NASA Technical Memorandum 8160 5

(NASA-TM-81605) AN AUTOMATED PROCEDURE FOR DEVELOPING HYBRID COMPUTER SIMULATIONS OF TURBOFAN ENGINES (NASA) 19 p HC A02/MF A01 CSCL 09B
N81-11688
Unclas
G3/61 29081

An Automated Procedure for Developing Hybrid Computer Simulations of Turbofan Engines

John R. Szuch, Susan M. Krosel, and William M. Bruton
Lewis Research Center
Cleveland, Ohio

Prepared for the
Fourteenth Annual Simulation Symposium
Tampa, Florida, March 18-20, 1981

NASA

AN AUTOMATED PROCEDURE FOR DEVELOPING HYBRID COMPUTER SIMULATIONS OF TURBOFAN ENGINES

John R. Szuch, Susan M. Krosel, and William M. Bruton
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Abstract. This paper offers a systematic, computer-aided, self-documenting methodology for developing hybrid computer simulations of turbofan engines. The methodology that is presented makes use of a host program that can run on a large digital computer and a machine-dependent target (hybrid) program. The host program performs all of the calculations and data manipulations that are needed to transform user-supplied engine design information to a form suitable for the hybrid computer. The host program also trims the self-contained engine model to match specified design point information. A test case is described and comparisons between hybrid simulation and specified engine performance data are presented.

INTRODUCTION

The development of aircraft propulsion systems depends, to a great extent, on one being able to predict the performance of the propulsion system and its associated controls. Computer simulations provide the means for analyzing the behavior and interactions of these increasingly complex systems prior to building and testing expensive hardware.

The hybrid (analog and digital) computer offers the opportunity to combine the best features of digital and analog computation to satisfy the stringent requirements of engine simulation. One can achieve the desired steady-state and dynamic accuracies with reasonable solution times and the user is provided "hands-on" interactive control of the simulation with convenient display and recording of simulation results.

Hybrid computer simulations of the Pratt and Whitney (PWA) TF30-P-3 and F100-PW-100 turbofan engines have been previously developed (References 1 to 3) and used to support the development of advanced electronic engine controls (References 4 to 7) at the Lewis Research Center. In the control applications, the engine simulations served as "test-beds" for evaluating new control laws and for verifying control software prior to engine testing.

Unfortunately, the development of accurate hybrid computer simulations has been viewed as an "art" which requires specialists experienced in dynamic system modeling, computer programming, and computer operations. There are, in fact, a number of problems that must be dealt with when developing hybrid computer simulations (see Figure 1). These include (1) formulation of a mathematical model that is detailed enough for the particular application yet doesn't consume excessive computing equipment or computing time, (2) the precalculation and data manipulation that are necessary to transform engine design information into a form suitable for the hybrid computer, (3) the implementation of the simulation on the hybrid computer, (4) the evaluation of the simulation model relative to available design and off-design engine data, (5) the modification of the simulation, if necessary, to match the reference data, (6) the documentation of the simulation and its development to facilitate changes/extensions of the simulation design as requirements and applications change, and (7) repetition of the development process for each new engine to be studied.

This paper addresses these problems and offers a systematic, computer-aided, self-documenting methodology for the development of a hybrid computer simulation of a turbofan engine. This methodology represents an extension of earlier work (Reference 8) aimed at developing a generalized engine simulation. However, the present paper focuses on the turbofan engine configuration and concentrates on the automation and documentation of the simulation de.elopment process. Nonetheless, the concepts and computer codes that have been developed to do this are sufficiently general to permit their adaptation to simulations of other engine types.

This paper describes the turbofan engine model and the approach used to develop the hybrid computer simulation. The organization and functions of the associated computer programs are also discussed. A test-case (a representative turbofan engine operating at sea-level, static conditions) is used to illustrate the simulation development process. Typical steady-state and transient results from the hybrid computer simulation are presented.

SIMULATION APPROACH

The proposed simulation development process is illustrated in Figure 2. A host digital program, written in Portran, runs on a suitable digital computer (in our case, the IBM 370/3033). The host program performs all of the aforementioned precalculations and data manipulations (including scaling of variables) using user-supplied engine design information. The host program then trims the self-contained engine model to match the user-supplied design-point performance data and, in addition, provides a quantitative measure of each engine component model's accuracy relative to supplied off-design point data. This permits off-line evaluation and refinement of the models using the host program, prior to implementation and operation of the hybrid simulation. Finally the host program provides a printout of all unscaled/scaled engine parameters and punched cards containing the hybrid computer set-up information.

The target (hybrid) simulation is, of course, machine dependent. As presented, it runs on the Electronic Associates, Inc. (EAI) PACER 600 hybrid computer system. That system includes a 32K 16-bit digital processor, a 10-volt analog processor, and an interface unit that provides communication between the analog and digital machines. The target simulation consists of Scaled-Fraction Fortran and assembly language routines that run on the PACER digital computer and a prescribed analog patching arrangement. The digital routines (1) set up the analog computer components, including setting of potentiometers, (2) compute state variable derivatives using an engine model identical to the one contained in the host program, and (3) provide printouts of selected steady-state simulation data. The analog computer is used, primarily, for continuous integration of the state-variable derivatives. Twenty-five analog-to-digital converters (ADC's) and twenty-one digital-to-analog converters (DAC's) are used to transfer information between the analog and digital processors. Strip-chart recorders are used to monitor and record the transient behavior of the simulation.

ENGINE MODEL

Figure 3 contains a schematic representation of the two-spool, augmented turbofan engine to be simulated. A single inlet is used to supply airflow to the fan. Air leaving the fan is separated into two flow streams: one stream passes through the engine core while the other stream passes through an annular bypass duct. The fan is driven by a low-pressure turbine. The core airflow passes through a compressor which is driven by a high-pressure-turbine. Both the fan and compressor are assumed to have variable geometry to improve aerodynamic stability at low rotational speeds. Engine airflow bleeds are also provided at the compressor exit (station 3) for turbine cooling (flow returned to the cycle) and accessory drives (flow lost to the cycle). Fuel flow is injected into 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 (hence, thrust). The augmentor flow is discharged through a variable convergent-divergent nozzle. The nozzle throat area (station 8) and exhaust area (station E) can be varied to maintain engine airflow and to minimize drag during augmentor operation. Figure 4 contains a computational flow diagram of the engine model. All symbols are defined in the Symbol List. Wide-range, overall performance maps are used to provide accurate steady-state representations of the engine's rotating components. The effects of the variable fan and compressor geometry on the components' performance are accounted for in the model. Factors such as fluid momentum, mass and energy storage, and rotor inertias are included to provide transient capability.

HYBRID COMPUTER PROGRAM

Analog Program

The computational split between the analog and digital portions of the hybrid simulation is weighted heavily toward the digital. That is, the analog computer is limited to doing continuous integration with respect to time and some related multiplication/division. By comparison, the real-time simulations developed earlier (References 1 to 3) used extensive analog computation with the digital limited to performing bivariate function generation. The decision to make more use of the digital computer was based on the desire to improve the steady-state accuracy of the simulation and to facilitate the automation of the simulation development process.

Scaled values for the engine state variables (stored masses, temperatures, duct flow rates, and rotor speeds) and the inter-component pressures are computed on the analog computer. Figure 5 illustrates the analog calculation of variables in an intercomponent volume. Identical circuitry is patched by the user for each of the six intercomponent volumes. Inputs to the analog from the digital are transmitted by 21 DAC's. These include the scaled stored mass derivatives, DWj, and scaled non-specific temperature derivatives, DTQWj, as shown in Figure 5. Outputs from the analog to the digital are transmitted by 25 ADC's. These include the scaled intercomponent pressures and temperatures. Similarly, blocks of pre-designed circuitry are patched by the user for each of the two ducts and two spools in the model.

The analog computer is also used to generate (or transmit from external sources) inputs to the simulation. These include scaled control inputs (fuel flow, nozzle area, etc.) and scaled flight condition inputs (altitude, Mach number, and sea-level ambient temperature).

Having selected and patched the analog components, the user can input the addresses of the integrators and potentiometers to the host digital program and have the host program automatically compute the integrator gains and potentiometer settings. Punched cards containing these values will be generated by the host program and used by the target digital program to set up the analog components (via the hybrid interface).

Digital Program

The target digital processor is used to perform the bulk of the computations in the hybrid computer simulation. That is, the digital does all of the arithmetic and function generation necessary to compute the time derivatives of the engine state variables. In addition, variables of interest, such as net thrust, are also computed in the digital and output to the analog for display and recording. The digital computer also provides steady-state data displays and automated set-up of the analog computer.

Figure 6 shows the structure and flow of the target digital program. The program consists of a number of subroutines running under control of a main program called EXEC1 (References 9 and 10). The major subroutines are INITAL, an initialization and set-up routine, and LOOP, the digital portion of the dynamic engine simulation. INITAL is executed once prior to entering the main dynamic loop. LOOP represents a Scaled-Fraction Fortran implementation of the steady-state engine model contained in the host program. Scaled versions of the equations defining the state variable derivatives are repetitively solved in LOOP at a fixed rate selected by the user on the basis of (1) the time required to perform the calculations in LOOP, and (2) the maximum allowable digital delay for stable operation of the hybrid simulation.

The EXEC1 program provides a time-shared, interrupt environment in which the user can interactively control steady-state and dynamic displays of digital data. The subroutine INFORM is called by EXEC1 when the user depresses a sense switch (F) at the computer console and if the computations in LOOP have been completed. If spare time is available, the user can obtain displays of simulation data while the program is running. The user can assign alphanumeric names and scale factors to selected memory locations and obtain, upon command, listings of simllation data in engineering units. This method of obtaining steady-state results was used in lieu of WRITE statements in the target program. Finally, the subroutine LEVEL8 is used to obtain map and function out-of-range information during simulation runs. Table I lists the significant statistics for the target program. The core requirements of the basic target program are only about 7.5 K words. However, the addition of the initialization and set-up routine INITAL, the EXEC1 main program, and their associated subroutines (INFORM, etc.) expands the total target program to about 24 K words. The LOOP execution time is about 23 msec with assembly language versions of several general-purpose subroutines and function routines used.

HOST DIGITAL PROGRAM

The organization and flow of the host digital program is shown in Figure 7. The program flow is controlled by the main program. The main program and its associated subroutines and function routines read user-supplied input data and operate on those data to obtain the information needed to set up the target program on the hybrid computer. For example, the main program calls subroutine MAPIN to accomplish the reading in and scaling of component performance map data. A DO loop (IP=1 to NTOTAL) is used to read in cycle data for NTOTAL selected operating points. The IP=1 point is assumed to be the non-augmented (dry) design point. If augmented operation is included in the NTOTAL points, the NDRY+1 point is assumed to be the augmentor design point. The DCOEF subroutine is used to compute digital simulation coefficients and model trim factors. DCOEF is called for the IP=1 and IP=NDRY+1 operating points only. At the IP=1 point, the digital coefficients are computed from the input values of pressures, temperatures, flow rates, etc. and the scale factors. The trim factors are then determined and applied to particular coefficients so as to compensate for interpolation errors, etc. This produces (essentially) zero derivatives at the IP=1 design point. At the IP=NDRY+1 point, additional trim factors are computed and applied to achieve a balanced condition in the augmentor at the maximum thrust condition.

For each specified operating point, the main program calls subroutines ENGINE, ANALOG, and PRINT. Subroutine ENGINE uses the trimmed coefficients and the scaled model equations to compute scaled values of state variable derivatives. At off-design points, no guarantee of an equilibrium condition exists. Therefore, calculations are performed in ENGINE to obtain ratios of individual component model outputs to the output values required to achieve the desired equilibrium condition. Non-unity values for these ratios indicate the need for modifications to the models to match the operating line data.

The ANALOG subroutine is used to (1) read in engine geometric data and analog component addresses, (2) compute analog integrator gains and potentiometer settings,

(3) compute bypass and augmentor duct model evaluation ratios, and (4) generate computer printouts of all pertinent analog set-up information.

PRINT is a multi-purpose output routine that performs one of three functions, depending on the value of a calling argument IPRINT. IPRINT is initialized to zero in the main program and is incremented in PRINT just after each call. For the first call in the DO loop, PRINT merely lists the user-supplied operating point data. For the second call in the DO loop, PRINT lists all of the scaled and unscaled variables computed in ENGINE. After the final call, PRINT lists digital coefficients, trim factors, and model evaluation ratios and punches the digital and analog set-up data on cards. Approximately 24 K words are required by the host program.

RESULTS AND DISCUSSION

To demonstrate the computer-aided simulation development process, a test case was run. The test case involved simulating a 111.1 KN (25,000 lbf) thrust class engine operating at sea-level static conditions from idle to maximum thrust. As previously described, the design characteristics of the test-case engine and the specifications of the analog components were input to the host program. Execution of the host program resulted in the generation of computer printouts and the punching of a deck of cards containing the hybrid computer set-up data.

Upon setting up and executing the target simulation, it was discovered that stable, closed-loop operation of the simulation was not possible with time scale factors much below 50:1. After some study, it was concluded that this was due to relatively high loop gains in the "hot" sections of the engine model and the effective time delays associated with those loops. While it was felt that the time scale factor could probably be reduced by modifying the structure of the LOOP subroutine (update the high gain loops more often than the others) it was decided not to attempt this and, rather, to concentrate on demonstrating the basic simulation methodology. The host and target programs could, then, be the basis for later work aimed at reducing the digital frame time (23 msec) and speeding up the simulation.

With the simulation inputs (potentiometers) fixed at their specified values, the turbofan engine simulation was allowed to run until an equilibrium (steady-state) condition was reached. The INFORM subroutine was then used to obtain a tabular listing of unscaled steady-state data. Figure 8 shows a steady-state printout obtained at the idle (lowest thrust) condition. Note that the unscaled data are displayed as XXXXX EE which represents 0.XXXXX times 10 to the EE power.

Also note that steady-state error ratios (observed value/desired value) are listed for each variable. At the design point, the steady-state errors were very small (0.5 to 1.0 percent). As one moved further away from the design point, slightly larger errors were observed. In general, the simulation produced steady-state errors less than 2.5 percent over the entire sea-level, static operating line. This level of accuracy should be acceptable for most applications.

While no reference input data were available for evaluating the transient performance of the simulation, it was felt that it was important to demonstrate the transient operation of the simulation. To avoid the need for simulating an engine controller, it was decided to test the simulation dynamics in an "open-loop" fashion. To do this, time histories of the simulation inputs were constructed from representative engine data and implemented using analog function generators driven by a ramp generator denoting time. Time histories were generated for a number of typical engine transients. Figure 9 shows simulation responses to a cyclic movement of the throttle. In all cases, the simulation responses were stable and exhibited reasonable response times, overshoots, etc. No analog component overloads or scaled-fraction overflows were observed indicating that the selected scaled factors and organization of the target program computations were satisfactory.

CONCLUDING REMARKS

The value of hybrid computation as a simulation tool has been aptly demonstrated in a variety of applications including gas turbine engine controls development. Despite the tremendous technological advances in digital computation (microprocessors, array processors, etc.) the hybrid computer continues to play a significant role in simulation because of the speed of the analog computer and the "hands-on" interaction available to the user. Still, it is recognized that problems exist when developing hybrid computer simulations and that these problems, if not solved, can significantly reduce the effectiveness of the hybrid approach. In particular, programming aids are needed that can support the development, implementation, and documentation of the simulation and that can ensure acceptable levels of steady-state and dynamic accuracy.

This paper has focused on the gas turbine engine simulation problem and has presented a systematic, computer-aided, self-documenting methodology for developing a hybrid computer simulation of an augmented turbofan engine. The proposed simulation development process has been exercised, demonstrated, and documented for a typical turbofan engine design. The results indicate that the process does satisfy most of the objectives. That is,

1. The host/target program concept does provide a convenient means of developing, analyzing, and evaluating the engine model by using a digital computer (host) without tieing up the hybrid computer (target) during the development process.
2. The inclusion of a scaled turbofan engine model in both the host and target computer programs minimizes the need for computer programming (formulating, scaling, coding, debugging) by the user.
3. The self-trimming features of the host program lead to a hybrid simulation that matches user-supplied steady-state design point data within 1 percent and matches off-design point data within 2 to 3 percent.
4. The task of modifying engine subsystem models to match off-design point data is simplified by having the host program calculate and print out tables of model evaluation ratios.
5. The computer subroutines, function routines, and blocks of computer code have been generalized and should prove valuable in constructing host/target programs for other engine configurations.

While the basic simulation development methodology has been demonstrated, the goal of applying the concepts to real-time engine simulation has not been achieved. However, it is felt that the host and target programs that have been developed provide an excellent vehicle and opportunity for exploring various approaches to reducing the digital frame time and the time scale factor (currently 50:1). Preliminary results from studies dealing with more frequent updates of high gain loops in LOOP indicate that 20:1 time scale is achievable without affecting the basic engine model. Other possibilities that should be explored include: replacing bivariate table lockups with analytic functions, revising the split between analog and digital computation, and introducing other forms of parallel processing. Again, the existing host and target computer programs should prove to be valuable tools in performing these studies and documenting the results. Further details and documentation of the host and target computer programs are available from the authors upon request.

SYMBOL LIST

A	cross-sectional area, sq cm (sq in)
ALT	altitude, m (ft)
DTQWj	scaled non-specific temperature derivative at station j
DWj	scaled stored mass derivative at station j

F	thrust, N (lbf)
IP	integer index
IPRINT	integer argument for subroutine PRINT
NDRY	number of non-augmented operating points input to host program
NTOTAL	total number of operating points input to host program
Δh	enthalpy change, J/kg (Btu/lbm)
M	Mach number
N	rotational speed, rpm
p	total pressure, N/sq cm (psia)
Q	torque, N-cm (in-lbf)
SF_x	scale factor on variable x, appropriate units
T	total temperature, K (^{o}R)
τ	update time, sec
t	time, sec
V	volume, cu cm (cu in)
W	stored mass, kg (lbm)
$\dot{w}$	mass flow rate, kg/sec (lbm/sec)

Subscripts:

AB	augmentor
am	ambient
BLHT	high-pressure-turbine cooling bleed
BLLT	low-pressure-turbine cooling bleed
BLOV	overboard bleed
C	compressor
E	nozzle exit plane
F	fuel
FAN	fan
H	high-pressure-spool
HT	high-pressure-turbine
ID	fan hub
j	station (see Figure 2), j=0,2,2.1,2.2,3,4,4.1,5,6,7,8,13,16
j'	entrance to volume at station j (see Figure 3), j=3,7,13
L	low-pressure-spool
LT	low-pressure-turbine
N	nozzle
OD	fan tip

Note - subscripts may be combined (example, $\dot{w}_{F,4}$)

REFERENCES

1. Szuch, John R.; and Bruton, William M.: Real-Time Simulation of the TF30-P-3 Turbofan Engine Using a Hybrid Computer. NASA TM X-3106, 1974.

2. Szuch, John R.; and Seldner, Kurt: Real-Time Simulation of the F100-PW-100 Turbofan Engine Using the Hybrid Computer. NASA TM X-3261, 1975.

3. Szuch, John R.; Seldner, Kurt; and Cwynar, David S.: Development and Verification of a Real-Time, Hybrid Computer Simulation of the F100-PW-100(3) Turbofan Engine. NASA TP-1034, 1977.

4. Cwynar, David S.; and Batterton, Peter G.: Digital Implementation of the TF30-P-3 Turbofan Engine Control. NASA TM X-3105, 1974.

5. Hrach, Frank J.; Arpasi, Dale J.; and Bruton, William M.: Design and Evaluation of a Sensor Fail-Operational Control System for a Digitally Controlled Turbofan Engine. NASA TM X-3260, 1975.

6. Arpasi, Dale J.; and Bruton, William M.: Method of Discrete Modeling and its Application to Estimation of TF30 Engine Variables. NASA TM X-3443, 1976.

7. Szuch, John R.; et al.: F100 Multivariable Control Synthesis Program - Evaluation of Multivariable Control Using a Real-Time Engine Simulation. NASA TP-1056, 1977.

8. Szuch, John R.: HYDES, A Generalized Hybrid Computer Program for Studying Turbojet or Turbofan Engine Dynamics. NASA TM X-3012, 1974.

9. Bruton, William M.; and Cwynar, David S.: Lewis Hybrid Computing System - Users Manual. NASA TM-79111, 1979.

10. Cwynar, David S.: INFORM - An Interactive Data Collection and Display Program with Debugging Capability. NASA TP-1424, 1980.

TABLE I. - TARGET PROGRAM STATISTICS

Analog Program	Interface	Digital Program
16 Integrators	25 ADC's	Basic program - 7487 words
19 Summers	21 DAC's	Total program - 23,805 words
6 Multipliers	2 Control lines	LOOP execution time - 23.0 msec
2 X^2	3 Sense lines	
8 Dividers	1 General purpose interrupt	
53 Potentiometers	1 Real-time clock	
10 Function relays		
4 "AND" gates		
1 BCD counter		
1 Logic pushbutton		

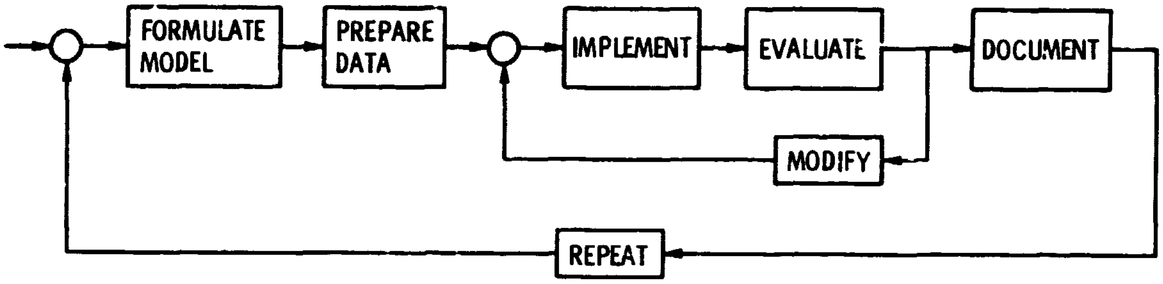


Figure 1. - Simulation development process.

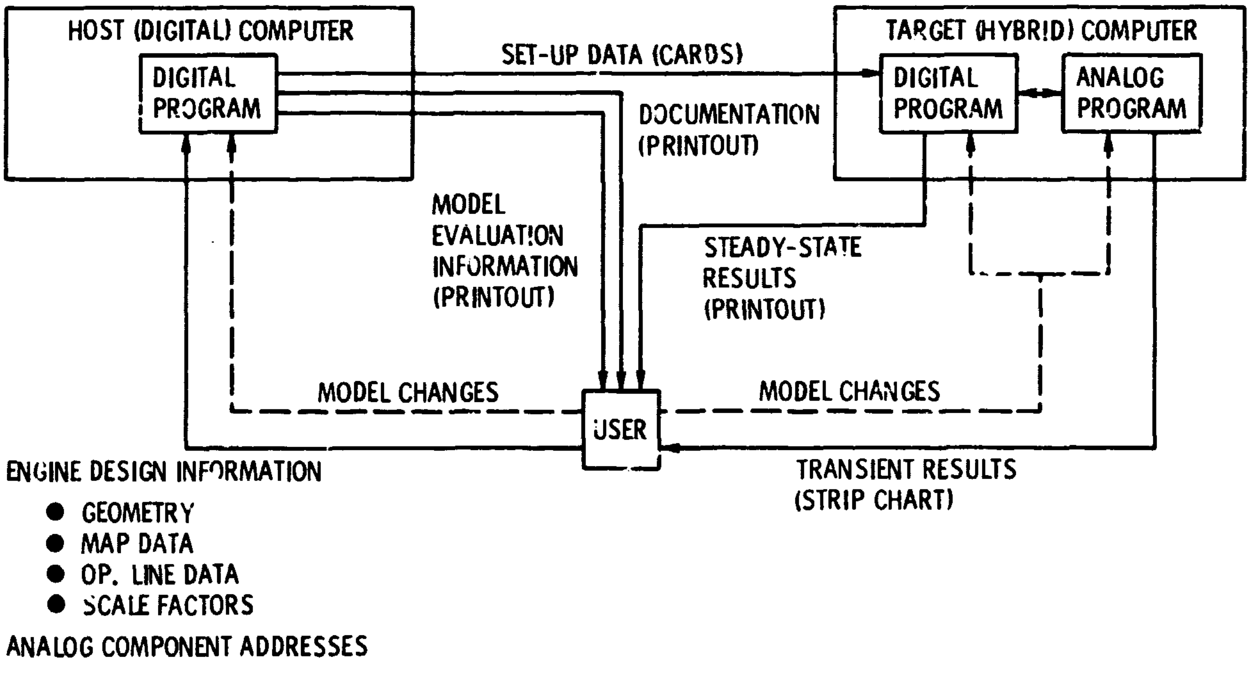


Figure 2. - Proposed simulation development methodology.

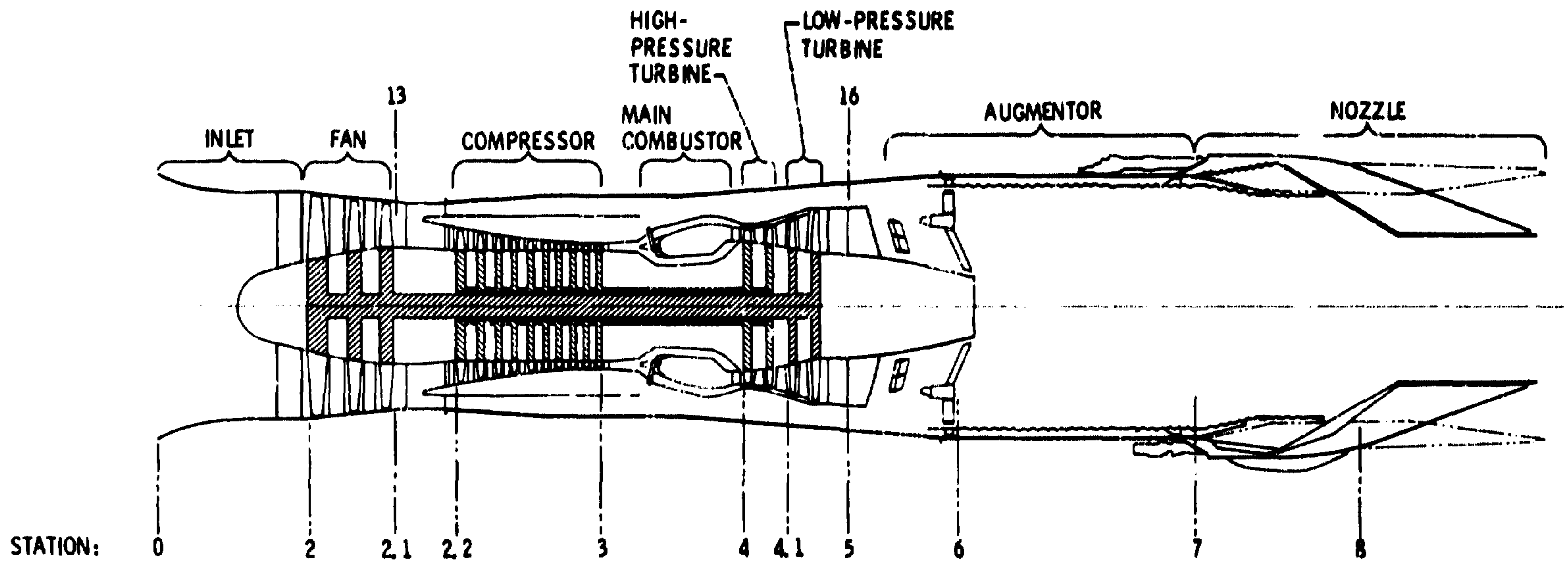


Figure 3. - Schematic representation of augmented turbofan engine.

CD-11819-07

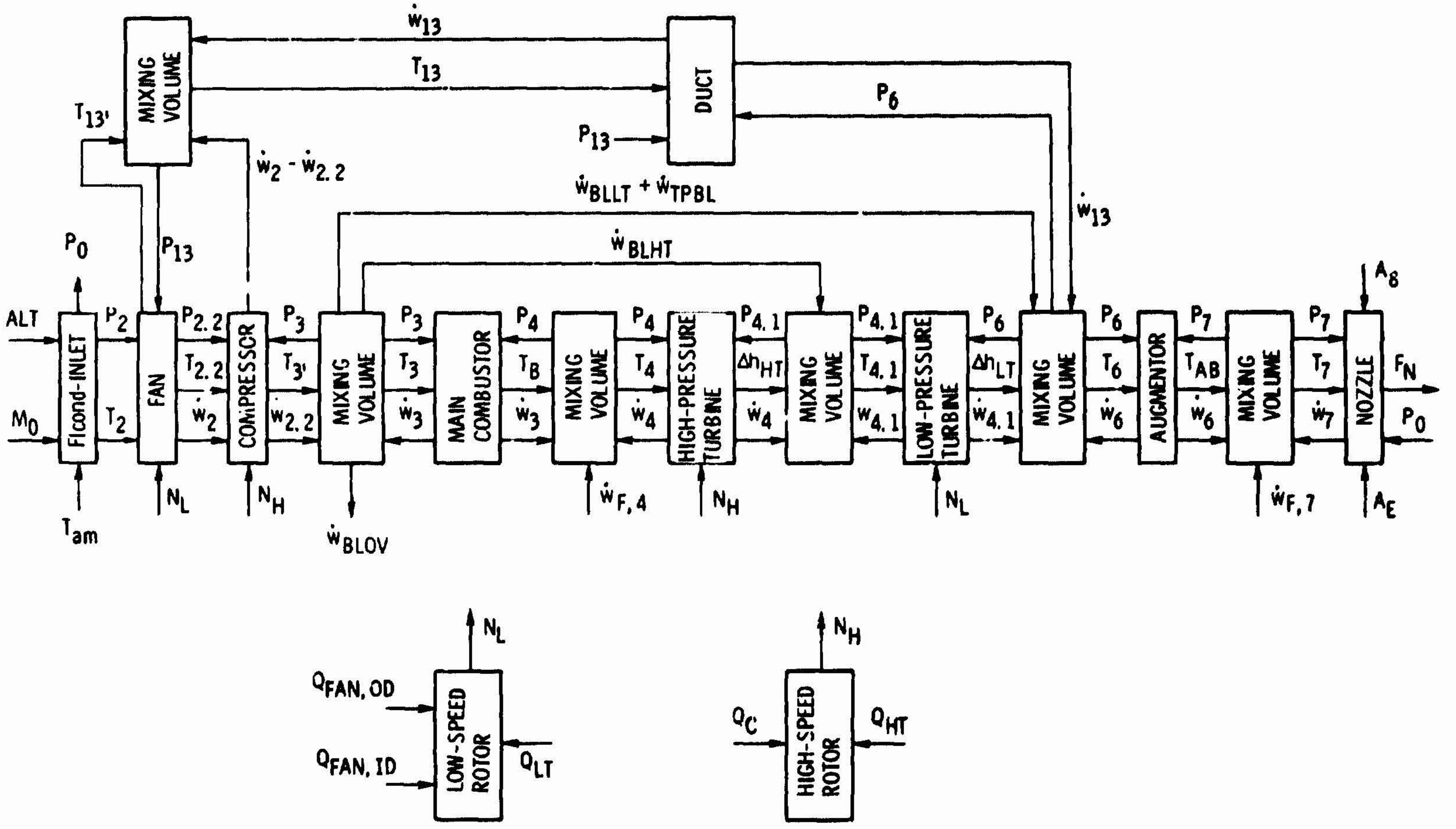


Figure 4. - Computational flow diagram of augmented turbofan engine simulation.

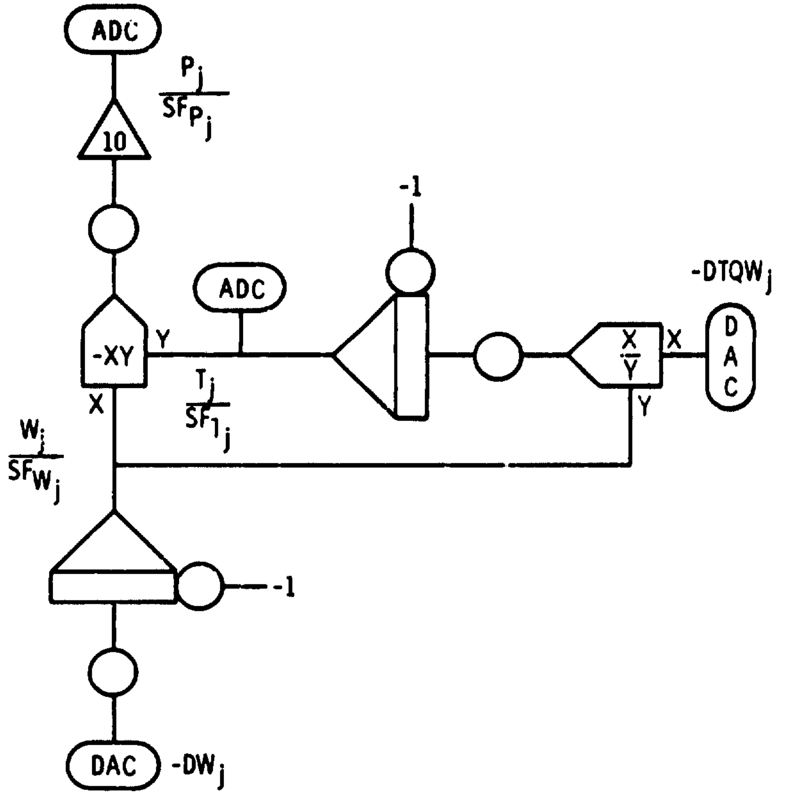


Figure 5. - Analog computation of intercomponent volume variables.

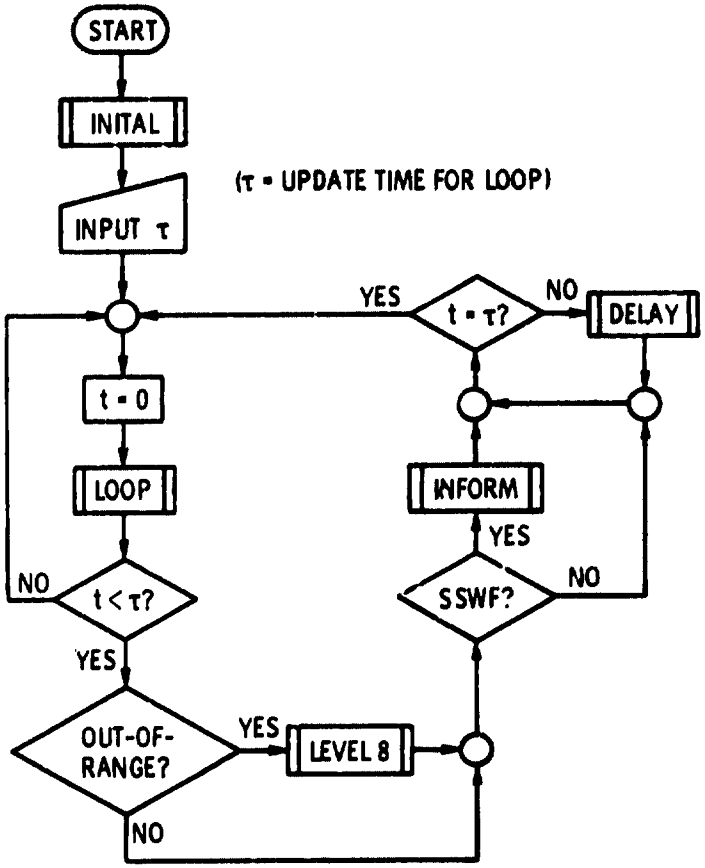


Figure 6. - Flowchart of target digital program, EXEC1.

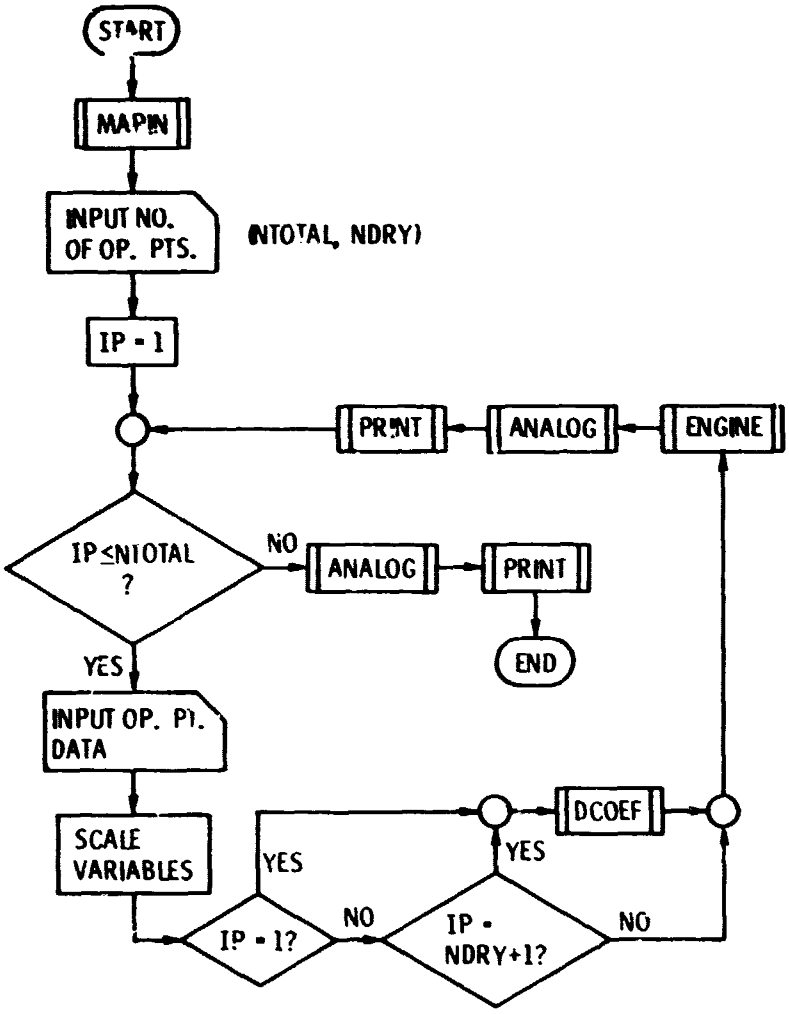
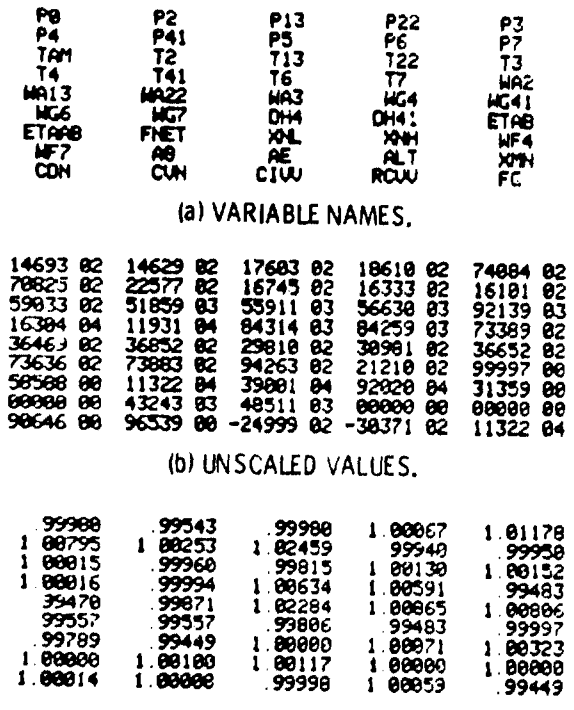


Figure 7. - Flowchart of host digital program.



```
  P0        P2        P13       P22       P3
  P4        P41       P5        P6        P7
  TAM       T2        T13       T22       T3
  T4        T41       T6        T7        WA2
 WA13      WA22       WA3       WG4      WG41
  WG6       WG7       DH4      DH41:     ETAB
ETAAB      FNET       XNL       XNH       WF4
  WF7       A8        AE        ALT       XMN
  CON       CVN      CIVV      RCVV       FC
```

(a) VARIABLE NAMES.

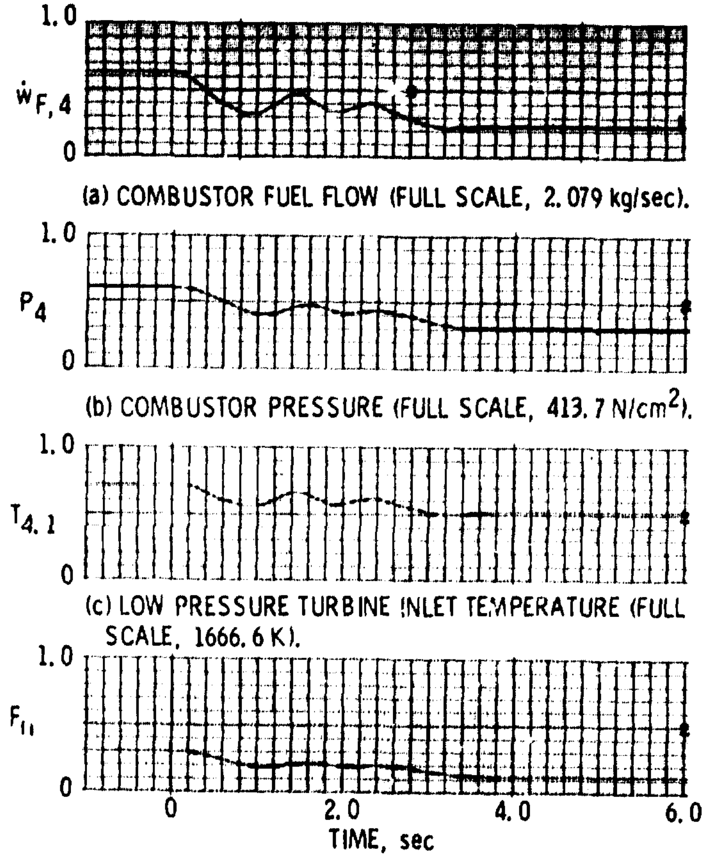
```
14693 02  14629 02  17603 02  18610 02  74084 02
70825 02  22577 02  16745 02  16333 02  16101 02
59033 02  51859 03  55911 03  56630 03  92139 03
16304 04  11931 04  84314 03  84259 03  73389 02
36469 02  36852 02  29810 02  30901 02  36652 02
73636 02  73883 02  94263 02  21210 02  99997 00
58508 00  11322 04  39001 04  92020 04  31359 00
00000 00  43243 03  48511 03  00000 00  00000 00
90646 00  96539 00 -24999 02 -30371 02  11322 04
```

(b) UNSCALED VALUES.

```
  .99980    .99543    .99980  1.00067   1.01178
 1.00795   1.00253   1.02459   .99940    .99950
 1.00015    .99960    .99815  1.00130   1.00152
 1.00016    .99994   1.00634  1.00591    .99483
  .99470    .99871   1.02284  1.00065   1.00806
  .99557    .99557    .93806   .99483    .99997
  .99789    .99449   1.00000  1.00971   1.00323
 1.00000   1.00100   1.00117  1.00000   1.00000
 1.00014   1.00000    .99998  1.00053    .99449
```

(c) ERROR RATIOS, OBSERVED/DESIRED.

Figure 8. - Target computer printout of steady-state results at idle thrust.



(a) COMBUSTOR FUEL FLOW (FULL SCALE, 2.079 kg/sec).

(b) COMBUSTOR PRESSURE (FULL SCALE, 413.7 N/cm^2).

(c) LOW PRESSURE TURBINE INLET TEMPERATURE (FULL SCALE, 1666.6 K).

(c) NET THRUST (FULL SCALE, 2.224x10^5 N).

Figure 9. - Simulated response to cyclic throttle movement.