.14159, T = 2.00
*
*   THE FOLLOWING VALUES LISTED ON THE FUNCTION CARD ARE FOR FUEL FLOW,
*   GAS GENERATOR SPEED, AND DYNO SPEED AS A FUNCTION OF TIME.
*   THESE VALUES WERE OBTAINED FROM STRIP CHART RECORDS AND ARE ENTERED
*   IN THE FORM (E.G. FUEL FLOW)  ...TIME(SEC), FUEL FLOW.....
*
*        THIS SET IS FOR EXPERIMENTAL RUN # 1.
*
AFGEN MFDATA = 0.0,104.3, .05,104.3, ...
```

```
          .1,144.4, .15,201.8, ...
          .2,219.0, .25,224.7, .3,224.7, .35,219.0, .40,236.2, ...
          .45,236.2, .50,224.7, .55,230.4, .60,241.9, ...
          .65,213.2, .7,213.2, .75,184.6, .80,167.4, .85,155.9, ...
          .90,155.9, .95,150.2, 1.0,144.4, 1.05,138.7, ...
          1.1,138.7, ...
          1.15,138.7, 1.2,138.7, 1.25,138.7
AFGEN NGDATA = 0.0,0, .05,0, ...
         .1,0, .15,1, ...
         .2,2.5, .25,4, .3,6, .35,8, .40,10, ...
         .45,11.5, .50,14, .55,16, .60,18, ...
         .65,20, .7,21.5, .75,23, .80,24, .85,25, ...
         .90,25, .95,25.5, 1.0,25.5,1.05,25, ...
         1.1,25, ...
         1.15,24.5, 1.2,24.5, 1.25,24.5
AFGEN NSDATA = 0.0,0, .05,0, ...
         .1,0, .15,0, ...
         .2,0.5, .25,1, .3,2, .35,2.5, .40,3.5, ...
         .45,4.5, .50,6, .55,7, .60,8, ...
         .65,9, .7,10.5, .75,12, .80,13, .85,13.5, ...
         .90,14.5, .95,15, 1.0,15.5,1.05,16, ...
         1.1,16, ...
         1.15,16, 1.2,16, 1.25,16
*
*        THIS SET IS FOR EXPERIMENTAL RUN # 4.
*
* AFGEN MFDATA = 0.0,1.0, .05,2.0, ...
*        .1,3.0, .15,6.0, ...
*        .2,13.0, .25,15.0, .3,17.0, .35,19.0, .40,21.0, ...
*        .45,22.0, .50,23.0, .55,23.0, .60,23.0, ...
*        .65,23.0, .7,22.0, .75,21.0, .80,20.0, .85,19.0, ...
*        .90,17.0, .95,14.5, 1.0,14.0, 1.05,10.0, ...
*        1.1,9.50, ...
*        1.15,7.00, 1.2,8.00, 1.25,8.00, ...
*        1.30,8.00, 1.35,8.00, 1.40,8.00, 1.45,9.00, 1.50,9.00, ...
*        1.55,9.50, 1.60,9.50, ...
*        1.65,9.50, 1.70,9.50, 1.75,9.50, 1.80,9.50, 1.85,9.50, ...
*        1.90,9.50, 1.95,9.50, 2.00,9.50, 2.05,9.50, 2.10,9.00
* AFGEN NGDATA = 0.0,4.00, .05,4.00, ...
*        .1,4.00, .15,5.00, ...
*        .2,6.00, .25,7.00, .3,8.50, .35,10.50, .40,13.0, ...
*        .45,15.0, .50,17.5, .55,20.0, .60,22.0, ...
*        .65,24.5, .7,26.5, .75,28.0, .80,30.0, .85,31.0, ...
*        .90,32.5, .95,33.0, 1.0,34.0,1.05,34.0, ...
*        1.1,34.5, ...
*        1.15,34.0, 1.2,34.0, 1.25,33.5, 1.3,33.0,1.35,33.0,...
*        1.40,33.0, 1.45,32.5, 1.50,32.0, 1.55,32.0,1.60,32.0
* AFGEN NSDATA = 0.0,1.00, .05,1.00, ...
*        .1,1.00, .15,1.00, ...
*        .2,2.00, .25,2.50, .3,3.50, .35,4.50, .40,6.00, ...
```

```
*          .45,7.00, .50,8.50, .55,10.0, .60,11.0, ...
*          .65,12.5, .7,14.0, .75,15.0, .80,16.0, .85,17.0, ...
*          .90,17.5, .95,18.0, 1.0,19.0,1.05,19.0, ...
*          1.1,19.0, ...
*          1.15,20.0, 1.2,20.0, 1.25,20.0, 1.3,20.0,1.35,20.0,...
*          1.40,20.0, 1.45,19.5, 1.50,19.5, 1.55,19.5,1.60,19.5
*
*          THIS SET IS FOR EXPERIMENTAL RUN # 10.
*
* AFGEN MFDATA = 0.0,3.0, .05,7.5, ...
*          .1,15.0, .15,22.5, ...
*          .2,22.5, .25,20.5, .3,21.0, .35,23.5, .40,22.0, ...
*          .45,21.0, .50,23.0, .55,24.0, .60,24.0, ...
*          .65,23.0, .7,24.0, .75,23.0, .80,23.0, .85,18.0, ...
*          .90,16.0, .95,13.0, 1.0,13.0, 1.05,8.00, ...
*          1.1,8.00, ...
*          1.15,7.00, 1.2,7.00, 1.25,7.00, ...
*          1.30,7.50, 1.35,7.50, 1.40,7.00, 1.45,8.50, 1.50,9.00, ...
*          1.55,9.00, 1.60,9.00, ...
*          1.65,9.00, 1.70,9.00, 1.75,9.00, 1.80,9.00, 1.85,9.00, ...
*          1.90,9.00, 1.95,9.00, 2.00,9.00, 2.05,9.00, 2.10,9.00
* AFGEN NGDATA = 0.0,5.00, .05,5.50, ...
*          .1,6.00, .15,7.00, ...
*          .2,8.50, .25,10.0, .3,11.5, .35,13.00, .40,15.0, ...
*          .45,16.5, .50,18.5, .55,19.0, .60,22.0, ...
*          .65,24.0, .7,26.0, .75,27.5, .80,29.0, .85,31.0, ...
*          .90,32.0, .95,33.0, 1.0,34.0,1.05,34.5, ...
*          1.1,35.0, ...
*          1.15,35.0, 1.2,35.0, 1.25,34.5, 1.3,34.0,1.35,34.0,...
*          1.40,34.0, 1.45,33.5, 1.50,33.5, 1.55,33.0,1.60,33.0, ...
*          1.65,33.0, 1.70,33.0, 1.75,33.0, 1.80,33.0, 1.85,33.0, ...
*          1.90,33.0, 1.95,33.0, 2.00,33.0, 2.05,33.0, 2.10,33.0
* AFGEN NSDATA = 0.0,4.00, .05,4.00, ...
*          .1,4.00, .15,4.50, ...
*          .2,5.00, .25,5.50, .3,6.00, .35,7.00, .40,8.00, ...
*          .45,9.00, .50,10.0, .55,11.0, .60,12.0, ...
*          .65,13.0, .7,14.0, .75,15.0, .80,16.0, .85,17.5, ...
*          .90,19.0, .95,20.0, 1.0,20.5,1.05,21.5, ...
*          1.1,22.0, ...
*          1.15,22.5, 1.2,23.0, 1.25,23.0, 1.3,23.5,1.35,23.5,...
*          1.40,23.5, 1.45,24.0, 1.50,24.0, 1.55,24.0,1.60,24.0
*
INITIAL
* "A" AND "B" MATRICES FOR NG =        25000.0000000
*                      AND NS =          960.0000000
*
*

           A11 =            -12.8091500
           A12 =             -4.4315500
           A13 =           2108.6270000
```

```
          A21 =               0.3025516
          A22 =              -5.2999960
          A23 =               0.7975193
          A31 =               0.0000000
          A32 =               0.0000000
          A33 =              -0.5000000
          B11 =               0.0000000
          B12 =               0.0000000
          B21 =               0.0000000
          B22 =            -368.9609000
          B31 =               0.5000000
          B32 =               0.0000000
*
*    "A" AND "B" MATRICES FOR NG =            29500.0000000
*                         AND NS =             1090.0000000
*
*
*         A11 =             -18.5633600
*         A12 =             -13.8810100
*         A13 =            2864.8290000
*         A21 =               0.5190844
*         A22 =              -7.5316480
*         A23 =              -1.7081720
*         A31 =               0.0000000
*         A32 =               0.0000000
*         A33 =              -0.5000000
*         B11 =               0.0000000
*         B12 =               0.0000000
*         B21 =               0.0000000
*         B22 =            -503.6293000
*         B31 =               0.5000000
*         B32 =               0.0000000
*
*    AVERAGE "A" AND "B" MATRICES FOR RUN#1.
*
*        A11 = -15.686
*        A12 = -9.157
*        A13 = 2486.73
*
*        A21 = 0.4109
*        A22 = -6.416
*        A23 = -0.455
*
*        A33 = -0.5
*
*        B22 = -436.3
*        B31 = 0.5
*
*    ESTABLISH INITIAL CONDITIONS.
*
```

```
          WRITE(3,11)
   11     FORMAT(/,2X,'INPUT INITIAL GAS GEN.  SPEED ,NGO')
          READ(4,10) NGO
   10     FORMAT(F15.7)
          WRITE(3,21) NGO
   21     FORMAT(F15.7)
          WRITE(3,31)
   31     FORMAT(/,2X,'INPUT INITIAL DYNO SPEED, NSO')
          READ(4,30) NSO
   30     FORMAT(F15.7)
          WRITE(3,41) NSO
   41     FORMAT(F15.7)
*
*    CALL STEADY STATE PROGRAM TO GET INITIAL FUEL FLOWRATE, MFO,
*    AND DYNO WATER WEIGHT, WWO.
*
          CALL STEADY(NGO,NSO,MFO,WWO)
*
          WRITE(3,33)    MFO
   33     FORMAT(/,2X,'THE ORIGINAL FUEL FLOW RATE IS : ',F15.7)
*
          WRITE(3,34)    WWO
   34     FORMAT(/,2X,'THE ORIGINAL WATER WEIGHT IS : ',F15.7)
*
*   SET INITIAL STATE PERTURBATION TO ZERO
*
          DNG = 0.0
          DNS = 0.0
          DE  = 0.0
*
          EO  = MFO
*
*
*
DERIVATIVE
*
*    COMPUTE INPUT TO THE NONLINEAR MODEL, MF(T), WW(T).
*
*
*    RUN #1
*
*
          MFM = AFGEN(MFDATA,TIME)
          WW1 = 2.263*RAMP(0.05)
          WW2 = -2.263*RAMP(0.75)
*
*
*    RUN #4
*
*         DIV = AFGEN(MFDATA,TIME)
```

```
*          MFM = MFO-4.489*(DIV-1.0)
*          WW1 = .7069*RAMP(0.00)
*          WW2 = -.7069*RAMP(1.60)
*
*    RUN #10
*
*          DIV = AFGEN(MFDATA,TIME)
*          MFM = MFO+5.217*(DIV-3.0)
*          WW1 = .6130*RAMP(0.00)
*          WW2 = -.6130*RAMP(1.60)
*
*    ALL RUNS
*
*    DYNAMIC EQUATIONS FOR NONLINEAR MODEL.
*
           E = REALPL(MFO,T,MFM)
           WW = WWO+WW1+WW2
           NGDOT = (DELQG/JG)*60/(2*PI)
           NG = INTGRL(NGO,NGDOT)
           NSDOT = (DELQD/JD)*60/(2*PI)
           NS = INTGRL(NSO,NSDOT)
*
           DMF = MFM-MFO
           DWW = WW-WWO
           DNGDOT  = A11*DNG + A12*DNS + A13*DE
           DNSDOT  = A21*DNG + A22*DNS + A23*DE  + B22*DWW
           DEDOT   = A33*DE  + B31*DMF
           DNG=INTGRL(0.0,DNGDOT)
           DNS=INTGRL(0.0,DNSDOT)
           DE =INTGRL(0.0,DEDOT)
           NGF = NGO + DNG
           NSF = NSO + DNS
           EF  = EO  + DE
*
*
DYNAMIC
*
*    THE STATEMENTS IN THE PREVIOUS (DERIVATIVE) SECTION YIELD VALUES
*    OF 'NG', AND 'NS' AS CALCULATED BY THE NONLINEAR AND STATE SPACE
*    MODELS.  THE STATEMENTS BELOW COMPUTE THE VALUES OF 'NG', AND 'NS'
*    AS RECORDED FROM GAS TURBINE TEST DATA.
*
*    RUN #1
*
           NGD = NGO+183.67*AFGEN(NGDATA,TIME)
           NSD = NSO+8.125*AFGEN(NSDATA,TIME)
*
*    RUN #4
*
*          NGDIV = NLFGEN(NGDATA,TIME)
```

```
*          NSDIV = NLFGEN(NSDATA,TIME)
*          NGD = NGO-185.36*(NGDIV-4)
*          NSD = NSO-16.27*(NSDIV-1)
*
*    RUN #10
*
*          NGDIV = NLFGEN(NGDATA,TIME)
*          NSDIV = NLFGEN(NSDATA,TIME)
*          NGD = NGO+192.96*(NGDIV-5)
*          NSD = NSO+10.65*(NSDIV-4)
*
*
*    CALL SUBROUTINE 'TORK', WHICH USES THE STEADY STATE MODEL TO GET
*    COMPRESSOR AND HPT TORQUE VALUES.
*
PROCED DELQG,DELQD = BLK(NG,NS,E,WW)
          CALL TORK(NG,NS,E,WW,DELQG,DELQD)
          CONTINUE
ENDPRO
TERMINAL
METHOD STIFF
RELERR NS = 1.E-6, NG = 1.E-6, DNS = 1.E-6, DNG = 1.E-6, DQ = 1.E-6
ABSERR NS = 1.E-5, NG = 1.E-5, DNS = 1.E-5, DNG = 1.E-5, DQ = 1.E-5
CONTROL FINTIM=2.00,DELT=1.E-5
PRINT 0.05,MFM,NGD,NG,NGF,NSD,NS,NSF
* SAVE 0.05,MFM,NG,NGD,NS,NSD
SAVE 0.05,MFM,NG,NGD,NS,NSD,NGF,NSF,EF
*
*    RUN #1
*
* GRAPH (DE=TEK618)  TIME(LO=0.0,SC=0.2,TI=.50,NI=10,UN=SEC) ...
*                   NS(LO=900,SC=100,TI=1.6,NI=5,UN=RPM) ...
*                   NSD(LO=900,SC=100,TI=1.6,NI=5,UN=RPM) ...
*                   NSF(LO=900,SC=100,TI=1.6,NI=5,UN=RPM) ...
*                   EF(LO=100,SC=10.,TI=2.,NI=4,UN='LB/HR')
*                   MFM(LO=100,SC=10.,TI=2.,NI=4,UN='LB/HR')
*                   NG(LO=24000,SC=1000,TI=1.1428,NI=7,UN=RPM) ...
*                   NGD(LO=24000,SC=1000,TI=1.1428,NI=7,UN=RPM) ...
*                   NGF(LO=24000,SC=1000,TI=1.1428,NI=7,UN=RPM) ...
*
*    RUN #4
*
* GRAPH (DE=TEK618)  TIME(LO=0.0,SC=0.2,TI=.50,NI=10,UN=SEC) ...
*                   NG(LO=24000,SC=1000,TI=1.33,NI=6,UN=RPM) ...
*                   NGD(LO=24000,SC=1000,TI=1.33,NI=6,UN=RPM) ...
*                   NS(LO=700,SC=100,TI=1.6,NI=5,UN=RPM) ...
*                   NSD(LO=700,SC=100,TI=1.6,NI=5,UN=RPM) ...
*                   MFM(LO=100,SC=10.,TI=1.6,NI=5,UN='LB/HR')
*
*    RUN #4
```

```
*
* GRAPH (DE=TEK618)  TIME(LO=0.0,SC=0.2,TI=.50,NI=10,UN=SEC) ...
*                   NG(LO=21000,SC=1000,TI=1.33,NI=6,UN=RPM) ...
*                   NGD(LO=21000,SC=1000,TI=1.33,NI=6,UN=RPM) ...
*                   NS(LO=900,SC=100,TI=2.0,NI=4,UN=RPM) ...
*                   NSD(LO=900,SC=100,TI=2.0,NI=4,UN=RPM) ...
*                   MFM(LO=80,SC=10.,TI=2.0,NI=4,UN='LB/HR')
*
*   RUN #1  THIS IS FOR THESIS PRESENTATION FIGURES
*
GRAPH (G2, DE=TEK618)  TIME(LO=0.0,SC=0.2,TI=.50,NI=10,UN=SEC) ...
                    NSD(LO=900,SC=100,TI=1.6,NI=5,UN=RPM) ...
                    NS(LO=900,SC=100,TI=1.6,NI=5,UN=RPM)
*                   NSF(LO=900,SC=100,TI=1.6,NI=5,UN=RPM)
*                   EF(LO=100,SC=10.,TI=2.,NI=4,UN='LB/HR') ...
*                   MFM(LO=100,SC=10.,TI=2.,NI=4,UN='LB/HR')
GRAPH (G1, DE=TEK618)  TIME(LO=0.0,SC=0.2,TI=.50,NI=10,UN=SEC) ...
                    NGD(LO=24000,SC=1000,TI=1.1428,NI=7,UN=RPM) ...
                    NG(LO=24000,SC=1000,TI=1.1428,NI=7,UN=RPM)
*                   NGF(LO=24000,SC=1000,TI=1.1428,NI=7,UN=RPM)
LABEL (G1)  GAS GENERATOR SPEED
LABEL (G2)  DYNAMOMETER SPEED
C
END
STOP
FORTRAN
C
C***********************************************************************
      SUBROUTINE STEADY(NG,NS,MF,WW)
C***********************************************************************
C
C
C   THIS PROGRAM PROVIDES THE INITIALIZATION PROCESS FOR THE DYNAMIC
C   PROGRAM. SPECIFICALLY, THE USER INPUTS GAS GENERATOR AND DYNO
C   SPEEDS, AND THE PROGRAM USES STEADY STATE MAPS (IN EQUATION FORM)
C   OF SYSTEM INPUTS/OUTPUTS TO FIND STEADY STATE VALUES.
C
C
C
      REAL NG,NS,MF,MAF,MA,NSO,NGO,MFO,MAFO,MAO,MFDEL,MFU,MFL,
     1     MF1,MF2,MFMIN,MERR
C
C
C   ESTABLISH THE CONVERGENCE TOLERANCES.
C
      WWERR = 0.05
      MERR = 0.001
      PERR = 0.05
      QERR = 0.01
C
```

```
C
C    FIND AN INITIAL "GOOD GUESS" FOR 'MF' GIVEN 'NG' AND 'NS'.
C
          CALL NGNSMF(NG,NS,MF)
C
          MF1 = MF
          XCOUNT = 0.0
C
C
C    CONVERGE ON 'P2' AND 'P4' FOR THE GIVEN 'MF','NG' AND 'NS'.
C
   5      CALL P2P4(NG,NS,MF,PERR,QPERC,P2G,P4G)
C
C
 299      CALL SUBMA(NG,P2G,MA)
          CALL SUBT2(NG,P2G,T2)
          CALL SUBQC(NG,P2G,QC)
          CALL SUBQHT(NG,MA,T2,MF,P4G,QHPT)
          MAF = MA + MF
          CALL SUBT4(NG,MA,T2,MF,P4G,T4)
          CALL SUBQFT(MAF,T4,NS,QFPT)
          QPERC = 100*(QHPT-QC)/QHPT
C
          IF(ABS(QPERC).LT.QERR ) GO TO 300
C
C
C    ESTABLISH UPPER AND LOWER BOUNDS ON 'MF'.
C
          XCOUNT = XCOUNT + 1.0
          X1 = 1.0
          XSIGN = -1.0 * SIGN(X1,QPERC)
          MFDEL = 2.0
          IF(XCOUNT.GT.1.5) GO TO 33
C
          XSIGN1 = XSIGN
          MF2 = MF1 + XSIGN * MFDEL
          MF = MF2
          QPERC1 = QPERC
          GO TO 5
C
  33      IF((XSIGN1*XSIGN).LT.0.5) GO TO 298
C
          MF2 = MF1 + XSIGN1 * MFDEL * XCOUNT
          MF = MF2
C
          GO TO 5
C
C
C    USE THE GOLDEN SECTION METHOD TO FIND THE VALUE OF 'MF' THAT
C    WILL LEAD TO ZERO GAS GENERATOR TORQUE MISMATCH.
C
```

```
  298     QPERC2 = QPERC
C
          IF(MF2.LT.MF1) GO TO 34
C
          MFU = MF2
          MFL = MF1
          QU = ABS(QPERC2)
          QL = ABS(QPERC1)
C
C
   34     MFU = MF1
          MFL = MF2
          QU = ABS(QPERC1)
          QL = ABS(QPERC2)
C
C
C
          TAU = 0.381966
          XN = -2.078 * ALOG(MERR/100.0) + 3
C         WRITE(6,*) 'XN = ', XN
          MF1 = (1.0-TAU)*MFL + TAU*(MFU)
          CALL P2P4(NG,NS,MF1,PERR,QPERC,P2G,P4G)
          Q1 = ABS(QPERC)
          MF2 = (1.0-TAU)*MFU + TAU*(MFL)
          CALL P2P4(NG,NS,MF2,PERR,QPERC,P2G,P4G)
          Q2 = ABS(QPERC)
          XK = 3.0
   90     XK = XK + 1.0
          MFMIN = (MF1 + MF2)/2
C         QDIFF = (Q1 + Q2)/2.0
C         IF(QDIFF.LT.QERR) GO TO 92
          IF(XK.GT.XN) GO TO 92
          IF(Q1.GT.Q2) GO TO 91
          MFU = MF2
          QU = Q2
          MF2 = MF1
          Q2 = Q1
          MF1 = (1.0-TAU)*MFL + TAU*(MFU)
          CALL P2P4(NG,NS,MF1,PERR,QPERC,P2G,P4G)
          Q1 = ABS(QPERC)
          GO TO 90
  91      MFL = MF1
          QL = Q1
          MF1 = MF2
          Q1 = Q2
          MF2 = (1.0-TAU)*MFU + TAU*(MFL)
          CALL P2P4(NG,NS,MF2,PERR,QPERC,P2G,P4G)
          Q2 = ABS(QPERC)
          GO TO 90
   92     MF = MFMIN
```

```
C
C
        CALL SUBMA(NG,P2G,MA)
        CALL SUBT2(NG,P2G,T2)
        CALL SUBQC(NG,P2G,QC)
        CALL SUBQHT(NG,MA,T2,MF,P4G,QHPT)
        MAF = MA + MF
        CALL SUBT4(NG,MA,T2,MF,P4G,T4)
        CALL SUBQFT(MAF,T4,NS,QFPT)
        QPERC = 100*(QHPT-QC)/QHPT
C
C
C
C   EQUATE QD = QFPT FOR STEADY STATE AND SOLVE FOR 'WW' USING
C   NEWTON'S METHOD.  NOTE THAT 'WWERR' HALTS THE ITERATION.
C
C
C   "WW1" IS THE INITIAL GUESS FOR DYNO WATER WEIGHT.  IT IS
C   NEEDED TO START NEWTON'S SCHEME.  THE VALUE IS FAIRLY
C   ARBITRARY BUT DO NOT USE 'WW1 = 0.0'!
C
        WW1 = 5.00
C
C   GG =  QD - QFPT, WHERE QD = FCN(WW1)
C
        C5 = 1.19294E-5
        C3 = 4.0E-6
        C4 = -20.0 + C3*NS*NS
 37     QD = C4 + C5*NS*NS*(WW1**1.3)
        GG = QD - QFPT
        GGP = 1.3*C5*NS*NS*(WW1**0.3)
        WW  = WW1 - GG/GGP
        WWDIFF = 100.0 * ABS((WW  - WW1)/WW1)
        IF(WWDIFF.LT.WWERR) GO TO 300
        WW1 = WW
        GO TO 37
C
C
 300    RETURN
        END
C
C****************************************************************
        SUBROUTINE TORK(NG,NS,MF,WW,DELQG,DELQD)
C****************************************************************
C
C   THE INPUTS TO THIS SUBROUTINE ARE 'NG','NS', AND 'MF'.
C   FOR THESE INPUTS THE SUBROUTINE CONVERGES ON 'P2'AND 'P4'
C   AND FINDS THE "STEADY STATE" TORQUE DIFFERENTIAL, WHICH
C   IS REPRESENTED AS 'DELQG' AND DELQD'.
C
```

```
         REAL NG,NS,MF,MAF,MA,NSO,NGO,MFO,MAFO,MAO,MFDEL,MFU,MFL
C
C    P2G AND P4G ARE NOMINAL VALUES OF COMPRESSOR DISCHARGE AND FPT
C    INLET PRESSURES. THEY PROVIDE AN INITIAL GUESS FOR THE CONVERGENCE
C    ROUTINE. P2ERR AND P4ERR ARE THE MAXIMUM ALLOWABLE DIFFERENCES
C    BETWEEN P2G AND P2 , AND BETWEEN P4G AND P4.
C
C
C
         P2G = 31.5
         P4G = 17.0
         P2ERR = 0.05
         P4ERR = 0.05
         ZS =  50.0
         ZX = 0.0
C
C
C    COMPUTE COMPRESSOR OUTPUTS.
C
  10     ZX = ZX + 1.0
C
         CALL SUBMA(NG,P2G,MA)
         CALL SUBT2(NG,P2G,T2)
         CALL SUBQC(NG,P2G,QC)
C
C
C    COMPUTE P2 AND CHECK AGAINST P2G.
C
 20      CALL SUBP2(NG,MA,T2,MF,P4G,P2)
C
         P2DIFF = 100.0 * ABS(P2 - P2G)/P2
         P2G = P2
C        P2G = P2G + 0.5*(P2-P2G)
         IF(ZX.GT.ZS) GO TO 511
         IF(P2DIFF.GT.P2ERR) GO TO 10
C
511       ZX = 0.0
C
C        COMPUTE HPT OUTPUTS.
C
         MAF = MA + MF
C        WRITE(6,*) 'MAF = ',MAF
C
 47      CALL SUBT4(NG,MA,T2,MF,P4G,T4)
C
C    COMPUTE P4 AND CHECK AGAINST P4G.
C
         CALL SUBP4(MAF,T4,NS,P4)
C
         P4DIFF = 100.0 * ABS(P4 - P4G)/P4
```

```
C
      P4G = P4G + 0.5*(P4-P4G)
C     P4G = P4
      IF(P4DIFF.GT.P4ERR) GO TO 20
C
C   COMPUTE THE TORQUE MISMATCH (QHPT-QC).
C
C
      CALL SUBMA(NG,P2G,MA)
      CALL SUBT2(NG,P2G,T2)
      CALL SUBQC(NG,P2G,QC)
      CALL SUBQHT(NG,MA,T2,MF,P4G,QHPT)
      CALL SUBQFT(MAF,T4,NS,QFPT)
      CALL SUBQD(NS,WW,QD)
      DELQG = QHPT-QC
      DELQD = QFPT-QD
      QPERC = 100*(DELQG)/QHPT
C
C
  919 RETURN
      END
```

APPENDIX E

STATE EQUATION FORMULATION

1. Note that for any variable Y,

$$Y = Yo + \Delta Y$$

where Y = final value
Yo = initial value
ΔY = change in Y

Assume the initial condition, Yo, is known. Then the final state, Y, can be found if ΔY is known.

Applying a Taylor series approximation to Y, where Y is a function of X (i.e. $Y = Y(X)$) yields,

$$Y-Yo = \Delta Y = Yo+Yo'(\Delta X)+Yo''(\Delta X)^2/2!+\ldots Yon/n! .$$

For very small changes in X, a linear approximation can be made:

$$Y = dY = Yo + Yo*dX.$$

As a convenience of notation the following substitution is made:

$$dY = y$$

2. If Y is a function of X (i.e. $Y = Y(X)$), then (making the linear assumption)

$$\partial Y/\partial X = \partial Yo/\partial X + \partial y/\partial X.$$

But $\partial Yo/\partial X = 0$, since Yo is a constant.
Then in general,

$$\partial Y/\partial X = \partial y/\partial X.$$

3. In what follows the state space equation set for the propulsion test facility is derived by using a linear Taylor series approximation for each of the input/output equations used in the nonlinear model.

$$\dot{NG} = (QH-QC)/JG$$

$$QH = QH(Ma,T2,E,NG,P4)$$

$$QC = QC(P2,NG)$$

$$\dot{ng} = (\partial QH/\partial Ma)ma + (\partial QH/\partial T2)t2 + (\partial QH/\partial E)e +$$

$$\ldots(\partial QH/\partial NG)ng + (\partial QH/\partial P4)p4 -$$

$$\ldots(\partial QC/\partial P2)p2 - (\partial QC/\partial NG)ng$$

$$\dot{ng} = a1*ma + a2*t2 + a3*e + a4*p4 + a5*ng -$$

$$\ldots a6*p2 - a7*ng \qquad (1)$$

$$Ma = Ma(P2,NG)$$

$$ma = (\partial Ma/\partial P2)p2 + (\partial Ma/\partial NG)ng$$

$$= b1*p2 + b2*ng \qquad (2)$$

$$P2 = P2(Ma,T2,E,P4,NG)$$

$$p2 = (\partial P2/\partial Ma)ma + (\partial P2/\partial T2)t2 + (\partial P2/\partial E)e +$$

$$\ldots(\partial P2/\partial NG)ng + (\partial P2/\partial P4)p4$$

$$p2 = e1*ma + e2*t2 + e3*e + e4*p4 + e5*ng \qquad (3)$$

$$T2 = T2(P2,NG)$$

$$t2 = (\partial T2/\partial P2)p2 + (\partial T2/\partial NG)ng$$

$$= c1*p2 + c2*ng \qquad (4)$$

substitute (2), (4) into (3),

$$p2 = e1(b1*p2+b2*ng)+e2(c1*p2+c2*ng)+\ldots \qquad (5)$$

$$e3*e+e4*p4*e5*ng$$

$$P4 = P4(Maf,T4,NS)$$

$$p4 = (\partial P4/\partial Maf)maf + (\partial P4/\partial T4)t4 + \ldots$$

$$(\partial P4/\partial NS)ns$$

$$p4 = d1*maf + d2*t4 + d3*ns$$

$$maf = ma + e$$

$$p4 = d1*(ma+e) + d2*t4 + d3*ns \qquad (6)$$

$$T4 = T4(Ma,T2,E,P4,NG)$$

$$t4 = (\partial T4/\partial Ma)ma + (\partial T4/\partial T2)t2 + (\partial T4/\partial E)e +$$

$$\ldots(\partial T4/\partial NG)ng + (\partial T4/\partial P4)p4$$

```
          t4 = f1*ma + f2*t2 + f3*e + f4*p4 + f5*ng    (7)
substitute (6) into (7) and solving for t4,
          t4(1-f4*d2) = ma*(f1+f4*d1) + t2*f2 +        (8)
           ...e*(t3+f4*d1) + f4*d3*ns + f5*ng
let       g1 = (1-f4*d2)
          g2 = (f1+f4*d1)/g1
          g3 = f2/g1
          g4 = (f3+f4*d1)/g1
          g5 = f4*d3/g1
          g6 = f5/g1
then      t4 = g2*ma + g3*t2 + g4*e + g5*ns + g6*ng
grouping terms in (5),
          p2(1-e1*b1-e2*c1) = ng(e1*b2+e2*c2+e5) +...
               e*e3 + e4*p4
let       g7 = (1-e1*b1-e2*c1)
          g8 = (e1*b2+e2*c2+e5)/g7
          g9 = e3/g7
          g10 = e4/g7
then      p2 = g8*ng + g9*e + g10*p4                    (10)
substitute (2), (4) into (1),
          ṅg = a1(b1*p2+b2*ng) + a2(c1*p2+c2*ng) + ...
                    a3*e + a4*p4 + a5*ng - a6*p2 - a7*ng
collecting terms,
          ṅg = p2(a1*b1+a2*c1-a6) + ng(a1*b2+a2*c2+a5-a7)..
               ...a3*e + a4*p4
let       g11 = (a1*b1+a2*c1-a6)
          g12 = (a1*b2+a2*c2+a5-a7)
          g13 = a3
          g14 = a4
then      ṅg = g11*p2 + g12*ng + g13*e+ g14*p4          (11)
substitute (10) into (11) and collecting terms,
          ṅg = ng(g11*g8+g12) + e*(g11*g9+g13) + ...
                    p4*(g11*g10+g14)
let       g15 = (g11*g8+g12)
          g16 = (g11*g9+g13)
```

```
          g17 = (g11*g10+g14)
then      ṅg = g15*ng + g16*e + g17*p4                    (12)
substitute (9) into (6) and collect terms,
          p4 = ma(d1+d2*g2) + e(d1+d2*g4) + t2*d2*g3 +...
                    ns(d2*g5+d3) + ng*d2*g6
let       g18 =  (d1+d2*g2)
          g19 =  (d1+d2*g4)
          g20 =  d2*g3
          g21 =  (d2*g5+d3)
          g22 =  d2*g6
then      p4 = g18*ma + g19*e + g20*t2 + g21*ns...       (13)
                    + g22*ng
substitute (2) and (4) into (13) and collect terms,
          p4 = p2(g18*b1+g20*c1) + ng(g18*b2+g20*c2+g22)...
                 ...+ g19*e + g21*ns
let       g23 =  (g18*b1+g20*c1)
          g24 =  (g18*b2+g20*c2+g22)
          p4 = g23*p2 + g24*ng + g19*e + g21*ns           (14)
substitute (10) into (14) and collect terms,
          p4(1-g23*g10) = ng(g23*g8+g24) + e(g23*g9+g19)...
                 ...+ g21*ns
let       g25 = (1-g23*g10)
          g26 = (g23*g8+g24)/g25
          g27 = (g23*g9+g19)/g25
          g28 = g21/g25
          p4 = g26*ng + g27*e + g28*ns                    (15)
substitute (15) into (12) and collect terms,
          ṅg = (g15+g17*g26) + e(g16+g17*g27) + ns*g17*g28
let       g29 = (g15+g17*g26)
          g30 = (g16+g17*g27)
          g31 = g17*g28
          ṅg = g29*ng + g30*e + g31*ns                    (16)


          Ṅs = (QF-QD)/JD
          QH = QH(NS,Maf,T4)
```

$Q_D = Q_D(NS,WW)$

$\dot{n}s = (\partial QF/\partial NS)ns + (\partial QF/\partial Maf)maf + (\partial QF/\partial T4)t4 -$
$\quad \ldots(\partial QD/\partial NS)ns + (\partial QD/\partial WW)ww$

$\dot{n}s = z1*ns + z2*maf + z3*t4 - z4*ns - z5*ww$

$maf = ma + e$

$$\dot{n}s = z1*ns + z2*ma + z2*e + z3*t4 \ldots - z4*ns - z5*ww \qquad (19)$$

substitute (9) into (19) and collect terms,

$\dot{n}s = ns(z1+z3*g5-z4) + ma(z2+z3*g2) + \ldots$
$\quad e(z2+z3*g4) + t2*z3*g3 + \ldots$
$\quad z3*g6*ng - z5*ww$

let $g32 = (z1+z3*g5-z4)$
$g33 = (z2+z3*g2)$
$g34 = (z2+z3*g4)$
$g35 = t2*z3*g3$
$g36 = z3*g6$

$$\dot{n}s = g32*ns + g33*ma + g34*e + \ldots g35*t2 + g36*ng + g36*ng - z5*ww \qquad (21)$$

substitute (2) and (4) into (21) and collect terms,

$\dot{n}s = ns*g23 + p2(g33*b1+g35*c1) + \ldots$
$\quad ng(g33*b2+g35*c2+g36) + \ldots$
$\quad e*g34 - z5*ww$

let $g37 = (g33*b1+g35*c1)$
$g38 = (g33*b2+g35*c2+g36)$

$$\dot{n}s = g32*ns + g37*p2 + g38*ng + \ldots g34*e - z5*ww \qquad (22)$$

substitute (10) into (22) and collect terms,

$\dot{n}s = ns*g32 + ng(g37*g8+g38) + e(g37*g9+g34)\ldots$
$\quad + p4*g37*g10 - z5*ww$

let $g39 = (g37*g8+g38)$
$g40 = (g37*g9+g34)$
$g41 = p4*g37*g10$

$$\dot{n}s = g32*ns + g39*ng + g40*e + \ldots g41*p4 - z5*ww \qquad (23)$$

substitute (15) into (23) and collect terms,

$$\dot{n}s = ns(g32+g41*g28) + ng(g39+g41*g26) \ldots$$

$$+ e(g40+g41*g27) - z5*ww$$

let $$g39 = (g32+g41*g28)$$

$$g40 = (g39+g41*g26)$$

$$g41 = (g40+g41*g27)$$

$$\dot{n}s = g42*ns + g43*ng + g44*e - z5*ww \qquad (24)$$

$$E = MF/(tS + 1)$$

$$e = -e/t + mf/t \quad (\ldots t = \text{time constant}) \qquad (25)$$

4. Equations (16), (24), and (25) comprise the plant state equations.

LIST OF REFERENCES

1. Miller, R.J., and Hackney, R.D., F100 Multivariable Control System Engine Models/Design Criteria, AFAPL-TR-76-74, 1976 .

2. Merril,W., Lehtinen, B., Zeller, J., The Role of Modern Control Theory in the Design of Controls for Aircraft Turbine Engines, Journal of Guidance and Control, Vol. 7, pp. 652-659, Nov.-Dec. 1984.

3. Johnson, P.N., Marine Propulsion Load Emulation, M.S. Thesis, Naval Postgraduate School, Monterey, California, June 1986.

4. Szuch, J.R., Models for Jet Engine Systems-pt.1 Techniques for Jet Engine System Modelling , Control and Dynamic Systems-Advances in Theory and Applications, Vol. 14, edited by C.T. Leondes, Academic Press, 1978.

5. DeHoff, R.L., and Hall, W.E. Jr., Jet Engine Systems Models-pt.2, State Space Techniques and Modelling for Control, Control and Dynamic Systems-Advances in Theory and Applications, Vol. 14, edited by C.T. Leondes, Academic Press, 1978.

6. Rubis, C.J., and Bodnaruk, A., Acceleration Performance Analysis of a Gas Turbine Destroyer Escort, Journal of Engineering for Power, Jan. 1971.

7. Wylie, C.R., Advanced Engineering Mathematics, McGraw-Hill, 1975.

8. Boeing Airplane Co., Instruction Book for the Boeing Model 502-6a Gas Turbine Engine, Seattle, Washington, 1953.

9. Thompson, W.T., Theory of Vibrations with Applications, Prentice-Hall, 1981.

10. Ogata, K., Modern Control Engineering, Prentice-Hall, 1970.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5000	2
3.	Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	1
4.	Prof. D.L. Smith, Code 69Sm Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	10
5.	Prof. R.H. Nunn, Code 69Nn Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	1
6.	LCDR. G.F. Herda, USN (ret.) P.O. Box 116 Albany, Minnesota 56307	1
7.	LT. Vincent J. Herda, USN Long Beach Naval Shipyard Long Beach, California 90822	5
8.	Mr. M. Resner, Code 56X1 Commander, Naval Sea Systems Command Washington, D.C. 20362	1
9.	Mr. R. Stankey, Code 56Z4 Commander, Naval Sea Systems Command Washington, D.C. 20362	1
10.	Mr. Jim Donnely NAVSSES, Phil. NSYD Philadelphia, Pennsylania 19112-5083	1

219259

Thesis
H48642 Herda
c.1 Marine gas turbine
modeling for modern
control design.