NOTICE

NASA Technical Memorandum 82825

Advancements in Real-Time Engine Simulation Technology

(NASA-TM-82825) ADVANCEMENTS IN REAL-TIME ENGINE SIMULATION TECHNOLOGY (NASA) 12 p HC A02/MF A01 CSCL 21E
N82-22915
Unclas
G3/62 09699

John R. Szuch
Lewis Research Center
Cleveland, Ohio

Prepared for the
Eighteenth Joint Propulsion Conference
cosponsored by the AIAA, SAE, and ASME
Cleveland, Ohio, June 21-23, 1982

NASA

E-1177

ADVANCEMENTS IN REAL-TIME
ENGINE SIMULATION TECHNOLOGY

John R.Szuch
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio

Abstract

The development of digital electronic engine controls can be facilitated by the use of computer simulations of the engine. This paper reviews many of the approaches currently being used to develop real-time engine simulations. Both digital and hybrid (analog + digital) techniques are discussed and specific examples of each are cited. These approaches are assessed from the standpoint of their usefulness for digital engine control development. A number of NASA-sponsored simulation research activities, aimed at exploring new real-time simulation techniques, are described. These include the development of a microcomputer-based, parallel-processor system for real-time engine simulation.

Introduction

Aircraft propulsion systems are becoming increasingly complex due to requirements for higher thrust-to-weight ratios, reduced fuel consumption, extended engine life, and greater operational flexibility. This increased complexity is reflected in an increased number of engine variables that must be manipulated and controlled. As shown in Fig. 1, these may include variable fan and compressor geometry, variable turbine nozzles, variable bypass valves and injectors, and variable convergent-divergent nozzles. The control of these variables must be coordinated with the control of fuel flows to the engine to provide the desired performance and engine protection. Also, in applications such as powered-lift, the engine control functions may have to be integrated with the flight controls.

It is generally agreed that hydromechanical controls will not be able to perform all of the control functions that will be required by future aircraft propulsion systems.[1] In fact, many of today's modern jet engines already use analog or digital electronic supervisory controls to complement hydromechanical hardware . In the future, it is expected that full-authority digital electronic controls will be needed to satisfy the demanding requirements for multivariable control, integrated propulsion-flight controls, and engine diagnostics.[2]

The development of digital electronic engine controls can be facilitated by the use of computer simulations of the engine.[3,4] Simulations can provide a basis for the design of the control logic and a cost-effective way of evaluating the dynamic behavior of the engine and control combination prior to any full-scale engine testing. In addition, the simulations can also serve as aids in solving problems that arise after the control development is completed.

Typically, a special-purpose engine simulation (i.e. mathematical model and computer implementation) will be developed for each phase of the engine/control development process. This is because of the widely varying simulation requirements associated with each application. As shown in Fig. 2, these applications demand varying degrees of simulation complexity (model detail) and computing speed. For performance studies, where the focus is on internal engine aerothermodynamics, detailed models are required but there is no critical need for fast execution. On the other hand, an application like a piloted, moving-base flight simulator requires a faster-than-real-time engine simulation. This is because a single digital computer is usually used to perform serial calculations of the airframe dynamics, engine thrust, engine and flight control functions, simulator motion commands, and cockpit displays. In this case, the engine's internal aerothermodynamics are not of primary interest and the engine model is usually simplified to achieve the required computing speed.

This paper reviews many of the approaches currently being used to develop real-time engine simulations. Both digital and hybrid (analog + digital) techniques are discussed and specific examples of each are cited. These approaches are assessed from the standpoint of their usefulness for digital engine control development. A number of NASA-sponsored simulation research activities, aimed at exploring new real-time simulation techniques, are described. These include the development of a microcomputer-based, parallel-processor system[5] for real-time engine simulation.

Engine Model

As indicated earlier, there are a multitude of application-specific engine simulations that are being used to support the development of digital engine controls. All of these simulations have, as their basis, an aerothermodynamic description of the engine. In many cases, the starting point for developing a time-critical engine simulation will be a detailed model such as the one illustrated in Fig. 3. Here, each engine component is represented by a set of maps that describes the steady-state performance characteristics of that component. The individual component models are interconnnected by control (mixing) volumes which account for the storage of mass and energy within the engine. Rotor speeds are calculated by integrating excess shaft torques. The effects of variable geometry are introduced into the model as corrections (biases) to the fixed-geometry performance maps. Additional levels of detail may be included in the model to improve the steady-state and/or transient accuracy of the model. These may include time-varying fluid properties, heat transfer between gas and metal, multi-lump component models, variations in turbine cooling air, and turbomachinery clearance variations.[6]

Unfortunately, the large number of calculations associated with a detailed aerothermodynamic engine model preclude real-time digital solutions on all but the fastest, most costly computer systems. This is illustrated in Fig. 4 for a Fortran implementation of an augmented turbofan engine model. If such a computer isn't available or can't be dedicated to the real-time simulation task then alternative approaches must be considered. The following sections describe software and hardware techniques that have been used/proposed for real-time engine simulation.

Software Approaches to Real-Time Simulation

The major drawback when implementing real-time engine simulations on digital computers is that the time it takes to calculate all of the simulation values is longer than the time step necessary to provide good resolution at the highest frequency of interest. This problem is aggravated when one attempts to use simple, explicit integration techniques (e.g. Euler). Then, an even smaller time step may be required to produce a stable solution. This is especially true for "stiff" systems like jet engines where there are widely varying time constants in the simulation model. The real-time simulation problem then becomes one of (1) reducing the calculation time, (2) taking measures to permit the step size to be increased, or (3) both.

Perhaps the most obvious approach to achieving real-time capability is to simplify the engine model. Elements of the model can be removed or modified to reduce the computing time while, hopefully, retaining the essential steady-state and dynamic characteristics of the engine.

Mihaloew [7] has proposed a real-time digital simulation technique that can produce a faster-than-real-time engine simulation when run on a dedicated, general-purpose mainframe (Xerox Sigma 8). In this case, the computer was part of a moving-base flight simulator facility and the engine model had to be integrated with a simulation of an airplane. This meant that the time step, for real-time operation of the flight simulator, was longer than the time allotted for the engine calculations. The proposed technique involves (1) multi-rate scheduling (updating) of various parts of the model, (2) the elimination of control volume dynamics, (3) the use of high-gain integrators to converge the resulting algebraic loops, and (4) curve-fitting of the component performance maps with segmented polynomial and geometric functions. The resulting simulation was stable with time steps as large as 50 msec. and exhibited satisfactory low frequency (rotor) dynamics. While this technique has been demonstrated successfully in a powered-lift application at low altitude-low speed conditions, more work is needed to determine if the simplifications can be generalized and applied to other engine systems over a wider range of operating conditions.

It is also possible to simplify the detailed engine model by linearizing the nonlinear model about selected operating points. The coefficients of the linear models can then be functionally related to the engine states and inputs. The resulting piecewise-linear models can be implemented as transfer functions using recursion formulas. [8] One of the significant advantages of this technique is the unconditional stability of the model regardless of the time step size. The major drawback of this technique appears to be the time and effort required to create a good model. The process of (1) generating full-state linear models from a nonlinear model at several operating points, (2) reducing the order of the linear models while retaining the important states in the models, (3) determining the functional relationships between the coefficients of the transfer functions, and (4) comparing the linear and nonlinear solutions may have to be repeated many times before satisfactory results are obtained.

If simplifications to the nonlinear model can't be made easily then one may have to resort to numerical methods to allow the time step to be increased. McKinney [9,10] and McLaughlin [11] have used a technique that involves an implicit formulation of the engine equations and an iterative solution of those equations. By using backward-difference integration, they are able to guarantee a stable solution even when using time steps larger than the smallest time constant in the engine model. However, steps must be taken to reduce the overhead associated with the iteration procedure so as to keep the computing time from exceeding the time step size dictated by the application. To do this, McLaughlin uses the Broyden technique [12] to effectively eliminate the need to calculate new Jacobian matrices during the course of a transient. Stable, real-time solutions have been obtained on a Digital Equipment Corporation PDP-11/55 with a time step of 27 msec. While this time step is too large for some control applications, it does demonstrate that the implicit method can produce real-time simulations of "stiff" systems on low-cost minicomputers. Some difficulties have been encountered with numerical instabilities which are attributed to his use of scaled-integer arithmetic. McLaughlin suggests the use of floating-point calculations in future applications of this technique.

Perhaps the most desirable approach to achieving real-time engine simulations is to somehow speed up the calculations. Faster hardware would permit the solution of detailed nonlinear models without the need for simplifications. This possibility exists if one considers the use of parallel processing (e.g., parallel computers, peripheral processors, hybrid computers, etc.). This approach is discussed in the next section.

Hardware Approaches to Real-Time Simulation

The concept of fashioning a network of computers that can concurrently (in parallel) solve different portions of a simulation model offers the promise of being able to solve large, complex, nonlinear engine models in real-time. The hybrid computer, which contains both analog and digital processors, is an example of a parallel processing system. By appropriately splitting the engine calculations between the analog and digital computers, one can increase the simulation speed relative to an all digital simulation and can improve the accuracy and repeatability of results relative to an all analog simulation. The hybrid computer has been used successfully for real-time engine simulation. [13,14] In those instances, however, it was necessary to limit the number of digital calculations (hence, the digital frame time) to preserve the dynamic accuracy and stability of the simulation. The result was a predominantly analog simulation with the digital used only for multivariable map table lookups. In general, the development of a real-time hybrid computer simulation of a gas turbine engine has proved to be a time-consuming task, requiring innovative modeling and programming techniques to achieve the desired result. Also, the general purpose hybrid computer is a relatively expensive, nonportable system that may not be suitable for many applications.

Paragon Pacific Inc. has pioneered the use of special-purpose, hard-wired hybrid computers (SPHYC's) to achieve a low-cost, portable real-time simulator. The SPHYC is made up of miniature operational amplifiers and digital circuit chips that have been configured into standard math modules. The modules are wire-wrapped together to solve a particular set of nonlinear differential equations. Parameters may be adjusted manually by means of potentiometers.

The Boeing Military Airplane Company has been investigating the use of the SPHYC for real-time engine simulation.[15] Boeing has developed an SPHYC implementation of an F100 engine model. However, the performance of that simulation has yet to be reported. The SPHYC approach has been used, quite successfully, to simulate helicopter and wind turbine rotor dynamics in real-time. The most obvious shortcoming of this approach would appear to be its lack of reprogrammability by the user. This shortcoming has led Paragon Pacific, Inc. and others to consider all-digital approaches to parallel processing in order to achieve a more general-purpose real-time simulation capability.

The requirements for high-speed signal processing in applications such as radar, communications, and image processing have stimulated the development of special-purpose peripheral digital processors that might be useful for real-time engine simulation. These devices include the AP-120B, manufactured by Floating Point Systems, Inc., the MAP-300, manufactured by Computer Signal Processors, Inc. and the AD-10, manufactured by Applied Dynamics, Inc. These peripheral processors employ the latest electronic components and design techniques to provide high-speed arithmetic computations at a moderate cost. They require, however, a general-purpose digital computer such as a Digital Equipment Corporation PDP-11/70 to act as a buffer for communication lines, terminals, I/O, etc. A system such as this will cost between $300,000 and $500,000. Sugiyama[16] has used the AD-10 to implement a generalized engine model.[17] Real-time response was demonstrated with a frame time of 1.1 msec when simulating a two-spool turbofan engine. The Pratt & Whitney Aircraft Group also uses the AD-10 for real-time engine simulation in a closed-loop bench test facility. The major disadvantages of this approach appear to be a lack of floating-point capability and the difficulty in programming the AD-10 to take advantage of its pipelined architecture. Also, the AD-10-host computer combination may not be portable enough to satisfy many simulation applications. Applied Dynamics Inc. has announced a high-order programming language MPS-10 that is intended to simplify the programming task.

The prospect of using a network of general-purpose digital computers for real-time simulation has been entertained for some time. Korn[18] showed that a system of three PDP-11/45's could outperform analog computers in most applications. More recently, the availability of very fast and inexpensive microprocessors has made it possible to consider the development of high-speed, low-cost, simulation-oriented parallel-processor systems that are compact enough to transport for on-site simulation applications. O'Grady[19] has proposed a two-dimensional array of up to 256 processors that are interconnected in such a way as to permit fast solutions of large dynamic systems. Under a NASA grant NAG 3-112, O'Grady is building prototype hardware and developing software that will permit an investigation of this concept and its application to engine simulation. However, to minimize cost and technical risk, O'Grady has selected commercially available microcomputers as the "building blocks" in his simulator system. As a result, realtime capability has been sacrificed in order to demonstrate the parallel processing concept. The following section describes other NASA research efforts aimed at achieving a low-cost, portable real-time engine simulation capability.

NASA-Sponsored Simulation Research

The NASA Lewis Research Center has, for many years, used the hybrid computer for real-time engine simulations. These simulations have served as tools in the development of advanced digital electronic control systems. But, recognizing the shortcomings of the hybrid approach and other current approaches to real-time engine simulation, Lewis recently embarked on a multi-faceted program to apply parallel processing and innovative software techniques to this problem.

One objective of the Lewis program is to design, build, and test a prototype digital simulator[5] that will be capable of simulating a detailed aerothermodynamic model of a turbofan engine in real-time. The basic structure of the simulator being proposed is shown in Fig. 5. The heart of the system is a transfer controller that synchronizes N 16-bit processing elements (P.E.'s) on a high-speed data transfer bus. All but two of the P.E.'s perform simulation computations. One of the remaining P.E.'s is dedicated to input/output functions. The last P.E. is a special-purpose processor that links low-speed, operator commands to the high-speed simulator. This P.E. is termed the "real-time extension" of the front-end processor. The microprocessor-based front-end processor provides the operator's interface to the simulator and handles peripheral communications, program loading, etc.

The simulator cycle is separated into two basic periods - a compute period and a transfer period. The transfer period is initiated when all P.E.'s have completed their computations. During the transfer period, data are exchanged between the simulation P.E.'s as dictated by the transfer controller. The real-time operation of the simulator depends on the simulation calculations being distributed among the P.E.'s by the user in such a way that the sum of the largest compute time and the transfer time doesn't exceed the specified step size (frame time).

In the Lewis effort, both commercially-available microcomputers and a custom SSI/MSI design are being considered for use as the P.E. The SSI/MSI processor is microprogrammable to allow experimentation with different simulation-oriented instruction sets. Its projected cycle time is 133 nanoseconds with many instructions taking only one cycle. The NASA-designed hardware is optimized for interprocessor transfers of data. It is hoped that the speed of the P.E. and the transfer mechanism will permit the implementation of detailed engine models and, if necessary, the incorporation of sophisticated numerical algorithms to achieve accurate, real-time solutions.

Simulator hardware and associated software are currently undergoing exploratory development at Lewis. Prototype hardware, such as the SSI/MSI P.E. shown in Fig. 6, is being fabricated and tested. Integration of all components into the simulator system is expected to be completed by mid 1983. At that time, it is planned to begin running applications programs including a nonlinear turbofan engine simulation.

The development of real-time engine simulations for the simulator will depend on the selection and implementation of a suitable numerical integration algorithm. Miranker and Liniger have proposed the use of parallel, predictor-corrector algorithms[20] for the solution of ordinary differential equations on parallel-processor systems. The most obvious advantage of these algorithms is that they don't require the user to partition the simulation model for parallel solution. Each processor computes all of the state-variable derivatives. However, each processor computes its derivatives for a different point in time. Since more than one time step is calculated during an update interval, the simulation is effectively speeded up by a factor of N/2 where N is the number of processors. Krosel and Milner have applied the Miranker and Liniger fourth-order algorithms to the engine simulation problem[21] and have concluded that, in the engine application, a practical limit of four processors exists due to the complexity of the algorithm. Further increasing of the number of processors requires smaller time steps to maintain accuracy and leads to numerical instabilities. For the engine model that was used, the Miranker and Liniger algorithm required the derivative calculations to be performed in less than 2 msec. for real-time operation. In the case of the Lewis simulator, this may be feasible because of the speed of the SSI/MSI processing elements. However, in other parallel-processor systems, it may be necessary to partition the engine model and, possibly, use implicit integration methods to handle fast (>10 Hz) dynamics.

As previously noted, McLaughlin has applied backward-difference integration and the Broyden update algorithm in a quasi-iterative scheme to produce a real-time engine simulation on a minicomputer. The structure of the integration and update algorithms suggests the possibility of partitioning both the algorithms and the engine model for solution on a parallel-processor system. This could lead to a high-fidelity, real-time simulation that is stable with any desired (achievable) step size. NASA Lewis is investigating this approach and developing generalized computer programs that will perform the partitioning tasks for any user-specified target computer/simulator system. It is planned to have the programs operational on the NASA Lewis IBM 370-3033 in early 1983.

Conclusions

It has been aptly demonstrated that 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. Because of the high cost of ground and flight tests, it is expected that more and more use will be made of simulations during all phases of engine/flight control development, including the validation and evaluation of control hardware and software.

While both hybrid and digital techniques are currently being used for real-time engine 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-processor systems that will allow the implementation of high-fidelity, real-time engine simulations in a cost-effective manner. Future advancements in VLSI and VHSIC technologies may permit these parallel-processor systems to be miniaturized and integrated with digital engine controls for flight applications such as adaptive control and fault detection/accomodation.

It is hoped that the availability of low-cost, portable, real-time engine simulations will lead to the acceptance and widespread use of unified control development methodologies that employ a common simulation data base and perhaps a single simulation implementation for the various phases (design, development, validation, evaluation, test support, etc.) of the control development. This capability would benefit contractors, government agencies, and universities engaged in control development activities and would lead to increased control system reliability and reduced control development costs.

References

1. Szuch, J.R., "Control Technology," NASA CP 2092, Oct. 1979.

2. Vizzini, R.W., Lenox, T.G., and Miller, R.J., "Full Authority Digital Electronic Control Turbofan Engine Demonstration," SAE Paper 801199, Oct. 1980.

3. Szuch, J.R., "Application of Real-Time Engine Simulations to the Development of Propulsion System Controls," AIAA Paper 75-1176, Sept. 1975.

4. Szuch, J.R., Skira, C., and Soeder, J.F., "Evaluation of an F100 Multivariable Control Using a Real-Time Engine Simulation," AIAA Paper 77-835, July 1977.

5. Blech, R.A., and Arpasi, D.J., "An Approach to Real-Time Simulation Using Parallel Processing," NASA TM-81731, 1981.

6. Khalid, S.J., and Hearne, R.C., "Enhancing Dynamic Model Fidelity for Improved Prediction of Turbofan Engine Transient Performance," AIAA Paper 80-1083 June 1980.

7. Mihaloew, J.R., "A Nonlinear Propulsion System Simulation Technique for Piloted Simulators," NASA TM-82600, 1981.

8. Hurt, J.M., "New Difference Equation Technique for Solving Nonlinear Differential Equations," Proceedings of the Spring Joint Computer Conference, Washington, D.C., vol. 25, April 1964, pp. 169-179.

9. McKinney, J.S., "Simulation of Turbofan Engine. Part I. Description of Method and Balancing Technique," AFAPL-TR-67-125-pt.-1, Nov. 1967, (AD-825197).

10. McKinney, J.S., "Simulation of Turbofan Engine. Part II. User's Manual and Computer Program Listing," AFAPL-TR-67-125-pt.-2, Nov. 1967 (AD-825198).

11. McLaughlin, P., "A Technique for the Implementation of Nonlinear Models as Real-Time Digital Simulations - with application to aircraft turbine engine design," Proceedings of Summer Computer Simulation Conference, Seattle, Washington, Aug. 1980, pp. 111-115.

12. Broyden, C.G., "Quasi-Newton Methods and Their Application to Function Minimization," Mathematics of Computation, vol. 21, July 1967, pp. 368-381.

13. Szuch, J.R., and Bruton, W.M., "Real-Time Simulation of the TF30-P-3 Turbofan Engine Using a Hybrid Computer," NASA TM X-3106, 1974.

14. Szuch, J.R., Seldner, K., and Cwynar D.S., "Development and Verification of Real-Time Hybrid Computer Simulation of F100-PW-100(3) Turbofan Engine," NASA TP 1034, 1977.

15. Carlin, C.M., and Tjonneland, E.,"Propulsion System Controls Design and Simulation," AIAA Paper 82-0322, Jan. 1982.

16. Sugiyama, N., "Real-Time Digital Simulation of Jet Engines," Technical Report, University of Michigan, April, 1979.

17. Szuch, J.R.," HYDES-A Generalized Hybrid Computer Program for Studying Turbojet or Turbofan Engine Dynamics," NASA TM X-3014, 1974.

18. Korn, G.A., "Back to Parallel Computation-Proposal for a Completely New On-Line Simulation System Using Standard Minicomputers for Low-Cost Multiprocessing," SIMULATION, vol.19, no.2, Aug. 1972, pp. 37-45.

19. O'Grady, E.P., "A Communication Mechanism for Multiprocessor Simulation Systems," SIMULATION, vol. 34, no.2, Feb. 1980, pp. 39-49.

20. Miranker, W., and Liniger, W., "Parallel Methods for the Numerical Integration of Ordinary Differential Equations," Mathematics of Computation, vol.21, July 1967, pp. 303-320.

21. Krosel, S.M., and Milner, E.J., "Application of Integration Algorithms in a Parallel Processing Environment for the Simulation of Jet Engines," NASA TM-82746, 1981.

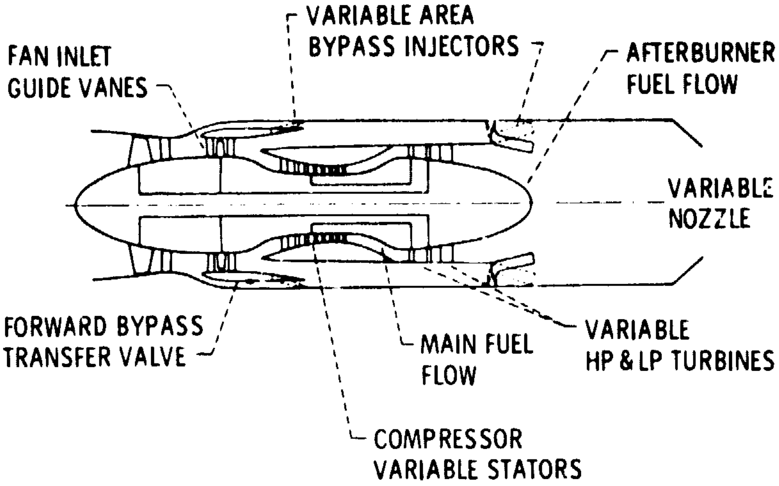

Figure 1. - Variable cycle engine.

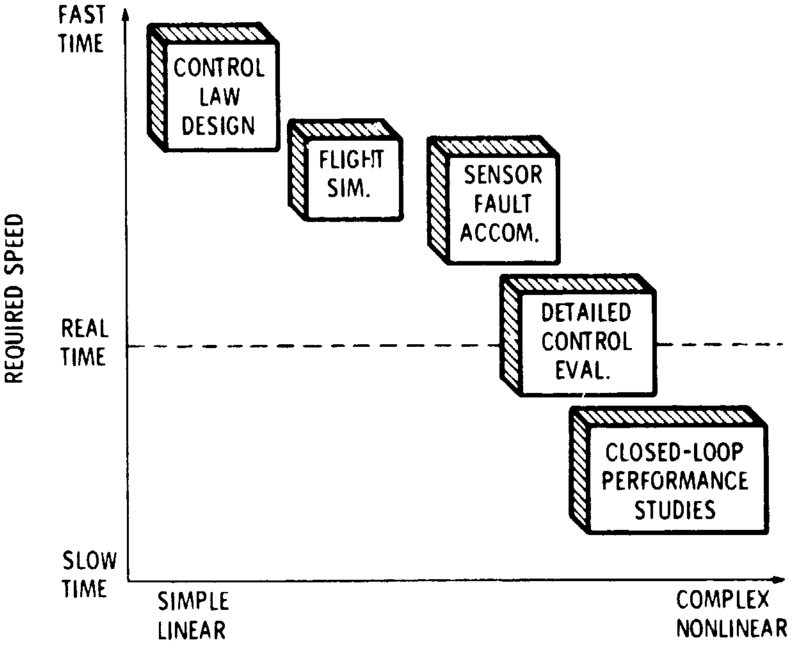

Figure 2. - Engine simulation requirements for control system development.

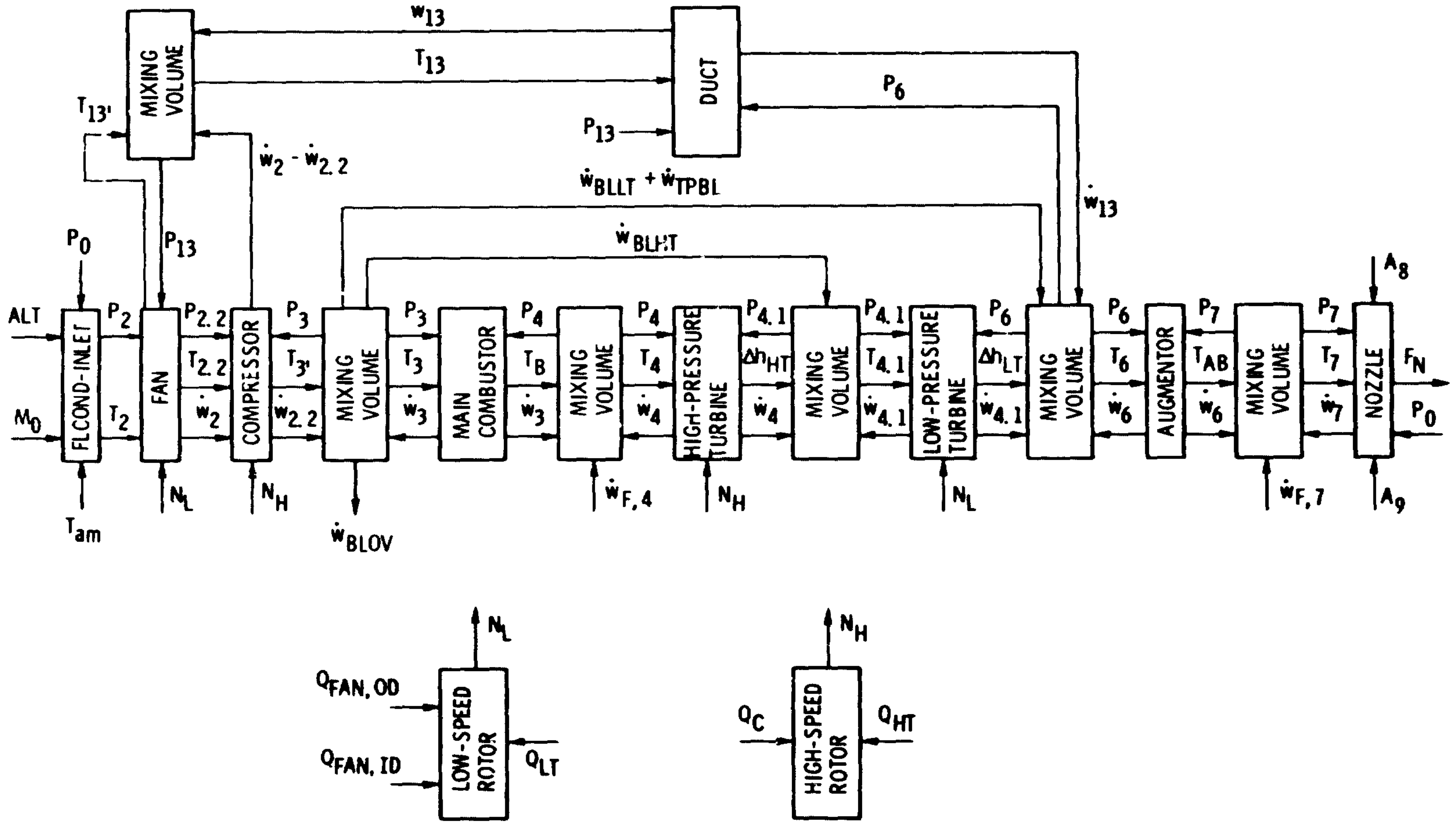

Figure 3. - Computational flow diagram of augmented turbofan engine simulation.

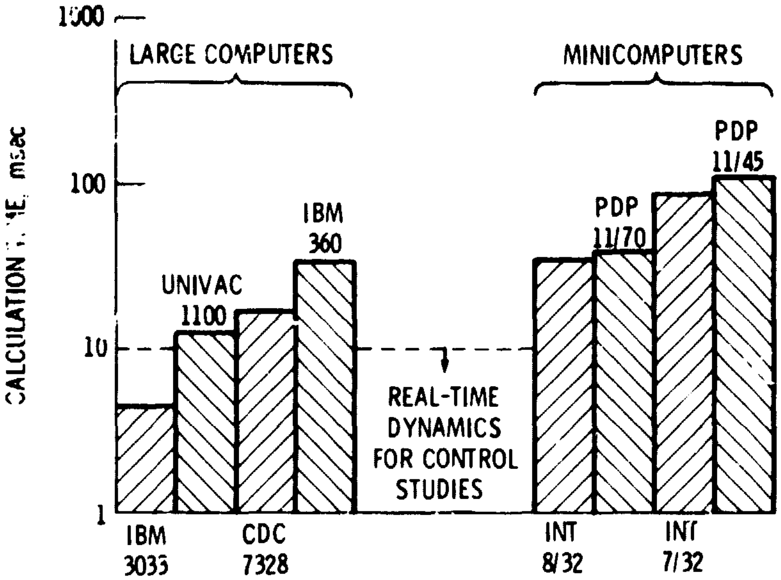

Figure 4. - Digital calculation times for FORTRAN implementation of turbofan engine model.

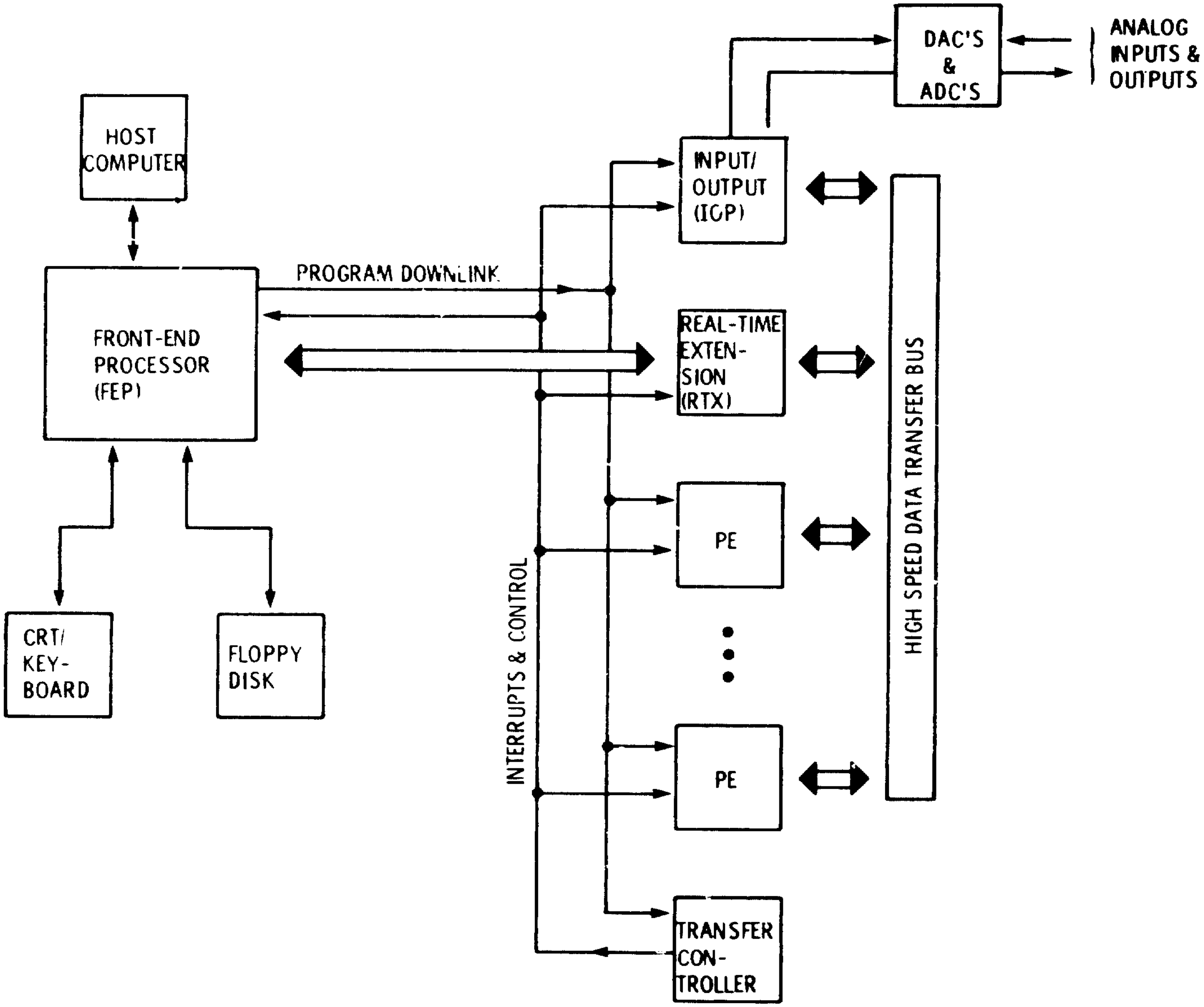

Figure 5. - LeRC real-time digital simulator structure.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

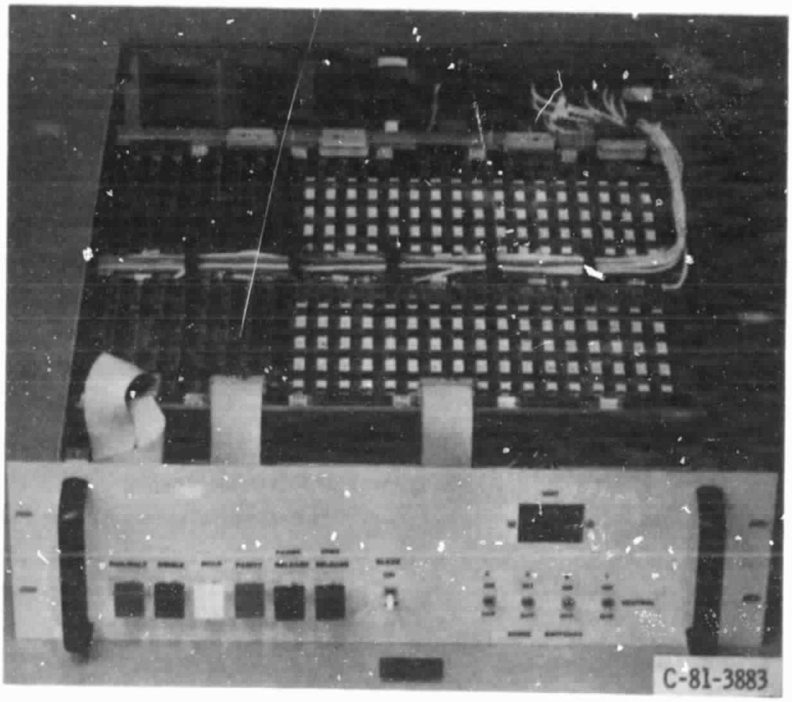

Figure 6. - LeRC digital simulation computer.