NASA Technical Memorandum 83508

A Generalized Computer Code for Developing Dynamic Gas Turbine Engine Models (DIGTEM)

(NASA-TM-83508) A GENERALIZED COMPUTER CODE FOR DEVELOPING DYNAMIC GAS TURBINE ENGINE MODELS (DIGTEM) (NASA) 18 p HC A02/MF A01 CSCL 21E
N84-12166
Unclas
G3/07 42457

Carl J. Daniele
Lewis Research Center
Cleveland, Ohio

Prepared for the
Aerospace Simulation Conference
sponsored by the Society for Computer Simulation
San Diego, California, February 2-3, 1984

NASA

E-1853

A GENERALIZED COMPUTER CODE FOR DEVELOPING DYNAMIC GAS TURBINE ENGINE MODELS (DIGTEM)

Carl J. Daniele

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

ABSTRACT

This paper describes DIGTEM (digital turbofan engine model) - a computer program that simulates two-spool, two-stream (turbofan) engines. DIGTEM was developed to support the development of a real-time multiprocessor-based engine simulator being designed at the Lewis Research Center. The turbofan engine model in DIGTEM contains steady-state performance maps for all the components and has control volumes where continuity and energy balances are maintained. Rotor dynamics and duct momentum dynamics are also included.

DIGTEM features an implicit integration scheme for integrating stiff systems and "trims" the model equations to match a prescribed design point by calculating correction coefficients that balance out the dynamic equations. It uses the same coefficients at off-design points and iterates to a balanced engine condition.

Transients are generated by defining the engine inputs as functions of time in a user-written subroutine (TMRSP). Closed-loop controls can also be simulated.

DIGTEM is generalized in the aerothermodynamic treatment of components. This feature along with DIGTEM's "trimming" at a design point make it a very useful tool for developing a model of a specific turbofan engine. Also, subsets of the turbofan engine configuration such as turbojet or a turboshaft can be simulated with minor modifications to the Fortran coding. With extensive modifications to the coding, arbitrary configurations can be modeled.

INTRODUCTION

The development and performance verification of controls for modern jet engines require accurate real-time simulations. To obtain the accuracy, the simulations must contain full range, thermodynamic representations of all engine processes. Currently available general-purpose simulators, such as hybrid computers, offer the necessary real-time capabilities but require large initial investments in both money and time and highly trained specialists for the operation and support of the simulations. An alternative approach is to use simplified process models on less costly simulators. These models are necessarily less accurate and therefore result in lower confidence in the simulation results.

The NASA Lewis Research Center is currently exploring new techniques for real-time engine simulation of jet engines. The work will lead to the development of a prototype, real-time digital multiprocessor system that provides the performance, versatility, and usefulness of a hybrid computer system at a fraction of the cost. The effort will focus on the use of high-speed microprocessors (e.g., Motorola 68000) and parallel processing to obtain the required computing speeds for real-time simulation. One of the keys to the development of the simulator is the development of software tools to reduce the programming complexity and number of steps required to go from specifications of the process to a working simulation.

To guide the hardware and software development, an analytical effort is being undertaken to study methods of partitioning engine models for parallel solution. The effort is two pronged. Pratt & Whitney Aircraft is under contract to Lewis to study model partitioning algorithms for parallel processing. That effort focuses on an implicit integration scheme that has been used for real-time engine simulation on a single processor (ref. 1). This paper describes an in-house effort to develop a generalized computer program for developing dynamic engine simulations and also for studying parallel processing and numerical integration techniques for real-time applications.

Many generalized digital engine simulations exist today. Most are limited to steady-state performance calculations for a fixed number of engine configurations (refs. 2 to 5). One generalized code, NNEP (ref. 6) lets the user build arbitrary configurations through input definitions. Another, DYNGEN (ref. 7), has transient capability but is limited to the fixed engine configurations of references 4 and 5 (GENENG and GENENG II). All the generalized codes described are limited to steady-state calculations or they have fixed engine configurations. Some (DYNGEN and GENENG) are difficult to change, and none has the capability of scaling its model equations to reflect real engine data. Thus, none of the available generalized digital codes satisfy all the requirements for dynamic real-time simulation development. Further, none of the available codes are suitable for multiprocessor-based simulations.

A generalized dynamic engine simulation has been developed for the hybrid computer. That program is called HYDES (ref. 8). The HYDES program led to the development of a systematic, computer-aided approach for generating hybrid computer simulations of a particular class of engine (i.e., two-spool, two-stream turbofan; refs. 9 and 10). This approach featured generalized aero-thermodynamic (variable gas properties) models of engine components and automated calculation of scale factors and simulation coefficients. Also, a specified operating point, designated as the design point, was used to scale the component maps and to determine correction coefficients that would balance the dynamic equations at the design point. This assured good steady-state accuracy at the design point. Thus, the hybrid model possessed many of the capabilities needed for the development of dynamic turbofan engine models.

Recently, an all digital computer model possessing the capabilities of the hybrid model has been developed and is the subject of this paper. The resultant digital computer code is called DIGTEM. A complete description of DIEGEM is presented in reference 11, including a complete description of the turbofan engine model, a users manual, a test case, and flow charts.

DIGTEM is generalized in a different sense than DYNGEN. DIGTEM, while having only one engine configuration in the code, is written in modular form to permit variations of the engine configuration (e.g., turbojets and turboshafts) to be simulated. This provides more flexibility (at the cost of recording the FORTRAN) than DYNGEN which is limited to a fixed set of configurations and which is difficult to change. Both DYNGEN and DIGTEM do component map scaling to match input data at a design point. DIGTEM, however, also calculates correction coefficients to balance the dynamic equations so that a steady-state balance at the design point is generated. The same values of the correction coefficients are used at off-design points. If the coefficients do not balance the dynamic equations at the operating points, DIGTEM iterates to a new balanced engine condition. DIGTEM's modular structure and its flexibility should allow it to be a useful tool for engine dynamics studies and controls analyses for parallel processing.

ENGINE DESCRIPTION

The engine model supplied with DIGTEM represents a two-spool, two-stream augmented turbofan engine (fig. 1). A single inlet supplies airflow to the fan. Air leaving the fan is separated into two streams - one passing through the engine core and another through an annular bypass duct. The fan is driven by a low-pressure turbine. The core airflow passess through a compressor driven by a high-pressure turbine. Both the fan and compressor are assumed to have variable geometry to improve stability at low speed. Engine bleed air is extracted at the compressor exit and used for turbine cooling and for accessory drives. Fuel is input to the main combustor and burned to produce hot gas for driving the turbines. The engine core and bypass streams combine in an augmentor duct, where additional fuel is added to further increase the gas temperature (and thus thrust). The augmentor flow is discharged through a variable convergent-divergent nozzle. The nozzle throat area (station 8) and exhaust nozzle area (station E) are varied to maintain engine airflow and to minimize drag during augmentor operation.

MODEL DESCRIPTION

Figure 2 is a computational flow diagram of the engine model. The analytical model includes multivariate maps which model the steady-state performance of the engine's rotating components. Bleed flows are modeled by assuming that some bleed air will be used for turbine cooling (flow returning to the cycle) and some for accessory drives (flow lost to the cycle). Fluid momentum in the bypass duct and augmentor, mass and energy storage within the control volumes, and rotor inertias are included in the model to provide transient capability.

A typical engine model will have time constants that differ by three or four orders of magnitude and thus represents what is termed a "stiff" system. This requires the use of very small time steps when using explicit integration routines. The integration technique featured in DIGTEM is an implicit integration scheme which is well suited for integrating stiff systems. The scheme uses a multivariable Newton-Raphson iteration method for convergence at each time point. In DIGTEM there are 16 iteration variables corresponding to the 16 dynamic (state) equations. Although DIGTEM features implicit integration, it also has the capability to use explicit integration methods. This was done so that DIGTEM could be used to investigate numerical integration schemes for turbofan engine modeling.

DIGTEM STRUCTURE

DIGTEM's structure is shown in figure 3. In the main routine DIGTEM, printout interval integration stepsize, system order, desired operating point, transient duration, and integration method are specified. Then the input data file, which contains normalized maps of fan and compressor flow shifts due to variable geometry, normalized fan, compressor, high pressure turbine, and low pressure turbine, is read. Component maps are normalized to values at the desired design operating point. Thus, operating point data are provided for engine variables at each operating point. Data for five typical operating points are supplied with DIGTEM: three nonafterburning and two afterburning points. The supplied data includes pressures, temperatures, and flows throughout the engine, rotor speed, turbine enthalpy drops, component efficiencies, and engine geometries. The flow data for the compressor, burner, and turbines are used to specify the cooling bleed and the overboard bleed flows. In DIGTEM the first nonafterburning and first afterburning operating points supplied are design points. The others are off-design operating points.

Once the data are read in, DIGTEM calculates the correction coefficients to scale the model equations to the design operating point. This is always done even if the operating point selected in the main routine is off-design. Then using the correction coefficients and the selected operating point data, the engine routines are called to calculate the engine performance. Next, DIGTEM calls the implicit or explicit integration routines as selected by the user. In steady-state, which is specified by setting the transient duration to zero, the implicit integration method will iterate if needed so that all the state derivatives are zero. If the explicit integration method is used, a transient must be run to drive the errors to zero (the engine inputs must remain constant). If a transient is selected (transient duration greater than zero), the implicit integration method will first iterate to a balanced initial condition. A transient is run by specifying engine inputs as functions of time in a user-written subroutine TMRSP. Engine inputs for the DIGTEM model are main burner fuel flow $\dot{w}_{F,4}$, afterburner fuel flow $\dot{w}_{F,7}$, nozzle throat area A_8, fan variable geometry parameter (FVGP), and compressor variable geometry parameter (CVGP). By making a minor change in the FORTRAN coding, transients due to changes in inlet pressure and temperature can also be run. Once the engine inputs are specified in TMRSP, the engine routines are called to calculate the engine response to the change in inputs. Time is then incremented and, if time is less than the specified transient duration, the control inputs are updated again. This procedure continues until time is greater than the desired transient duration.

Although DIGTEM is set up to satisfy engine inputs as a function of time, it can be used to simulate closed-loop controls. This can be accomplished by integrating the controls with the state variables or by using control routines in DYNGEN which were written to be compatible with the implicit formulation.

Example of DIGTEM Transient - Test Case

Figure 4 shows time histories of the engine inputs for a typical engine acceleration from a low-power operating point to a high-power (augmented) operating point. Main burner fuel flow $\dot{w}_{F,4}$ is ramped in 2 sec from 0.17 to 0.77 kg/sec (0.37 to 1.7 lbm/sec). FVGP and CVGP are varied in a manner designed to stay within the ranges of the fan and compressor flow shift maps. After 10 sec, afterburning is initiated, and $\dot{w}_{F,7}$ is ramped in 3 sec from 2.27 kg/sec (5.0 lbm/sec). Also, at time equal 10 sec, the nozzle throat area A_8 and the exhaust nozzle area A_E are ramped to their new operating point values (also in 3 sec). The values for the engine inputs were selected to match the steady-state operating point values in DIGTEM. Figure 5 shows the simulated turbofan engine variables for the test case. Shown are plots of high rotor speed N_H, low rotor speed N_L, burner pressure P_3, turbine-inlet temperature T_4, and augmentor temperature T_7 versus time. All five variables increased smoothly to their new values and then held constant until augmentor fuel flow was added at $t = 10$ sec. Note that all engine variables stayed constant during afterburning except the augmentor temperature, which increased smoothly to its final value. For the 20-sec transient shown, the integration stepsize was 0.01 sec, and the CPU time was 13.1 sec on the IBM 370/3033 computer.

SIMULATION OF OTHER CONFIGURATIONS

DIGTEM contains normalized component maps and a generalized aerothermodynamic treatment of its components. It also has the capability for scaling the analytical model to match a user-specified design point. These features make it useful for simulating turbofan engines other than the one described in DIGTEM. Also, with minimal Fortran reprogramming, variations from a turbofan engine such as a turbojet or turboshaft engine can be simulated. It is possible to model arbitrary engine configurations; however, major modifications to the coding would be required.

Turboshaft Engine Model

To simulate an engine such as a turboshaft, the user need only delete those areas of code that are not needed (by comparing the engine computational flow diagram with fig. 2) and equate variables where needed. The order of the state variables has been set to facilitate the required modifications to the implicit integration.

To demonstrate this capability, a turboshaft engine model was implemented by using DIGTEM. A computational flow diagram of the engine is shown in figure 6. A comparison with the turbofan engine computational flow diagram of DIGTEM in figure 2 indicates the need to make the following changes in DIGTEM: The inlet, fan, duct, augmentor, and nozzle must be eliminated; the low pressure turbine must be disconnected from the fan and connected to the load; and the back pressure on the power turbine must be fixed (at atmospheric pressure) with turbine flow (and energy) dumped to the atmosphere. The turboshaft engine model was implemented in DIGTEM by using the normalized maps already in DIGTEM and by specifying a new design point. The following changes in the Fortran coding will cause the structure of figure 2 to look like figure 6:

$$P_{2.2} = P_{13} = P_2 = P_0$$

$$T_{2.2} = T_{13} = T_0$$

$$\dot{w}_{13} = 0$$

$$\dot{w}_7 = \dot{w}_6 = \dot{w}_{4.1}$$

$$T_7 = T_6$$

$$\dot{w}_{BLLT} = 0$$

$$P_6 = P_0$$

$$FVGP = 0$$

$$CVGP = 0$$

Eight state variables are needed for the turboshaft engine. These correspond to the first eight state variables for the turbofan engine. Thus, to reflect a change in engine states, the number of states, N, defined in the main program must be changed from 16 to 8. Also, recoding must be done in subroutine TMRSP where the inputs to the model are specified as functions of time. For the turboshaft engine the inputs are fuel flow $\dot{w}_{F,4}$ to the main burner and load torque Q_{load} change on the power turbine. TMRSP was set up to give a step change in both fuel flow and load torque.

Figure 7 shows the transient response of the turboshaft engine to simultaneous steps in fuel flow and load torque. Shown are normalized values of fuel flow, load torque, low rotor speed, high rotor speed, burner pressure, and turbine inlet temperature. Note that N_H, P_3, and T_4 all increase with addition of fuel. Normally, N_L would increase also, but the increase in load caused N_L to drop off. For this 2-sec transient, the integration stepsize was 0.01 sec. The CPU time was 1.06 sec.

Turbojet Engine Model

For particular engine configurations, some change to the state variable order may be necessary. For example, one may wish to simulate a single-spool turbojet such as the one shown in figure 8. In comparing this configuration with the turbofan configuration of figure 2, it is clear the fan duct, fan, and low turbine must be eliminated. This results in a state variable order different from that provided with DIGTEM.

Variables must be eliminated or equated in the engine code and input data as follows:

$$\dot{w}_{BLLT} = 0$$

$$FVGP = 0$$

$$CVGP = 0$$

$$\dot{w}_{13} = 0$$

$$\dot{w}_{2.2} = \dot{w}_2$$

$$P_{2.2} = P_2$$

$$T_{2.2} = T_2$$

$$P_6 = P_{4.1}$$

$$T_6 = T_{4.1}$$

In the main routine DIGTEM the number of state variables must be reduced to 10. The FORTRAN recoding to accomplish the variable changes and state variable reordering is done in the appropriate subroutines as specified in reference 11.

Thus, it is possible to use DIGTEM to model engines other than a two-spool, two-stream, turbofan engine. The resultant engine model will have a realistic aerothermodynamic treatment of its components and will be scaled to a user-specified design point.

CONCLUDING REMARKS

The development of digital engine controls, integrated propulsion flight controls, and engine diagnostic systems can be facilitated by the use of real-time engine simulations. While both hybrid and digital techniques are currently being used for real-time simulation, the trend appears to be toward all digital approaches because of the availability of low-cost digital hardware, powerful software tools, and well-trained programming personnel. The emergence of microprocessors promises to make possible the development of simulation-oriented, parallel processing systems that will allow the implementation of high fidelity, real-time engine simulations in a cost-effective manner.

Work is currently being carried on at Lewis Research Center to develop the hardware and software for a low-cost, parallel processing system. DIGTEM, the generalized turbofan-engine computer code described in this paper, was developed in response to a need for a tool for studying numerical integration techniques for real-time applications and for developing dynamic engine simulations that can be run on a parallel processing system.

So far, DIGTEM has been used to model both turbofan and turboshaft engines. These models will be programmed to run on a parallel processor system. Eventually, DIGTEM will be programmed to run directly on the parallel processing system in real-time. Because the model and data in DIGTEM are not proprietary, NASA plans to make DIGTEM available to industry and university researchers involved in NASA-supported programs.

SYMBOLS

A	cross sectional area, cm^2 (in^2)
ALT	altitude, m (ft)
CVGP	compressor variable geometry parameter, deg
F	thrust, N (lbf)
FVGP	fan variable geometry parameter, deg
h	specific enthalpy, J/kg (Btu/lbm)
M	Mach number
N	rotor speed, rpm
P	total pressure, N/cm^2 (psia)
Q	torque, cm N (lbf)
T	total temperature, K (°R)
$\dot{w}$	mass flow rate, kg/sec (lbm/sec)
	efficiency

Subscripts:

A	air
AB	afterburner
am	ambient
BLHT	high pressure turbine cooling bleed
BLLT	low pressure turbine cooling bleed
BLOV	overboard bleed
E	exit
F	fuel
H	high pressure spool
HT	high pressure turbine
ID	fan hub region
j	station j=0, 2, 2.1, 2.2, 3, 4, 4.1, 5, 6, 7, 8, 13, 16
L	low pressure spool
LOAD	LOAD
LT	low turbine
OD	fan tip region

REFERENCES

1. McLaughlin, Peter: A Technique for the Implementation of Nonlinear Models as Real-Time Digital Simulations - With Application to Aircraft Turbine Engine Design. Proc. of the 1980 Summer Computer Simulation Conference, Seattle, Wash., AFIPS Press, 1980, pp. 111-115.
2. McKinney, John S.: Simulation of Turbofan Engine. Part I: Description of Method and Balancing Technique. AFAPL-TR-67-125-Pt.-1, Air Force Aeropropulsion Lab, 1967 (AD-825197).
3. McKinney, John S.: Simulation of Turbofan Engine. Part II: User's Manual and Computer Program Listing. AFAPL-TR-125-Pt.-2, Air Force Aeropropulsion Lab, 1967 (AD-825198).
4. Koenig, Robert W.; and Fishbach, Lawrence H.: GENENG: A Program for Calculating Design and Off-Design Performance for Turbojet and Turbofan Engines. NASA TN D-6552, 1972.
5. Fishbach, Lawrence H.; and Koenig, Robert W.: GENENG II: A Program for Calculating Design and Off-Design Performance of Two- and Three-Spool Turbofans with as Many as Three Nozzles. NASA TN D-6553, 1972.
6. Fishbach, Lawrence H.; and Caddy, Michael J.: NNEP: The Navy NASA Engine Program. NASA TM X-71857, 1975.
7. Sellers, James F.; and Daniele, Carl J.: DYNGEN: A Program for Calculating Steady-State and Transient Performance of Turbojet and Turbofan Engines. NASA TN D-7901, 1975.
8. Szuch, John R.: HYDES: A Generalized Hybrid Computer Program for Studying Turbojet or Turbofan Engine Dynamics. NASA TM X-3014, 1974.
9. Szuch, John R.; Krosel, Susan M.; and Bruton, William M.: Automated Procedure for Developing Hybrid Computer Simulations of Turbofan Engines. Part I: General Description. NASA TP-1851, 1982.
10. Szuch, John R.; Krosel, Susan M.; and Bruton, William M.. Automated Procedure for Developing Hybrid Computer Simulations of Turbofan Engines. Part II: Computer Printouts. NASA TM-81730, 1981.
11. Daniele, Carl J.; Krosel, Susan M.; Szuch, John R.; and Westerkamp, Edward J.: Digital Computer Program for Generating Dynamic Turbofan Engine Models (DIGTEM). NASA TM-83446, 1983.

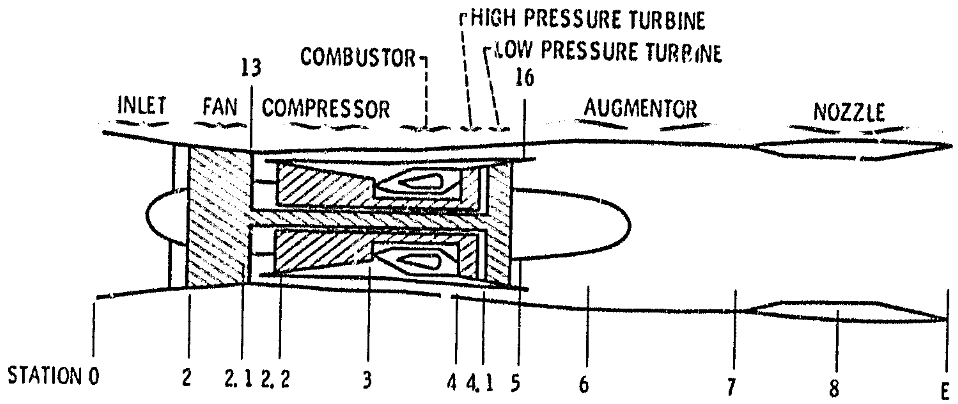

Figure 1. - Schematic of augmented turbofan engine.

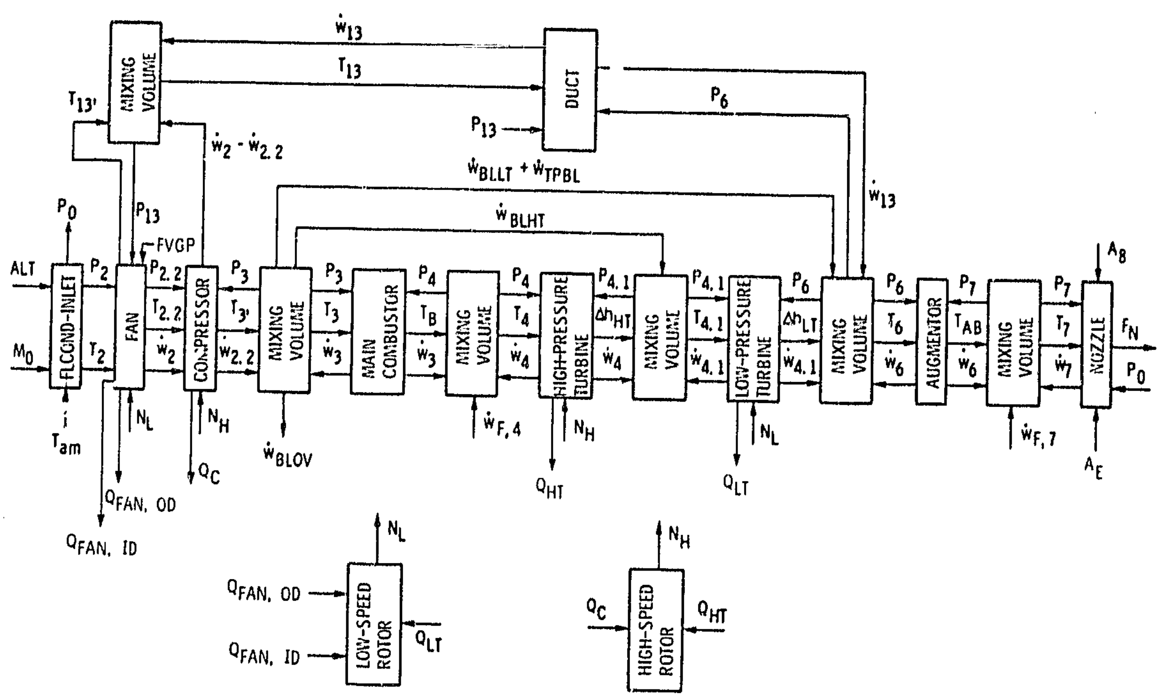

Figure 2. - Computational flow diagram of augmented turbofan engine simulation.

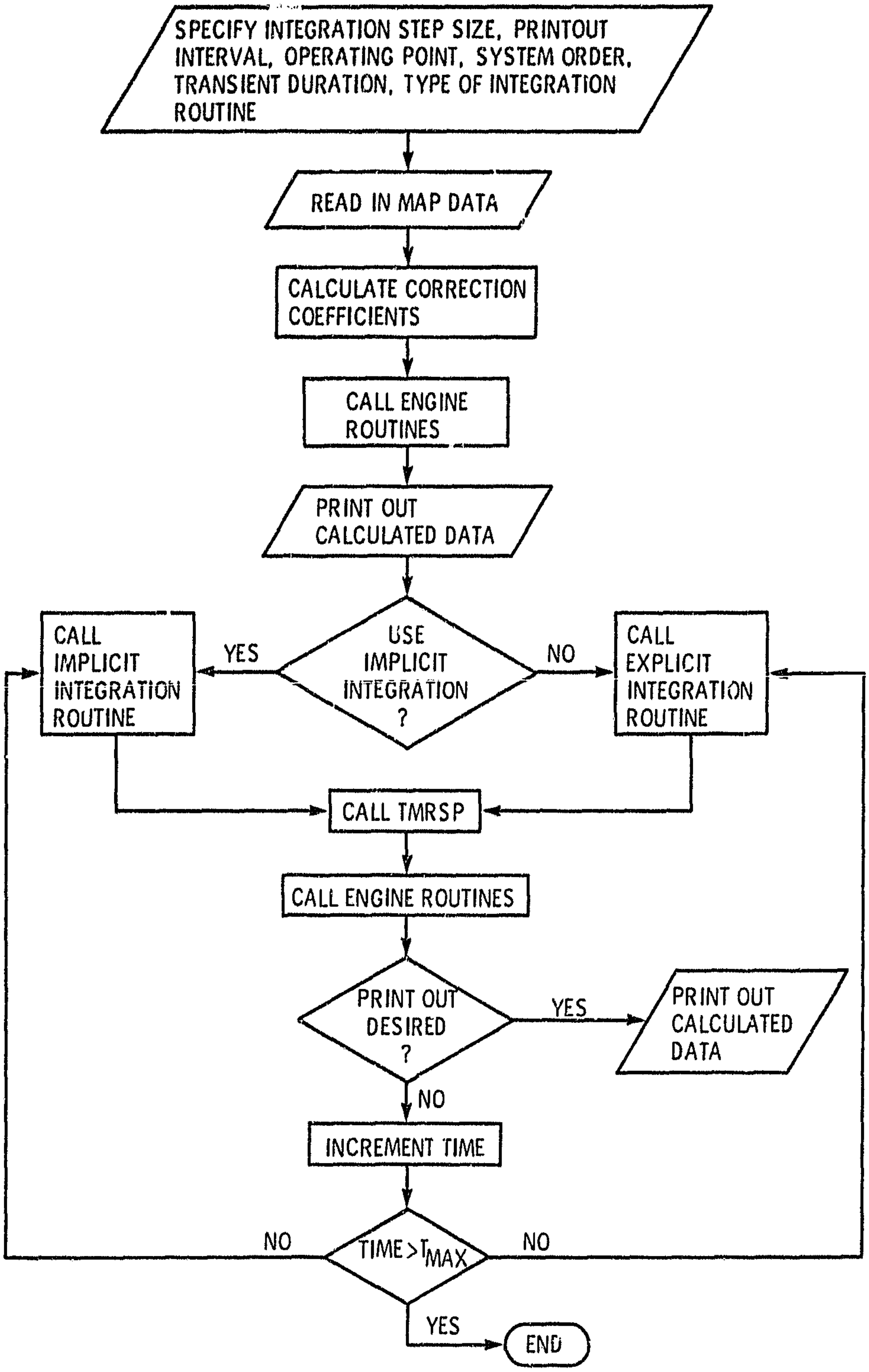

Figure 3. - DIGTEM structure.

ORIGINAL PAGE IS
OF POOR QUALITY

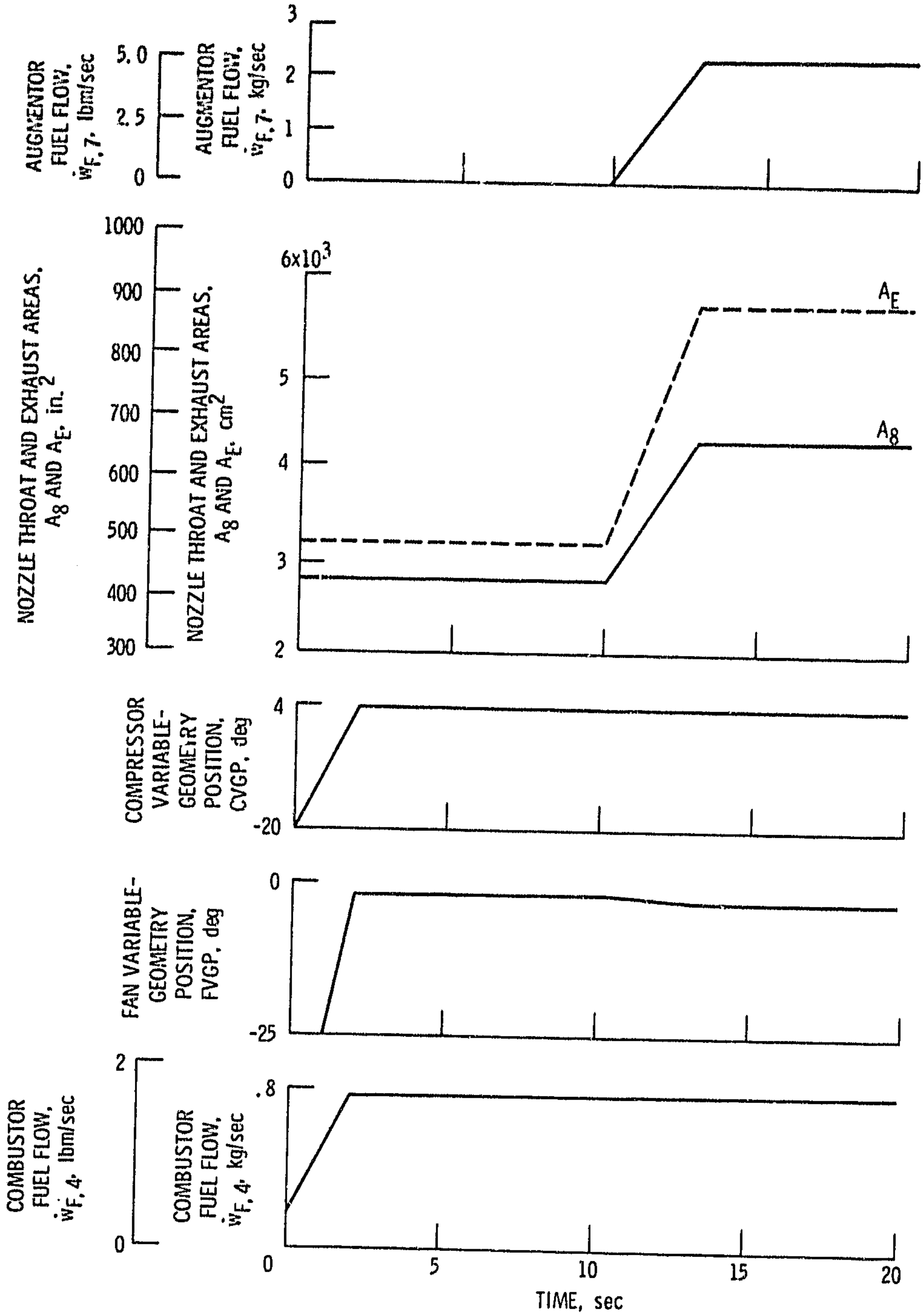

Figure 4. - Engine inputs for DIGTEM Test case.

ORIGINAL PAGE IS
OF POOR QUALITY

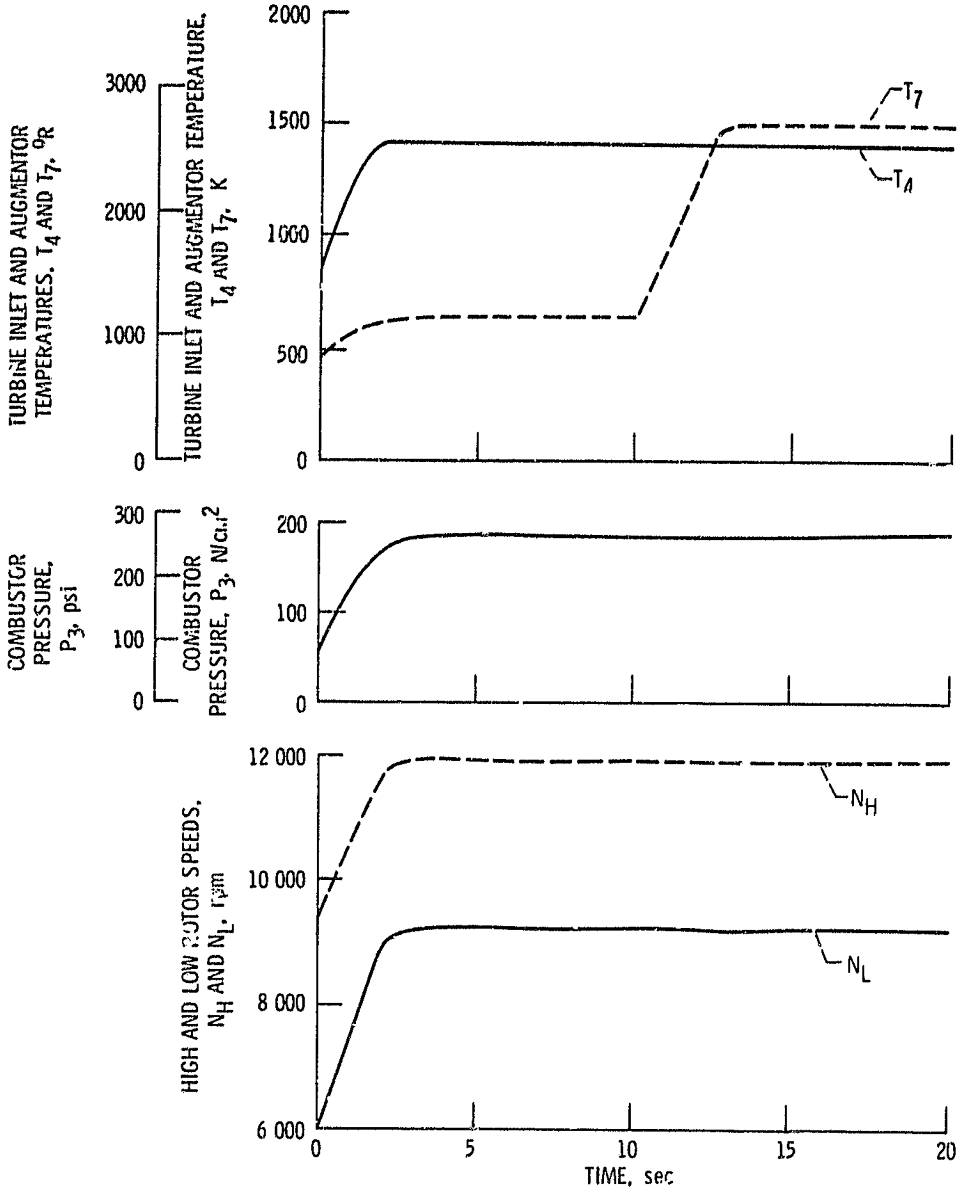

Figure 5. - Turbofan engine response for test case.

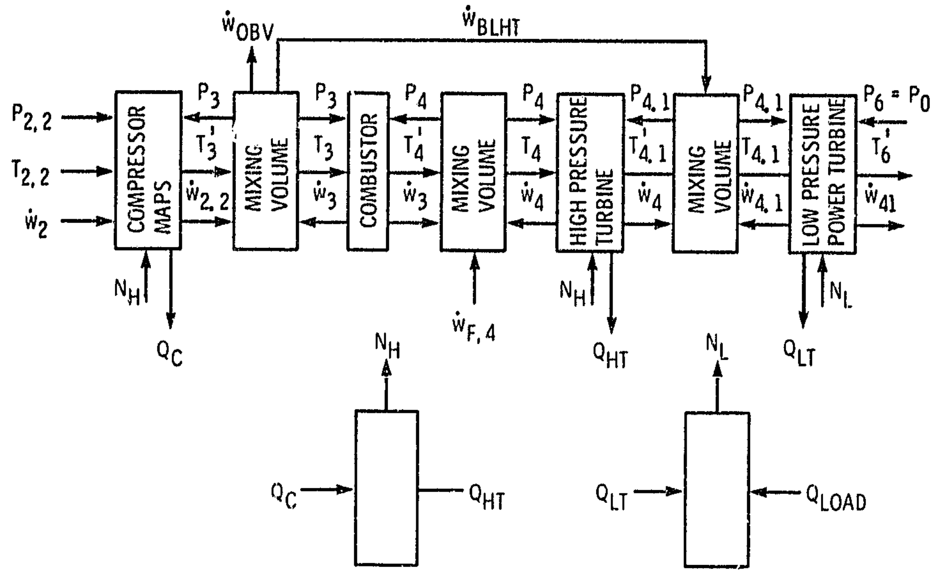

Figure 6. - Computational flow diagram of turboshaft engine.

ORIGINAL PAGE IS
OF POOR QUALITY

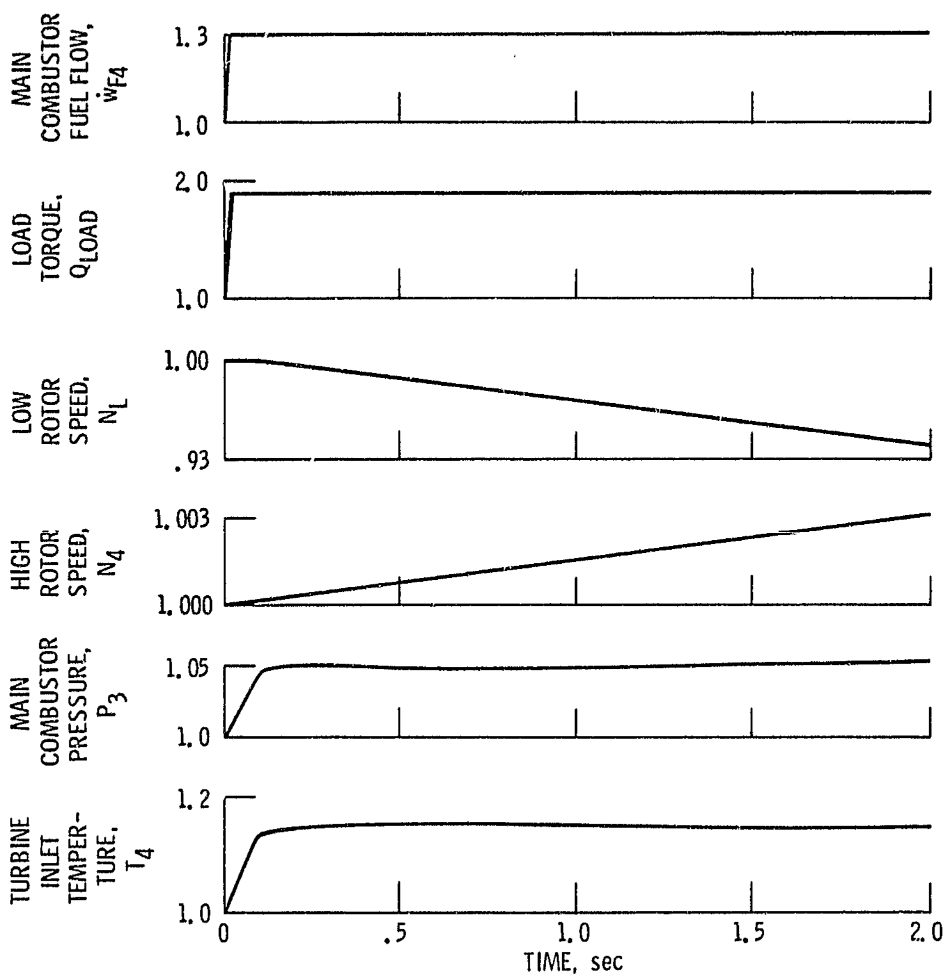

Figure 7. - Transient response of a simulated turboshaft engine to simultaneous steps in fuel flow and load. Data are normalized to design point values.

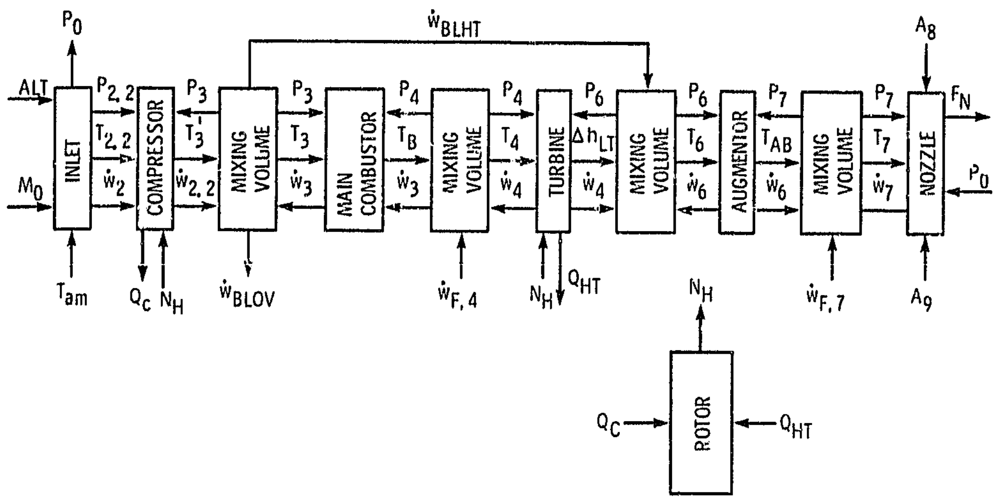

Figure 8. - Computational flow diagram of a turbojet engine.