AEROSPACE INFORMATION REPORT

AIR4548™	REV. B
Issued 1995-12 Reaffirmed 2013-04 Revised 2022-01 Superseding AIR4548A	

Real-Time Modeling Methods for Gas Turbine Engine Performance

RATIONALE

This document has been updated to include an advancement in aerothermodynamic real time model transient simulation methodology. In recent years, there has been an emphasis on using a single model for both non-real-time and real-time simulation, without any loss in accuracy and without the extra software maintenance efforts associated with having a separate real-time model. Discussion of a method to optimize the equation structure of the non-real-time aerothermal model for real-time execution is included.

FOREWORD

A Real-Time Engine Model is defined as a transient performance computer program whose engine outputs are generated at a rate commensurate with the response of the physical system it represents.

Real-Time models were first used in hardware and software development of engine and aircraft controllers. The early models were relatively simple and primarily analog devices. As model complexity increased to satisfy more demanding requirements, analog models became too costly and difficult to use and digital models became predominant. The early digital models were simply digital implementations of the analog models. As digital computer capabilities increased and costs decreased, the number of real-time model types expanded to best utilize these resources.

Current practices vary greatly in terminology and methods depending upon application. This document is intended to provide a vehicle for presentation of model types and definitions to be used as a basis for communication between supplier and customer. It is also intended to complement AS681 and ARP4148.

TO PLACE A DOCUMENT ORDER: **Tel: 877-606-7323 (inside USA and Canada)**
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
SAE WEB ADDRESS: **http://www.sae.org**

For more information on this standard, visit
https://www.sae.org/standards/content/AIR4548B/

TABLE OF CONTENTS

1. SCOPE

This SAE Aerospace Information Report (AIR) provides a review of real-time modeling methodologies for gas turbine engine performance. The application of real-time models and modeling methodologies are discussed. The modeling methodologies addressed in this AIR concentrate on the aerothermal portion of the gas turbine propulsion system. Characteristics of the models, the various algorithms used in them, and system integration issues are also reviewed. In addition, example cases of digital models in source code are provided for several methodologies.

1.1 Purpose

The purpose of this document is to provide a source of information on current practices and procedures for developing real-time turbine engine models, not models of controls or other installation subsystems. This document is intended to be a basis for communication between supplier and customer.

1.2 Overview

Current applications of real-time engine models include development and testing of control systems, embedded software for control systems, flight simulators, and engine/airframe integration. The implementation of these models must balance the constraints of cost, execution time, consistency with the system being modeled and the capability to be scaled to match new information. Common engine model types are: aerothermodynamic (aerothermo), piece-wise linear, and transfer function. The real-time models require supporting mathematical utilities such as integrators, optimizers, transfer function algorithms, and data handlers. A real-time model of a turbine engine powerplant often requires models of the control system, lubrication system, engine starting system, ignition system, and others.

2. REFERENCES

2.1 APPLICABLE DOCUMENTS

The following publications form a part of this document to the extent specified herein. The latest issue of SAE publications shall apply. The applicable issue of other publications shall be the issue in effect on the date of the purchase order. In the event of conflict between the text of this document and references cited herein, the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

2.1.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or +1 724-776-4970 (outside USA), www.sae.org.

ARP4148 Gas Turbine Engine Real Time Performance Model Presentation for Digital Computers

AS681 Gas Turbine Engine Steady-State and Transient Performance Presentation for Digital Computer Programs

AS755 Gas Turbine Engine Performance Station Identification and Nomenclature

2.2 Definition of Symbols

Nomenclature and symbols used in this document are consistent with AS755.

∂	partial differential operator
A	matrix of partial derivatives relating state derivatives to states
B	matrix of partial derivatives relating state derivatives to input variables
C	matrix of partial derivatives relating output variables to states
CP	specific heat at constant pressure
d	differential operator

D	matrix of partial derivatives relating output variables to inputs
h	convection coefficient
H	specific enthalpy
P	total pressure
QU	heat transfer rate
s	Laplace operator
T	total temperature
t	time
U	model input vector
W	mass flow rate
WF	fuel flow rate
x	unit of length
X	engine state vector
$\dot{X}$	engine state derivative vector
XJ	polar moment of inertia
XL	length
XN_	rotational speed (where "_" designates a specific shaft)
XN_C	corrected rotational speed (where "_" designates a specific shaft)
XNU	rotational acceleration
Y	model output vector
Δ	difference operator
γ	ratio of specific heats
π	pi, 3.14159
τ	time constant
θ	referred temperature
Θ	arbitrary variable
1, 2, 3...	engine station designation
AMB	ambient
b	base operating point
film	film coefficient

gas gas property

HeatSoak thermal capacitance of engine parts

H high pressure rotor

in entering control volume

lag lag transfer function

lead lead transfer function

L low pressure rotor

mb main burner

metal metal property

out exiting control volume

ss steady-state

TM temperature of metal

2.3 Definitions

A/D or ADC (ANALOG-TO-DIGITAL CONVERTER): An integrated circuit that resides on a digital bus, monitors the voltage on a high-impedance input signal pin and generates a number in an on-board register proportional to the voltage on the signal pin.

ACTUATOR: Typically hydraulic devices which cause mechanical movement of engine hardware such as stators and a variable nozzle.

ADAPTIVE ALGORITHMS: Algorithms which change based on the past history of inputs and outputs.

AEROTHERMO: Abbreviation for aerothermodynamic.

BACKGROUND (Background Task Processing): In multirate real-time digital machines, nontime critical tasks run whenever processor time is "left-over" between frames. This includes any output monitor port updates, engine trend monitoring algorithms, system data logging, etc.

BITE (BUILT-IN-TEST-EQUIPMENT): New generation aircraft have systems and avionics that, when prompted, perform internal automated test procedures and store or provide systems status and test results.

BLADE ELEMENT THEORY: Analysis of fluid-blade interaction based on one or more streamtubes passing through the blade row.

BLOCK DIAGRAM: Simplified representation of an algorithm or piece of an algorithm as a series of transfer functions.

BREADBOARD: Nonproduction engine or system control which can be easily modified.

CAA: Civil Aviation Authority

CMC: Central maintenance computer

CONTROL BENCH: Facility for testing of an engine control using a model or other artificial input in lieu of a hardware engine.

COUNTS: The units associated with the digital numbers that go into or out of D/A and A/D converters. A digital-to-analog converter with "12 bit resolution" can represent voltages as varying from 0 counts (0 volts) to 4095 counts (10 volts) since $4095 = 2^{12} - 1$.

CURVE DATA: Data in tabular, spline, or polynomial form.

CYCLE MODEL: A computer simulation of an engine cycle normally consisting of a specific set of component and process representations. Typical components are compressors, combustors, turbines, and nozzles. Processes include friction losses, heat transfer, and mixing.

D/A or DAC (DIGITAL-TO-ANALOG CONVERTER): An integrated circuit that resides on a digital bus, receives digital data off of the bus, and produces an analog voltage signal proportional to the magnitude of the data. Typical D/A's produce 0 to 10 VDC at a small output current (<500 mA).

DERIVATIVE MATRIX: A vector of analytical or fixed coefficients (C's) which are used to minimize cycle balance errors (Y's) by updating the cycle balance guesses (G's), i.e., $G_i = G_i * (1+\Sigma C_{i,j} Y_j)$. The analytical coefficients are usually needed when variable areas or high Mach numbers exist between the error and guess locations of the engine.

DISTRIBUTED PROCESSING: Processing on a group of networked computers as opposed to a mainframe computer.

EEC: Electronic engine control

EGT: Exhaust gas temperature

EMBEDDED MODEL: An engine model included as part of a larger digital control application.

EPR: Engine pressure ratio

EXPERT SYSTEM ALGORITHMS: List processing or decision tree algorithm as opposed to a conventional computational algorithm.

FAA: Federal Aviation Agency

FADEC: Full-Authority Digital Engine Control

FAR: Fuel-air ratio

FAST CURVE READ ROUTINES: Curve read routines (for unequally spaced tables) in which the pointer location used for interpolation is remembered from the previous evaluation and can be moved only one location per time step. Also: Curve read routines designed to take advantage of the use of equally spaced tables.

FMC: Flight Management Computer

FOM: Figure of Merit

FOREGROUND: Foreground task processing. In multirate digital controllers or simulators, the time-critical control tasks are known as foreground tasks while all other tasks are called background tasks.

FRAME RATE: The base rate at which a digital simulator or control updates the output values.

HEAT SOAK: The transfer of energy between the gas path and the engine hardware.

HMU: Hydromechanical Control Unit

Hz, Mhz: Hertz, mega-Hertz

IATA: International Air Transport Association

ICD: Interface Control Document

INTEGRATED FLIGHT PROPULSION CONTROL: Control which is designed to manage both the engine and aircraft to the benefit of both as opposed to conventional controls where the aircraft and engine controls are designed separately.

IRON BIRD MODEL: Model used in developing the aircraft control system where higher fidelity and nonstandard conditions are expected.

IVMS: Integrated Vehicle Management System

kB, MB: Kilo-bytes(s) and mega-byte(s)

LRU: Line Replaceable Unit. Any aircraft component that can be removed and replaced on the flight line.

mA: milliamps

MICRO/MINI-COMPUTER: Computer class between a mainframe computer and a personal computer.

ms: milliseconds

MULTIBYPASS ENGINE CYCLES: Cycles in which bypass flows may be separated from the core flow (individually or collectively) at two or more stations in the compression process. These flows may subsequently mix (partially or completely) with each other and/or the core flow to provide turbine and afterburner cooling.

MULTIRATE: A real-time digital computer's program may have several distinct tasks to perform in a real-time manner. To optimize the system performance for a given processor, some tasks may run less than once every frame.

OBJECT-ORIENTED: Software design approach in which software pieces have clearly defined inputs, outputs, and attributes which facilitate reuse.

PARALLEL COMPUTATION DEVICE: Computer configured to allow pieces of a program to be executed in parallel on multiple processors in lieu of serial execution of the program on one processor.

PIECE-WISE LINEAR MODEL: Model based on a set of matrix equations that describe the variations about a base operating point, also referred to as state-space model and state-variable model.

RTD: Resistive temperature detector

SENSOR: Electrical or other processed signal of an environmental or engine condition such as temperature, pressure, or rotor speed.

STATE VARIABLES: Variables that represent energy storage, such as rotor speeds and metal temperatures.

STATE-SPACE MODEL: Model based on a set of matrix equations that describe the variations about a base operating point (same as piece-wise linear model).

STATE-VARIABLE MODEL: Model based on a set of matrix equations that describe the variations about a base operating point (same as piece-wise linear model).

THROUGHPUT: Digital information transfer rate.

TRANSFER FUNCTION MODEL: Simplified dynamic model which approximates the system based on a ratio of LaPlace polynomials.

UPDATE TIME: Time between updates of the model with sensor data, environmental or other information; interval within which model execution typically must be completed.

VDC: Volts, direct current

VOLUME DYNAMICS: The time-dependent response of the thermodynamic properties within a confined volume. Simulation of volume dynamics is usually associated with high frequency phenomena (such as surge) or long pipes and ducts.

3. REAL-TIME MODEL APPLICATIONS

A wide variety of specific real-time model applications result from the categories of current uses (see 1.2). The overall objectives that each application is intended to address is discussed in the following sections. In addition, the specific requirements that each application must meet are also discussed in terms of four attributes:

a. Consistency: Refers to the accuracy of the model output as compared to the reference data base.

b. Versatility: Refers to the adaptability of the model to match new data, to simulate performance over a wide range of normal and abnormal events, and to the adaptability of the model to new or upgraded applications.

c. Bandwidth: Refers to the range of frequencies for which the model accurately represents engine transient performance.

d. Execution Rate: Refers to the rate at which the model software must execute to produce accurate and numerically stable results.

3.1 Control Development Benches

Mechanical systems are expensive to develop; a real-time engine digital simulation driving generic, nonflightworthy actuation equipment allows on-line design modification during development at a relatively low cost. Development using a strictly hardware-based cycle incurs large delays early in the program waiting for hardware modifications that themselves will not be final. Early in development, simulations of components can be used until prototype hardware is available; the bench component simulations can themselves be used to develop the control components' final configuration. Digital component simulations can be designed to interface not only to a control bench, but also to the real engine. This enables early support of engine testing with pre-prototype "proof-of-concept" component operation before flightworthy hardware is available.

3.1.1 Objectives for Control Benches

The principal objectives of control benches include the following:

Reduction of Development Cost and Cycle Time: The main benefit to be had from the use of a complete, integrated, control system bench facility is that irregular behavior of both the complete system and its individual components can be identified and corrected off-engine, often before an engine is available for control system development testing in engine test cells. Separate component-level facilities provide the same level of functionality as the system bench component by component, but can never reveal the behavior of the components once they are interacting as a system with the other control components on the engine.

Principal motivations for a control bench simulation center around the gathering of real hardware performance data without the cost and complexity of engine testing. While a full-system control bench can look expensive, the cost is small in comparison to that of building and maintaining development engines. Further, the complexity of modern gas turbines typically hold down engine test time for controls work during early development testing. The devotion of engine hardware assets to control development tasks not only increases the control system development costs, but also reduces the availability of engine time for other testing early in a development program. Use of a control bench can reduce development cycle time by focusing effort on the control system without the distraction of engine development problems.

Development Using the Complete System: The control bench concept is more than just a digital simulator. It is a system of all the devices making up the control system and/or external lubrication system components (oil tank, etc.) on an engine plus all the hardware needed to produce mechanical loads, attitudes, temperatures and pressures an engine would exert on the components. The system can be more accurately tested by interfacing the engine model with actual hardware. On a control bench, these devices are actual, working engine control subsystems connected to mechanical loads, pressures, and temperatures, etc., that not only give the control an accurate "feel" for what will happen on the target system, but also facilitate development of those components.

Using these devices on the bench allows the control logic and parameters to be modified almost continuously until the desired performance is achieved; all before risking engine hardware with the control. It also opens up the possibility of real-time operability optimization using real components without the danger of damaging long-lead time engine parts.

a. Hardware in the Loop on Control Benches: A simulated engine interface is required for each piece of hardware connected "in the loop". Interfaces consist of:

1. Pertinent sensed parameters such as rotor speed, pressure, temperature, and flame or flame simulation for combustion detectors

2. Supply fluids such as cooling flow, pressurized hydraulic fluid, lubrication circuits

3. Loads such as pressurized fuel and mechanical actuation or reaction forces and heat

4. Electrical power supply

5. Any other physical conditions which must be included to achieve the necessary system fidelity

A desirable feature of a control bench is configuration flexibility. A modular design approach to control benches allows any desired mixture of simulated and real components on the bench. This strategy enables the bench to be a general purpose tool that solves many problems common to all types of turbine engines. Topics to consider when designing a bench include:

1. Intended Use Definition: Visualization and definition of a control bench's features and capabilities before hardware is built reduces rework from undersizing and avoids extra costs due to oversizing or redesigning a bench design.

2. Minimum Required Hardware and Dependencies: Some hardware is more efficiently included in a bench only if it is available as part of a larger assembly, such as a pump + Hydromechanical control (HMU) assembly. For example, if the HMU is to be simulated, it may be best to simulate the pump as well. Further, the relationships between control components and bench driver hardware define bench modules (e.g., pumpdriver + fuel supply + pump, thermocouple simulator + thermocouple harness connectors, air supply + supply control + pressure transducer, etc.).

3. Optional Hardware Components: Optional hardware must be grouped into dependent modules. Once this is accomplished, simulations can be defined for each component module. Many components are independent of the others; for example, an inlet pressure transducer is independent from the oil pump. Grouping modules by dependencies also further enhances the utility of the bench: different tests can conceivably be run simultaneously on independent subsystems.

4. Electrical Interfaces: The necessary electrical interface simulations (voltage, current, source/sink) for system electrical work must be included as part of a bench module; this can take the form of an electrical source/sink network or a software simulation using D/A and A/D converters.

Reconfiguration of the control bench may be expedited by simulating a component when a real piece is not available. Simulation of components is critical in keeping control work schedules in parallel with hardware development. In order to create a useful simulation of a piece of hardware for use in development, the sub-system must have the equivalent of a completed mechanical layout. This intricate mechanical layout is then implemented in a digital simulation. Each simulated device's configuration can then be rapidly changed, if desired, by the addition of model configuration variables in the simulation.

3.1.2 Requirements for Control Benches

Requirements for real-time engine models used for control development benches are discussed below.

a. Consistency: The primary consideration for the control bench is its engine model, or how closely the bench approximates the target system. If the model is too complex for the processing capability of the bench, the bench will take too long to complete its loop and will feed the control inaccurate or "old" information, resulting in low dynamic bandwidth, low fidelity and ultimately, low utility. On the other hand, if the model runs rapidly but is lacking in sufficient complexity, bench test results can lead to unnecessary hardware modifications due to a false picture of component effectiveness.

b. Versatility: As both engine and control systems evolve during the design, the real-time model must be capable of parallel growth. In particular, the model must be capable of being readily modified to interface with new control software and hardware added to the control loop.

c. Bandwidth: The real-time engine model must be capable of simulating all transient behavior of the real engine that the actual control is capable of sensing and controlling.

d. Execution Rate: With modern equipment, the model is usually able to run to the full accuracy required on every foreground pass of the control; that way the controller and components will have the best information available. Since this model is an approximation of an analog system (the real engine), there is a certain amount of inherent inaccuracy in using a discretized model. With the faster updates, this inaccuracy will be minimized. The problem of update time versus accuracy is one that cannot be determined on an external basis. It must be decided after careful frequency-domain analysis and consideration for the different systems that are to be implemented.

3.2 Integrated Flight/Propulsion Control Evaluation Tools

Some aircraft systems integrate the flight and propulsion controls. An engine model is required to properly design and evaluate this control system. In a global approach, the engine model becomes part of the aircraft model and the aircraft model is used to design an aircraft/engine controller. Another approach is to use the engine and aircraft models to define generalized transfer functions and the flight and propulsion controller is designed using the individual models.

There are three principal categories of integrated flight/propulsion control evaluation tools:

a. Digital Simulation: The entire system is simulated on a digital computer. This type of simulation is generally used for preliminary design work, particularly for control law development.

b. Electronic Bench: Checkout of the actual engine control system software with the simulation of the engine, aircraft, actuators, etc. Interfacing avionics will either be incorporated in the bench or simulated.

c. Iron Bird: Checkout testing of the system hardware and software. This is sometimes referred to as a "wet" bench.

3.2.1 Objectives for Engine Models in Flight/Propulsion Control Evaluation

The objective is to provide a real time engine model to support the flight/propulsion control system simulation. The flight/propulsion control system simulation is an engineering tool used in the design of integrated aircraft control laws. One example where these evaluation tools are utilized involves autoflight control development, and particularly autothrottle control law design.

3.2.2 Requirements for Engine Models in Flight/Propulsion Control Evaluation

Requirements for real-time models used for flight/propulsion control evaluation are discussed below:

a. Consistency: The interface of the engine control and indication system with the other aircraft systems must be simulated per design. The accuracy requirements for steady-state and transient performance vary with the application. These requirements dictate whether secondary effects such as bleed air extraction and horsepower extraction need to be modeled.

b. Versatility: When using these tools in flight control design, problems typically are found during flight testing that will require improvement to the simulation to assist in the control development. The model will have to evolve with the flight control system as more and more tests are conducted, test data becomes available, and the control laws are modified (autothrottle is a common example).

c. Bandwidth: The requirements for bandwidth vary with the application. These requirements are dictated by overall system considerations.

d. Execution Rate: The configuration of the lab, and the type of tests to be conducted, will determine the type of real time engine model to be used. Note that if the engine control system is simulated then the engine control system model must operate at the update rate of the simulation. As a minimum requirement, critical interfaces between the control and actual engine model software must operate in this manner, although it is possible to run different portions of the model at different rates.

3.3 Embedded Software for Flight Systems

The engine is a sub-system of the overall aircraft system. As such, embedded engine models can be used to improve the performance and controllability of the aircraft through the use of predicted engine performance and response characteristics.

3.3.1 Objectives for Embedded Engine Models

Embedded engine models provide engine performance characteristics in the flight management system for use in optimizing aircraft flight profile. In addition, embedded engine models provide engine response characteristics in the flight control system (i.e., autothrottle and flight control computer) for use in aircraft control logic.

3.3.2 Requirements for Embedded Engine Models

Requirements for embedded real-time engine models for flight systems are discussed below.

a. Consistency: The accuracy requirements for steady-state and transient performance vary with the application. These requirements dictate whether secondary effects such as bleed air extraction and horsepower extraction need to be modeled.

b. Versatility: The structure of the model should be such that upgrades, due to engine modifications that affect performance or dynamics, can be readily accomplished.

c. Bandwidth: The requirements for bandwidth vary with the application. These requirements are dictated by overall system considerations, for example when the real-time engine model is being used closed loop with an autothrottle the model must be able to respond to the frequency content of the autothrottle input or flight management system.

d. Execution Rate: The requirements for size, memory, and update rate determine the type of real time engine model to be used, i.e., transfer function, piece-wise linear, or aerothermo. For example, a model used in optimizing flight profile might only require steady-state relationships whereas an integrated flight and propulsion control system would require a dynamic real-time engine model. The dynamic model could be as simple as a transfer function model or as complex as an aerothermo model, depending on the system requirements.

3.4 System Model Within Engine Control

The rapidly expanding area of system models within an engine control is limited by processor speeds. Recent applications of embedded models within engine controls include:

a. Fault Detection: Engine parameters are modeled as a function of other sensor values to determine if a sensor is reading an "unreasonable" value. If test criteria are exceeded, the control will switch to other sensors or use the model value with a possible reduction in control response or functionality.

b. Sensor and Parameter Synthesis: Sensors are often too expensive or impractical to install on an engine, so sensors measuring other parameters are used in conjunction with a model to produce a "synthesized" value. That synthesized value is sometimes more accurate than a real sensor due to noisy environment, poor installation options, etc. This reduces the instances of multiple sensors used for redundancy in fault accommodation.

c. Performance Tuning: Using a sufficiently detailed engine model in conjunction with airframe data (angle of attack, sideslip, etc.), a control can achieve stability or surge margins which can selectively enhance performance or engine life.

d. Trend Monitoring/Maintenance Aid: Self-diagnosis and maintenance scheduling can be achieved if a reference cycle is compared against actual engine data in real-time using expert-system algorithms. Deviations are not only noted, but control set points are re-optimized on-the-fly to maintain the engine at peak performance for its own current state of wear and adjustment. These algorithms can also allow maintenance departments to schedule major maintenance accurately and correct bad trends before they result in premature part replacement.

e. Adaptive Model: Engine model with adaptive algorithms can be used for all the above applications under real time environments.

3.4.1 Objectives for System Models Within Controls

The reason a system model would be included in a control must come from a need for increased performance or improved availability and cost-of-ownership. If it is too costly or impractical to change a design or an existing fleet in the field to achieve these objectives, a system model should be considered to improve some performance or reliability situations.

The different models can be of different designs; each one optimized for the amount of detail needed by the customers algorithms: a Trend Monitoring system that tracks long-term performance changes can use a simpler engine model that runs as a background task, while a real-time cycle performance optimization controller may need a highly accurate and detailed model to deliver the desired improvements to engine operation.

Using an embedded model as a data reference can result in the need to simultaneously run two or more system models in real time. This situation can come about when one model, used for performance optimization and fault accommodation, tracks the real engine's health and performance with adaptive algorithms that determine performance modifiers. Since the model is now "adapted" to the real engine, that model can no longer provide data on the reference engine cycle's performance. Another model must run concurrently using nominal component performance to provide the reference standard for built-In-test and trend monitoring functions.

Wide variations in computer capabilities require tradeoff decisions to be made between computer cost, computer speed, and acceptable performance levels. If a selected design delivers inadequate throughput, the control designer may have to use a quicker-running (and less precise) model so that the selected controller hardware can be used.

3.4.2 Requirements for System Models Within Controls

Requirements for real-time models embedded within the engine control are discussed below:

a. Consistency: The steady-state accuracy of a model is purely a function of the mathematical representation of the system. However, the transient accuracy is not only a function of the mathematical representation of the system but also the update rate selected (delay distortion) and the accuracy of the numerical integration scheme selected to update model states (Euler, Trapezoidal, Runge-Kutta, etc.). Optimal consistency is achieved within the control using digital data extraction from the control feedback parameters and independent engine data to compare to data generated by the system model within control and then adjust model parameters to match real data. As the system model is usually intended to detect real engine deviations in flight from the model's output (for further action by the control), the consistency (and the accuracy) is more important than for bench simulators.

b. Versatility: For digital implementations, the software language is the key to allowing adaptation to new configurations. Beyond physical changes, design documentation must be required to deliver a description of the process used to create the model. Adaptation and update procedures then can be made based upon the original design. It is the procurement document's author who is responsible for specifying any process that is required to allow for simple updates or adaptations subsequent to initial delivery.

c. Bandwidth: The bandwidth is primarily determined by the update rate (real-time step size). Secondary bandwidth effects are introduced by the choice of numerical integration schemes, since each scheme has a calculable distortion level at a given step size (update rate) for the highest frequencies passed through the model. The distortion, defined by the output error compared to a reference output, is a function of both the delay in the frametime selected and the accuracy of the numerical integration algorithm used for the system being modeled.

d. Execution Rate: A minimal system model typically consists of a 32 kilobytes (kB) (code+data) piece-wise linear or transfer function model. This model can provide basic information for 5 to 10 variables of interest on typical steady-state values and information on rotor speed and thermal transient behavior. Size of a system model above entry-level increases steadily with increasing levels of modeling detail.

   As a rule of thumb, the update rate should be 10 to 20 times higher than the highest frequency being modeled in the system (e.g., 2 Hz => 20 updates/s or 50 ms/update). The rigorously calculated update rate is sized to provide a model of the system that responds to the desired frequency range with a given level of distortion (this problem is similar to home audio performance specifications, only using digital circuits). The minimum update rate can be calculated from the allowable distortion in the model and the integration algorithm being used.

   The update rate calculation combined multiplicatively with the size of the model is used to determine the computational performance requirements for the simulation computer's processor. It is likely that initial estimates of model size will be too small, so a processor with 2 to 3 times execution margin should be selected to avoid re-development of the processor module.

3.5 Engine/Control Models in Flight Simulators

Current flight simulator applications fall into three general categories which include engineering development, crew training, and maintenance training simulation devices.

a. Engineering Development Simulators: These simulators are usually used for "man in the loop" studies, which include overall aircraft handling qualities, cabin design studies, and procedural studies for simulated emergency scenarios. In particular, during accident studies, the scenario is replayed in the simulator to assess the crew's performance and the subsequent aircraft behavior. Also, studies can be made regarding changes to the aircraft's configuration, including the integration of avionics and display systems.

b. Crew Training Simulators: These simulators are used to train crew members in the proper use and control of aircraft systems, including normal and emergency procedures. Further, these devices are also used for instruction in the theory and operation of specific aircraft systems and their components. In these devices a multitude of system failures (including propulsion systems) can be caused, resulting in realistic cockpit indications and cues. The added versatility along with the ability to automatically select initial conditions, distinguishes crew training from the more basic engineering development simulators.

c. Maintenance Training Simulators: These simulators are a relatively recent simulator application. Each new generation of aircraft has featured more automation in the flight deck in the form of digital computers and associated actuators, and vastly increased performance and equipment monitoring capabilities. A current trend allowing flight crews to activate built-in test equipment (BITE) from the flight deck, and to receive results, has resulted in the ability to use the flight simulator as a maintenance training device. On these devices, the overall system models must be comprehensive in scope, so that all recorded and displayed information is accurately represented.

3.5.1 Objectives for Propulsion Models in Simulators

The objectives of a real-time engine model must consider the flight simulator's use as an engineering development tool, or as a crew training device, or as a maintenance training device. The real-time model must be of adequate complexity considering the number of interfaces necessary to address the simulator device's application. In general, real-time engine models used in flight simulation devices are composed of both performance and systems models, as needed for the specific application. In the flight simulator application, the real-time model must also consider systems and performance other than those related directly to the gas generator components and the controller. For example, the engine oil and lubrication system and associated parameter sensing hardware must be an integral portion of the complete model. These items are further discussed in the following paragraphs.

In addition to the model of the gas generator performance, a model of the engine ancillary systems such as oil, thrust-reverser, accessory gearbox, fuel, anti-ice, and start systems is very important in flight simulation. These system parameters are monitored by aircrews, and many warning and caution annunciations in an aircraft cockpit are related to these ancillary systems. As an example, for the oil system, models are needed to simulate pressure regulators and heat exchangers within the oil fluid loop. A rigorous model for this system is needed since many malfunctions involve the oil system, such as oil leaks, pressure regulator failures, bypass valve failures, and heat exchanger failures. The indicated values of these parameters are another important health monitoring tool used by the aircrews.

The propulsion system indications need to be modeled sufficiently to provide realistic pilot cues. An example of this is the modeling of data generated by the engine electronic control (EEC), where displayed indications may not reflect actual parameter values. Often, processed data is lower-limited, which provides totally different indications during start transients. Also, time lags associated with sensors such as thermocouples and pressure sensors need to be considered. Many aircraft are equipped with engine vibration monitor systems, requiring accurate models for the appropriate vibration amplitudes.

In the case of engineering development simulators, propulsion simulation often only requires a performance model for support of aircraft handling properties studies. The basic interface with the flight dynamics simulation is the primary focus for the propulsion model. This model can be a very basic transfer function model with base parameters expressed as functions of power setting demand. It is possible that the thrust setting parameter (i.e., engine pressure ratio, EPR or low rotor speed, XNL) and net thrust are the only outputs of this simulation. Aircraft response and crew action analysis can be quantified, even during accident studies, with very basic performance models. However, if the engineering simulator is utilized for studies of aircraft systems integration, then a combined performance and ancillary systems simulation is required. Even in the case of aircraft handling studies, this more comprehensive modeling approach may be required. For example, the loss of an engine impacts other aircraft systems such as electrical, hydraulics, and pneumatics. The effect of the engine failure on these systems often will result in further degradation in aircraft control which must be considered. Therefore, further interfaces must be modeled which increase the scope of the propulsion simulation model. This more detailed simulation must often support studies of the integration of cockpit displays, controls, and associated avionics. For the situation of an accident investigation, it must be determined whether or not the displays and avionics were providing pertinent data to the crew, both to identify and respond to the initiating failure. In general, propulsion models for engineering development simulators need to address critical interfaces between the propulsion and other aircraft systems for evaluation of aircraft in-flight performance and response to major (or catastrophic) failures.

The application of flight simulation devices to the training of aircraft flight and ground crews invokes requirements on propulsion models that must meet or exceed those appropriate for the engineering development application. Depending upon the type of training to be performed, the system aspects of the model may be more critical than the performance ones. Many simulator devices are utilized for flight crew training in systems operation, and both normal and emergency cockpit procedures. In this application the primary objective of the propulsion model is to provide representative response to instructor-induced failures (or malfunctions) and interfaces with other aircraft systems. Performance needs to be representative of the actual engine, only to the degree necessary to support the interfaces with aircraft avionics, systems, and the flight performance. Any displays or annunciations related to the propulsion system will stipulate the minimum accuracy demanded of the performance portion of the model. Simulation of engine starting is a standard feature for crew training simulations that is not necessarily required for engineering development devices. Total mission training with a simulator device will require modeling of engine malfunctions and their associated causal effects. This is because cockpit indications are a very important cue, as well as the aural and motion cues, for realistic crew training. These requirements are enforced by regulatory agencies (such as the Federal Aviation Administration, FAA; Civil Aviation Authority, CAA; etc.) that monitor the training of commercial airline crews. Further, the International Air Transport Association (IATA) defines the needed data and model accuracy necessary to meet the regulatory agencies' requirements.

Accurate display of propulsion systems data in the simulated cockpit serves a two-fold purpose in the flight simulator environment. First, realistic indications of engine parameters are monitored by the flight crew for establishing power settings to attain specific flight profiles, and also to assess whether or not a specific powerplant is operating in a normal fashion (i.e., monitoring for possible propulsion system failures). Second, many avionics systems onboard the aircraft rely upon information from the engine control system as well as performance data. In particular, avionics such as autopilots, flight management computers (FMC), data acquisition unit (DAU), integrated avionics computer (IAC), engine indications and crew alerting system (EICAS), and central maintenance computers (CMC), depend upon full-authority digital electronic control (FADEC) and engine performance data to effectively perform their functions. Often, the actual aircraft Line Replaceable Units (LRUs) are utilized in the simulator so proper propulsion system inputs to these units are essential to ensure that the boxes will operate correctly.

Further, the use of simulators in maintenance crew training increases the scope of propulsion systems simulation. Other interfaces not normally observable by the flight crew are now needed as part of the model. In particular, BITE testing of specific avionics LRUs, including engine FADECs, will require the simulation of propulsion parameters to provide the indications and annunciations that may be accessed either in the cockpit or in some other region of the aircraft. These panels are simulated by displays at a remote instructor's station. Further, systems schematics are displayed at the instructor's station that illustrate the lay-out and components of specific systems. These displays will change in real-time in response to actions taken inside the simulator cockpit environment. The propulsion model must contain models of propulsion systems down to the wire and relay level of detail for subsystems such as the ignition harness, the starter system, the reverser system, and the fuel control system. In this manner, the instructor can very readily explain the consequence of very specific failures (or malfunctions) related to the actual system or component, and also explain the ramifications of crew action or inaction. To correctly simulate the BITE, additional FADEC functions such as input data selection logic, fault management logic, pilot initiated tests, and continuous BITE must be included in the model.

In summary, the objective for a propulsion model in a flight simulator environment is to provide a representation of the simulated engine that supports the simulator's application. This involves modeling of all pertinent interfaces to other aircraft systems and avionics, and also to all pertinent displays. Further, the model must be presented in such a way that malfunction modeling does not require additional complex algorithms to provide appropriate system and performance responses. Therefore, to some degree, the propulsion model best suited to the simulator application needs to develop from an object-oriented approach that provides simulation of both propulsion performance and systems components.

3.5.2 Requirements for Propulsion Models in Simulators

Despite the variability of the applications and objectives of flight simulators and their embedded propulsion models, there are some basic requirements for models used on these devices. Details for each of the basic requirements for these models are provided below.

a. Consistency: The accuracy required of the engine model in the flight simulator is the same as for all other aircraft systems. Essentially, the model should accurately represent the engine in terms of performance, indications, and crew alerting for all flight regimes and events. Model tolerances are negotiated with the customer, and must meet or surpass certification tolerances established by the regulatory agency (such as the FAA). Due to recertification requirements, model consistency must be such that test results are repeatable within the prescribed model tolerances.

b. Versatility: A simulator requirement for generic, propulsion real-time models is tuneability or versatility. In flight simulator applications, the simulator manufacturer will need to add many "enhancements" for simulating a multitude of component failures (e.g., rotor bearing seizure) for both the gas generator and the other systems.

   Propulsion models in the flight simulator application nearly always require performance simulation for starting, spooldown, and windmilling. A model that is continuous across all engine operating regimes is necessary to address this requirement.

   In relation to FAA certification requirements and annual reviews, several performance tests need to run automatically, over and over. Resetting of the simulator to specific initial conditions is a prerequisite for these automated tests. Therefore, any modeling approach must provide features for easily establishing the initial steady-state operating point.

   A further requirement for real-time engine models deals with the computational algorithms utilized within the model. In particular, algorithms for leads, lags and lead/lags must be described, including their performance limitations caused by changes in the simulation update rates. The selected algorithms should produce consistent results when used in different computer environments and at different update rates. Also, the interpolation schemes used with the various table lookups need to be defined (i.e., linear, trapezoidal, Lagrangian, etc.).

   Careful selection of state variables is of particular advantage in flight simulation. A simple example would be modeling of total net thrust calculations for a turbofan engine. Usually, total net thrust is provided as either a function of core engine pressure ratio (EPR) or corrected low pressure rotor speed (XNLC). For the turbofan, it would be more appropriate to make separate core and fan bypass stream thrust computations. In this manner, thrust vectoring and reversing, and failures like a seized fan shaft and thrust reverser malfunctions could be more readily simulated. Exhaust gas temperature (EGT) is often simulated as a simple function of EPR or corrected high pressure rotor speed (XNHC). However, simulation of stalls, surges, and hot or hung starts is difficult with simplified functional relationships. The above examples should stress that the physical relationship of component parameters needs to be, even if grossly simplified, reflective of the thermodynamic cycle.

c. Bandwidth: In general, the flight simulator engine model needs bandwidth performance that allows all engine transient responses observable by the flight crew.

d. Execution Rate: In the flight simulator environment, there is very limited available processing power due to the number of real-time simulations that must coexist with the engine model. Update rates typically range from 7 Hz for a turbofan model to 250 Hz for a detailed turboshaft/helicopter simulation utilizing blade element theory. Often, there may be only 5 to 10 ms of computing time available for the total engine simulation, including the controls. It should also be noted here that these constraints are for the complete propulsion system, which includes ALL engines on a particular aircraft. Four-engine aircraft are not uncommon.

4. REAL-TIME ENGINE MODELING METHODOLOGIES

4.1 General Model Characteristics

The design, development, implementation, and use of a real-time gas-turbine engine model is accomplished through a complex series of tasks which draw upon a variety of engineering, mathematical and programming disciplines. This section will discuss issues arising from this process that bear upon the success of a given real-time modeling project. While several of these issues are perceived to be highly critical in nature, real-time modeling, due to the functional interdependence of actual hardware and its human operators, is characterized by the need for consistently high quality. There is little in this document that can be safely ignored if the model is to successfully serve its ultimate purpose.

At the crux of the problem is the need to obtain a convergence among three competing aspects of the process: cost, accuracy, and speed. A balanced plan that optimizes the return on investment and, hence, the profitability, of a real-time engine modeling project requires a complete understanding of the interaction among cost, accuracy and speed. It should be recognized that these three factors cannot be specified independently; that fixing any two of them is tantamount to setting the third.

Each of the methodologies described in this document has characteristics that suit specific aspects of the requirements to a greater or lesser extent. These characteristics directly impact the cost, accuracy and speed of the real-time engine model. This document is intended to provide information concerning the characteristics of the various methods so that the trade-off process can be better understood. At the same time, this understanding will lead to the formulation of more coherent model requirements.

4.1.1 Requirements Specification

Real-time engine models are generally supplied as a data item that supports the sale of the engine. As such, its content is driven completely by requirements specified by the customer. The knowledge-base that the customer brings to this process greatly affects the quality of the requirements presented to the supplier. Inadequate understanding of the requirements can have an adverse impact on the model's ability to achieve its operational objectives.

Real-time models that are called upon to serve in a variety of different applications present the model designer with the greatest challenge. While a common methodology is generally sought such that each of the model's intended applications can be achieved with a single entity, there exist projects where at least two different methods are required. In this event, commonality is generally limited to the source of the real-time model data.

In the current practice, formal documentation of the characteristics of the real-time engine model and its interaction with the simulation environment is made necessary by engineering and legal reality. Of particular importance to the success of a complex real-time simulation project is the quality of the Interface Control Document (ICD) and design documents which specify the information and data necessary to install the engine model in the larger airframe or simulator system.

An ICD is the standard method for defining the inputs and outputs of any system's connections to other systems. To the extent that the ICD is maintained, adaptation to new configurations is managed through modification of first the ICD, then the actual system. When combined with a design document, the ICD will allow adaptation of an existing system to a new or modified application of the original system, in a manageable format.

Design documents that address procurement documents' requirements for adaptability will spell out the scope of work required for adaptation to new configurations. It is the responsibility of the procurement document's author to specify design methods and/or hardware that ensures a design's adaptability. Examples of methods and hardware requirements include standard interfaces, buses, links, programming languages, connector types (as well as bolt patterns, case sizes, etc.), and power inputs (like 28 VDC for aircraft applications) commonly found in the target environment.

4.1.2 Computational Requirements

The most uncertain aspect of the real-time model planning process is the choice of a computing platform. Without knowledge of the execution time and size requirements fostered by a specified level of accuracy, the customer cannot make an objective decision.

Obviously, the answer to this dilemma is to confer with the engine supplier prior to the issuance of a specification. In this way, the important characteristics of the model can be ascertained. Experience with similar applications will also play an important role in defining the computational requirements. A useful concept is to assume that unforeseen requirements will increase the computational load. This leads to the specification of a computing system that will, at least, comfortably meet the requirements of the provisional modeling methodology.

The amount of excess capacity requirement assumed must depend on the applicable experience available. First-of-a-kind applications have been known to exceed estimated computational requirements by a factor of two.

Validation is the process by which the real-time program that implements the model is shown to meet the specified requirements. This is generally expressed in terms of a test plan that defines the expected execution rates, stability and accuracy for the particular operating modes required by the applications. In addition, the operation of any special features is demonstrated by the successful execution of the test plan.

In many applications, it is necessary to adjust the data used in a real-time engine model so as to match improved representations of the actual engine performance. The ease with which this can be accomplished can be an important factor in selecting the most effective method.

The adjustment of a real-time engine model is a necessary part of the development process. It is a minimization process in which model parameters and tables are altered in such a way as to reduce the difference between the model and desired results. Generally, as an engine project moves along its development path, more representative test data and models become available. As this occurs, the real-time engine model is adjusted accordingly. In anticipation of this continuing requirement, automated validation procedures employing optimization algorithms can provide long-term savings in the effort required.

4.1.3 Input/Output

Input/output requirements are the most obvious interface items that must be addressed in a real-time engine model specification. The real-time program is generally integrated into a larger collection of software and must communicate with the programs which depend upon it. In particular, initialization and reset capabilities are generally required. Failure mode requests from the calling programs must also be processed.

The Interface Control Document is the sole source of input/output characteristics. It should be maintained concurrently by the customer and supplier, preferably electronically, and can then be the source of computer data files that are utilized by the simulation software.

4.1.4 Size Requirements

The computing equipment used in many current real-time engine modeling applications presents the program designer with a virtually unlimited storage capability. Some applications will be encountered in which this is not the case, however. When a size limitation is required, it will have the same impact as the more typical timing requirement in that it is absolute. A large discrepancy between required and available storage is likely to increase program development costs substantially.

4.2 Tradeoff Between Speed, Accuracy, and Cost

4.2.1 Speed

Real-time models must execute fast enough to provide data at rates consistent with the characteristics of the real systems to which they are connected. This is an absolute requirement; if the program does not execute rapidly enough, the system cannot operate properly. The speed of the program is determined by the number and complexity of calculations required by the model, the execution rate characteristics of the computer processor, and the compiler.

In some instances, the speed of a model proposed for a real-time application on a processor other than the one where it is implemented must be estimated. This is generally accomplished with simple benchmark programs that can be easily coded and executed on each processor. Timing test data can then be used to estimate the execution rate for the real-time engine model.

Timing estimates can also be used to specify the execution rate of a processor required to support a given implementation. In this way, the computational requirements for differing methodologies can be compared and used to support design decisions.

4.2.2 Consistency and Bandwidth

The accuracy of a real-time engine model is characterized by its subjective nature. While the term accuracy implies an objective numerical measurement of the differences between the model and the system it is intended to represent, it is impossible to relate accuracy measurements directly to the real issue: How consistent with reality must the model be in order to successfully meet its operational requirements? In the face of this dilemma, an attempt will be made to define the manner in which model consistency is determined and how accuracy measurements are used in the design and development of real-time engine models.

Accuracy measurements should reflect the requirements of the system in which the model will function. Therefore, the design of an accuracy algorithm, like the model itself, should depend on as much knowledge as is available concerning the application. This information should be used to assign priorities to the several different aspects of the model's accuracy. The algorithm should produce a single value, usually termed a "figure of merit" (FOM), that can then be used to assess alternative model design approaches.

For all of this analysis, it should be recognized that the characteristics emphasized by prioritization are primarily determined by subjective means, i.e., the responses "look right". Observation of the model's transient response in comparison with the desired results is the first step in the process.

Accuracy can be assessed by categorizing the modes in which the model must operate. Generally, the two major categories are steady-state and transient. Steady-state accuracy is a relatively simple concept while transient accuracy can be measured in both the time and the frequency domains.

The steady-state accuracy of specified output variables is most important in applications where the model is interfaced with electronic engine controls in system integration tests. Fault detection logic included in the control software is sensitive to the engine performance levels that it senses. Because of the interactions that exist among the sensed variables, a measurement that averages the errors between the model and its data source is preferred.

The model's bandwidth is the range of frequencies over which accurate dynamic response is expected. It is a function of the time step employed in that higher bandwidths generally result from the use of smaller time steps. It can be estimated by analytic means and confirmed by testing the model.

Bandwidth requirements often differ for different parts of the engine model. Engine variables used for one portion of the control system may respond at much different rates than those of another portion. These requirements can lead to a design that employs more than one time step. Generally, a small part of the overall engine model is executed at a much faster rate than the major portion and the control samples the variables computed in that portion at a higher rate.

Transient accuracy can also be measured by directly comparing the model's response with the desired results. A root-mean-square measurement of the differences between the actual and desired model response is often used in the evaluation of candidate methods.

4.2.3 Cost

The cost of a real-time engine modeling project is made up, primarily, of the labor and material expended in its production and the acquisition of computing hardware. The challenge to the project planner is to balance these two factors. For example, lower-performance hardware, which is typically available at lower cost, will probably require greater effort in the preparation of the model. At the same time, computing architecture matched to a particular modeling methodology may require a greater programming effort than is necessary for conventional processors.

Experience has shown that real-time engine model development efforts are often underestimated. This may be due to a perceived need to select a methodology too early in the process. Overall project costs can be reduced by spending more time on the evaluation of alternative methods within the context of well-established requirements.

4.2.4 Accuracy Versus Cost

Real-time engine models are characterized by their reduced accuracy in comparison with other, more sophisticated, models. This is necessitated by the computational speed requirements of real-time operation. These simplified models are often defined heuristically. The process is iterative in nature in which both the model structure and the data arrangement are improved in response to the results obtained from previous versions.

Given typical execution rate requirements, then, greater accuracy involves greater cost. It is critical that both the real-time engine model supplier and customer be aware of the close relationship between the effort allocated to model development and validation and the accuracy of the resulting model.

4.2.5 Accuracy Versus Speed

This tradeoff lies in the fact that model accuracy can be improved by adopting more detailed approaches which require faster processors. Once again, accuracy costs.

Assuming a given project cost, accuracy and speed compete for the limited resource. The speed requirement is generally more rigid than accuracy. Since accuracy is often a function of the model's complexity, it is clear that the timing requirement directly limits the level of accuracy possible. For this reason, conservative assumptions in sizing the computational requirement are indicated.

4.2.6 Cost Versus Speed

Although to a lesser extent than is true for accuracy, additional effort in the design of the model can yield speed improvements. It is likely, however, that the purchase of more capable computing equipment will be more effective in achieving a balance between cost and speed while maintaining accuracy at the specified levels.

4.2.7 Risk Factors

Several factors in the design of a real-time engine model present high levels of risk. These are:

a. Ability to achieve real-time execution rates with the chosen modeling methodology

b. Reliability (convergence) of the numerical integration algorithm at the required update rate

c. Subjective evaluation of responses inconsistent with the results of the validation process

4.3 Aerothermodynamic Models

The aerothermodynamic (aerothermo) real-time engine model can best be defined as a cycle model which has been modified for transient simulations with constraints on consistency, size and computational speed. The extent of the modifications which are required will depend upon the cycle, the application and the processor. Models for multibypass engine cycles with compressible flow mixing planes are complex, which increases the difficulty of meeting the real-time model constraints. Ironically, the potential value of an aerothermo real-time model increases with the complexity of the cycle. The applications and processors which impose constraints upon the engine simulation may result in a series of real-time models for the same cycle. The application dictates how fast the model must execute and the processor determines how fast the model can execute. The size of the model is processor dependent.

The model may also be split into pieces which can run in parallel on two or more different processors. The original motivating force in the development of aerothermo real-time models was the need for a control embedded engine simulation. This application imposes the most severe constraints on consistency, size and computational speed. The simplification of bivariate curves in the component representations can substantially reduce the size of the model with very small effects on consistency. Cycle simplifications may be required for aerothermo real-time models of complex engines.

4.3.1 Mathematical Model Approach

4.3.1.1 Theory

The transient aerothermo model normally uses essentially the same representation as would a steady-state cycle model. The time dependent aerodynamic equations of motion (see 4.3.1.5) have been used for engine transients where the assumption of airflow continuity does not apply. The time step which is used with these equations must be consistent with the dimensions of the volume. Stall simulations use a time step on the order of 1 ms.

The transient model is the same as a steady-state model running to fuel flow and variable geometry with two primary exceptions. Component torques acting upon the shaft are assumed to be unbalanced and together with moments of inertia define acceleration/deceleration (accel/decel) rates which are integrated in time to determine the transient characteristics of the engine. Also the transient model requires that appropriate heat soak representations be added in order to simulate the effect of heat transfer between the gas path and metal on the engine performance. Volume dynamics are included in transient models only for special cases, e.g., long ducts, large volumes, high frequency events such as stall and stall recovery.

4.3.1.2 Form of Equations (Aerothermo)

The basic parameters of the engine model are mass flow (W), pressure (P), enthalpy (H), and fuel-air ratio (FAR). Equation 1 allows the components to alter the thermodynamic conditions of the air.

$$\begin{aligned} \text{Mass}: W_{out} &= W_{in} + \Delta W \\ \text{Momentum}: P_{out} &= P_{in} + \Delta P \\ \text{Energy}: H_{out} &= H_{in} + \Delta H_{ss} + \Delta H_{Heat\ Soak} \\ FAR_{out} &= \frac{(WF_{in} + \Delta WF)}{(W - WF_{in})} \end{aligned} \qquad \text{(Eq. 1)}$$

Component processes which alter the thermodynamic properties include:

a. Compression

b. Expansion in turbines and jet nozzles

c. Heat addition or exchange

d. Frictional pressure loss

e. Mixing

4.3.1.3 Form of Equations (Rotor Dynamics)

The states of the transient engine which are universally utilized are the rotor speeds. These states are computed by integrating time dependent equations which relate rotor acceleration rates, unsteady horsepower, rotor inertia, and speed (see Equation 2).

$$\frac{d(XN)}{dt} = \left(\frac{60}{2\pi}\right)^2 \cdot \frac{\Sigma Power}{XJ \cdot XN} \qquad \text{(Eq. 2)}$$

4.3.1.4 Form of Equations (Metal Heat Soak)

The metal temperatures of the components in the gas path of the engine are transient states which are dealt with directly by some, and indirectly by others. Lags on the outlet temperatures of some engine components (usually those in the hot section) have been applied for decades. More modern modeling methods relate the outlet temperature lags directly to changes in metal temperatures and to heat transfer rates (see Equation 3).

$$\begin{aligned} QU &= h_{film} \cdot Area \cdot (T_{gas} - T_{metal}) \\ dT_{gas} &= \frac{-QU}{W \cdot (CP)_{gas}} \\ \frac{dT_{metal}}{dt} &= \frac{QU}{Mass \cdot (CP)_{metal}} \end{aligned} \qquad \text{(Eq. 3)}$$

4.3.1.5 Form of Equations (Volume Dynamics)

A simplified form of the equations of motion which assumes low Mach number airflow can be used to define the dynamics of the thermodynamic properties (see Equation 4).

$$\begin{aligned} \frac{dW}{dt} &= \text{Area} \cdot \frac{\Delta P}{\Delta x} \\ \frac{dP}{dt} &= (\gamma - 1) \cdot \frac{\Delta(W \cdot H)}{\text{Volume}} \\ \frac{dH}{dt} &= (\gamma - 1) \cdot \frac{H}{P} \cdot \frac{\left[\Delta(W \cdot H) - \frac{(H \cdot \Delta W)}{\gamma}\right]}{\text{Volume}} \end{aligned} \qquad \text{(Eq. 4)}$$

4.3.1.6 Equation Sequence

The equations for the engine components describe how each element alters the airflow, pressure and enthalpy of the gas path. It is most convenient, therefore, to begin with the known inlet pressure and enthalpy and proceed from the front components to the rear to define the aerothermo properties throughout the engine.

After executing the aerothermo model, the unbalanced torques and metal to gas path heat transfer rates are known, so that accel/decel rates and metal temperatures may then be integrated in time.

4.3.2 Usage

The aerothermo real-time models were originally developed for engine control hardware applications. They are now being used for a variety of engineering design and development purposes at both engine and airframe companies. Aerothermo models can be used anytime that performance data and off-design characteristics are available for all the important components of an engine. They are not suitable for use if the only information available is overall engine performance data. Aerothermo models may be used for any operating regime that the engine is expected to encounter, as long as component characteristics are known or can be predicted easily for that regime. These models have been used regularly for starting, overspeed or failure mode studies, as well as for routine transients and control schedule development. In general, the more complex the engine and the greater the use of multivariable digital controls on variable geometry features, the greater the advantage of aerothermo models over the simpler types. In current practice, more than one level of aerothermo real-time model is developed for a single engine model. This allows conflicting application requirements to be satisfied. The processor constraints associated with a model embedded in a control are more stringent than those available in a flight simulator which, in turn, maybe more severe than those in a control development test cell. There are few, if any, limitations on aerothermo real-time models other than those imposed by the processor.

Models embedded in a control usually have more limited features and sub-idle or overspeed capability may not be required. Flight simulators usually require that more features be modeled, such as starters, lubrication systems, parasitic loads, and bleed systems.

4.3.3 Characteristics

The aerothermo real-time engine model has been defined as a cycle model which has been modified for transient simulations with constraints on consistency, size and computational speed. Other model characteristics include input /output, scalability, modes of calculation and range of operation. The capabilities which characterize these models cannot be standardized when the requirements and constraints are not.

4.3.3.1 Size

A real-time model for a complex cycle with sub-idle and overspeed capability can be reduced to a size of approximately 10 to 20 MB for combined engine model and control, with a similar requirement for data statements. For most applications size is not the critical constraint, and real-time models developed with object-oriented programming languages can be several times larger. The size can most effectively be reduced by limiting the range of operation, which minimizes the data required without affecting consistency.

4.3.3.2 Consistency

As with size, the application requirements dictate the consistency of the aerothermo real-time model with the steady-state cycle model. The size and execution time can be reduced by one or two orders of magnitude without compromising steady-state consistency by eliminating unnecessary calculations, replacing complex thermodynamic calculations with simplified curves, and utilizing fast curve-read routines. For complex cycles in applications with stringent size and execution time constraints, further simplification of the cycle and components may be required. In such cases the steady-state consistency goal is often quoted as ±2% on thrust, speeds, and control sensor parameters. Consistency of an aerothermo real-time model with a transient version of a steady-state cycle model is readily achieved if the model was developed to be consistent with the steady-state cycle. This is accomplished by using the same moments of inertia and heat soak characteristics.

Achieving transient consistency with engine test data can best be accomplished by executing the engine simulation using transient test data files as the source for engine input of fuel and variable areas and stators. If the data is from a test cell which does not maintain constant inlet/exit conditions, this data must also be available. Inconsistencies in the transient response of speeds and sensors can then be resolved by comparing the output of the aerothermo real-time engine model to those of the actual engine. Using this approach the goal should be to achieve a level of consistency with the test engine which is substantially better than engine to engine variations. The transient consistency goal should be ±5% on thrust, speeds, and control sensor parameters.

It is important to do consistency checks with input values of fuel flow, variable areas and stator positions, etc., versus time. Otherwise, the errors in modeling of the control system will cause perceived errors in the aerothermo model behavior.

4.3.3.3 Tradeoff Between Accuracy, Speed, and Cost

The process of developing aerothermo real-time engine models depends upon the maturity of the technology available. The cost of developing models which meet the speed and accuracy constraints should not be seen as the average cost of the model but rather as the methods technology cost with a relatively small addition for the individual models which simply apply the technology. When the technology exists there will be components available in existing models which can be combined to meet the required accuracy and speed objectives of a new model. These component models will have been developed to maximize their speed and their consistency with reference steady-state models. Flow solver methods will exist which have proven their capability to maintain the cycle balance in real-time applications and transient effects methods will be available which have been developed for speed and consistency with reference transient models. When new models can be constructed by assembling existing elements and replacing the curves and constants with those taken from a reference model, then most of the tradeoff between speed, accuracy and cost has been made in the methods technology development stage.

There will be exceptional cases where even a well-developed technology will be inadequate to meet the speed requirements. In some cases first time simplifications of complex cycles where multiple secondary flow circuits can be combined may involve increasing the cost of the model with some compromise in accuracy. In other exceptional cases the number and complexity of components may exceed the capability of the existing technology with the chosen processor. In most cases the investment choice is between a faster (including parallel) processor and improved modeling methods.

When a single aerothermo real-time model is to be developed without the benefit of adequate existing technology, the cost is still a combination of methods development and the application expense. In such cases it is tempting to compromise accuracy in an effort to increase speed and reduce cost, especially when the stated accuracy requirement of the initial application is not severe. In the development of real-time models it is difficult to save time and money by looking for shortcuts. Only if the fundamental components and elements of the model are complete and correct does the model have the potential to meet the accuracy requirements of future applications such as condition monitoring, power management, or performance-seeking controls.

In recent years, new modeling approaches have made it possible to have a single, high-fidelity simulation capable of being executed in the “conventional,” as well as in real-time mode, without any loss of accuracy and without the extra maintenance efforts associated with having a separate real-time model. Given that computer processing speed has increased dramatically over the years, it may appear that this would be trivial. In practice, however, the desire/need to model system performance at higher levels of detail outstrip the increase in computing power. Furthermore, modern object-oriented simulation systems used to simulate gas turbine systems, such as NPSS (Numerical Propulsion System Simulation), provide increased flexibility and capability compared to legacy systems, but are inherently slower than procedural systems. Execution of high fidelity real-time simulations continue to present a challenge even now.

4.3.3.4 Versatility

There are two possible approaches which can be taken in the process of developing and calibrating an aerothermo real-time engine simulation. A combination of both is preferable to either one individually.

a. To the extent that the steady-state cycle model and its transient equivalent are accepted as representative of the engine, they should be used as the data base for calibrating the real-time model and its components.

b. Evaluation and calibration of the real-time model using engine test data is a valuable supplement and may be the primary data base for the calibration. This is accomplished by executing the real-time model with engine data input and altering components to match engine data outputs.

4.3.3.5 Modes of Operation

a. Steady-State: In some applications the normal procedures for initialization of the real-time model would be used to generate steady-state solutions. The engine heat soaks are switched off and the shaft acceleration rates are treated as error terms in a simultaneous engine/control iterative cycle balance procedure. This could be the same procedure used in balancing a normal steady-state cycle model.

b. Transient: In real-time models where execution speed is extremely limited, cycle balance may be achieved using a single pass with a fixed or analytical derivative matrix. With time steps of 20 to 40 ms, adequate convergence can normally be achieved with careful choice of the independent and dependent variables and an appropriate derivative matrix. Where processor speed allows, a fixed-pass iterative scheme may be used. The derivative matrix can be determined on the steady-state initialization point and updating formulae can be determined on the steady-state initialization point and updating formulae can be employed to refine the matrix during the transient. Adequate convergence of these aerothermal models can be verified by checking results against a full iterative scheme or by executing with a reduced time step and examining the effect of the change.

   Transient execution of real time models generally requires the model to converge on a solution to the governing equations at each time step. This is typically accomplished in two ways: 1) optimize the simulation to execute as fast as possible for a single solver pass, and 2) control the number of passes the solver executes. The aero-thermodynamic real-time model execution speed is optimized by either eliminating or simplifying particular features of the component representation, followed by recalibration of the real-time simulation against the base simulation. There are a variety of approaches to reducing the number of solver passes.

   In the typical gas turbine simulation, a Newton-Raphson numerical method is used to solve the governing equations in time. The NR method is iterative in nature with multiple solver passes needed to compute the state variables at the next time step. First there are perturbation passes, which are used to generate the Jacobian matrix (which are the influences of each single independent perturbation on all dependent errors in the simulation). Jacobian matrix generation requires n perturbation passes – where n is the number of solver balances (independent/dependent variable pairs) in the model. Then, the solver runs a series of iteration passes to find the solution where all dependent errors are within some tolerance. The number of iteration passes tends to stay constant and small as long as iteration guesses are carried over from the previous time solution and Jacobian matrix is consistently calculated every point.

   For a gas turbine model with a given level of fidelity, a specific transient time step (or control update rate) and a given computing processor speed, there is a limit to the number of passes than can be executed while sustaining real-time execution speed. To remain below this limit, it is necessary to keep the number of balances (and thus the number of perturbation passes) small, or to eliminate Jacobian generation altogether. Elimination of matrix generation can be achieved by using the Broyden method (see section 6.1.4) to scale the original Jacobian matrix as the simulation proceeds, and use only iteration passes to achieve a converged solution. In this approach, it is typical to limit the number of iteration passes to ensure operation with the real-time constraint, even if the model has not converged within the solver tolerance. If update rates are small, and the quality of the Jacobian remains good, this approach can yield acceptable results. A second approach is to use pre-calculated Jacobian matrices as a function of certain cycle variables. As the simulation proceeds, the appropriate matrices are loaded into the simulation and used for the iteration passes. Note that the accuracy of this approach is dependent on the “coverage” of the Jacobians during the transient.

   In general, the accuracy of the simulation is improved by generating a Jacobian at each time step. However, even with a small number of iterations passes, this requires that the number of balances in the model is relatively small. In the past this was accomplished by reformulating the model to minimize the number of balances, which unfortunately leads to separate “conventional” and real time models.

An alternative method which focuses on the optimization of the balance structure to make the number of perturbation passes more manageable and still allow enough margin for at least several iteration passes was developed.

Balance structure of a typical turbofan engine simulation consists of 3 types of balances – continuity, power, and use-before-calculated (UBC). Continuity balances ensure the flow calculated by each component (using turbomachinery maps or based on flow parameter calculations) match the flow initially guessed and iterated upon. Power balances ensure the integrated speed value, which is calculated from overall moment of inertia of components connected to the shaft, matches the shaft speed. In steady state mode, this reduces to an iteration on shaft speed to ensure power supplied by the turbine to the shaft matches the power demand of compressors, parasitic losses and other power extractions from that shaft. Finally, "used-before-calculated" balances are defined in a simulation because certain calculations require knowledge of parameters that can only be calculated after it is used. For example, the calculation of the turbine clearance may be a function of turbine exit temperature, which is itself influenced by result of the turbine clearance calculation. The solution for the "used-before-calculated" situation is to define a balance which iterates on a guessed value of turbine exit temperature in the clearance calculation until the guessed and actual turbine exit temperature agree within some tolerance. Often, used-before-calculated balances make up the majority of overall balances in a typical gas turbine engine simulation.

Use-before-calculated balances can be eliminated from the balance structure of a transient simulation by replacing the balance with logic that sets the guess variable value for the next transient point based on the result of the actual variable from the previous transient point. For the turbine clearance example, the turbine exit temperature guess for the next transient point is set equal to the actual turbine exit temperature calculated in the previous transient point. This effectively creates transient lag, roughly equal to a single execution time step, between the high fidelity aero-thermodynamic simulation operating in real-time mode and conventional transient mode. If the time step is small enough, the effect of the lag is insignificant. Furthermore, with small time steps, the power balances can be removed by using explicit integration with no negative effect.

Applying this balance restructuring can significantly reduce the number of balances in a simulation. A typical high-fidelity turbofan gas turbine simulation with over 40 balances can be reduced to just seven continuity balances for operating in real-time mode. With a limit of 3 iteration passes, a simulation can generate a new Jacobian at each time step with only 10 total passes. Given a suitably fast computer, this should be sufficient to run the model in real time mode.

In some applications where the engine is interfaced with control hardware the system may be reinitialized at sea level static idle and automatically transitioned to the desired flight condition and power setting.

c. Sub-idle: Sub-idle operation (ground starts, air starts and spool-down) capabilities are required for some applications of real-time models. Model to engine consistency in such transients is a requirement. Accurate component off-design characteristics are the key to sub-idle capability in an aerothermo real-time model.

d. Failure Mode Analyses: Aerothermo models are readily adaptable to the real-time modeling of failure events. Special logic must be incorporated into these models if failure modes are to be analyzed. A customer-activated switch must be provided for each type of failure to be studied. Control system sensor and actuator failure provisions are common, but engine shaft failures are often simulated, as well.

4.3.3.6 Dynamic Bandwidth

Dynamic bandwidth of the engine is generally in excess of 10 Hz. For a complete propulsion model, dynamic bandwidth comparisons are not made since the engine bandwidth is greater than the control bandwidth.

4.3.3.7 Input/Output

In general, the input/output requirements can be consistent with those of the detailed model. Basic input requirements include environmental conditions (e.g., P2, T2, PAMB), engine control variables (e.g., fuel flow, stator position, variable areas, and cooling flow valve positions), and external power requirements (e.g., compressor bleed air, power extraction, starter power input). Most of the cycle values available from the detailed model are computed in the real-time model and are available as output. Those which are required but not available can be added.

NOTE: The essence of achieving real-time capability is to compute only what is required as efficiently as possible.

4.3.3.8 Data Origin

Aerothermo real-time models are normally derived from a detailed aerothermo model. The recommended approach is to construct and verify each component individually. In most cases this is accomplished simply by modifying an existing component of the same type with a new set of map curves. The translation of the curves used by the detailed aerothermo model into the desired range, density, and format can be automated to include the preprocessing of adders and scalers. If a real-time version of a component is not available, one is constructed by optimizing the logic of the detailed model's component.

4.3.4 Example Case

An aerothermo real-time model of a turbojet engine has been developed for demonstration and educational purposes. It has been combined with a simple control so that it might be exercised. The FORTRAN code of the engine simulation and sample input and output are provided in Appendix A.

4.4 Piece-Wise Linear Models

Piece-wise linear models, also referred to as state-space models, represent the majority of the engine's aerothermodynamic processes with a set of matrix equations that describe the variations about a set of base characteristics. The engine processes are characterized as a whole rather than modeling individual components such as compressors, combustors and turbines. This type of model is generally derived from a nonlinear aerothermodynamic model, although tuning of the model with engine data may also occur for certain applications. When engine control functions are included as part of an engine system model, these are simulated outside the aerothermodynamic piece-wise linear engine model.

4.4.1 Mathematical Model Approach

4.4.1.1 Theory

The piece-wise linear modeling techniques extend classical linearized dynamic theory, which employs partial derivatives about a single operating point, by including multiple model points. Interpolation between these model points is used to define a base point which moves as engine power changes. Adjustments for flight condition and augmentation provide accurate steady-state and transient characteristics of an engine operating between minimum and maximum power settings. The engine is modeled using three types of variables: states, inputs, and outputs. States represent energy storage parameters such as speed and metal temperature. Inputs are variables which perturb the systems, such as fuel flow and jet area. Outputs are additional parameters of interest other than the state and input variables, such as airflow and thrust. Dynamically, the engine is characterized by differential equations relating the time rate of change of the state variables to the state variables themselves and the input parameters. States are obtained transiently by calculating the derivatives and numerically integrating forward in time.

4.4.1.2 Form of Equations (Engine Model)

The linearized time rate of change of a state is formed as the sum of partial derivatives with respect to pertinent parameters using the Chain Rule. Using low pressure spool rotor speed (XNL) as an example (see Equation 5):

$$\frac{dXNL}{dt} = \frac{\partial \dot{XNL}}{\partial XNL}(XNL - XNL_b) + \frac{\partial \dot{XNL}}{\partial XNH}(XNH - XNH_b) + \frac{\partial \dot{XNL}}{\partial WFE}(WFE - WFE_b) + \ldots \quad \text{(Eq. 5)}$$

where:

XNL = low pressure spool rotor speed

XNH = high pressure spool rotor speed

WFE = main burner fuel flow

b = denotes a base operating point

For convenience, the equation is broken into two parts, one dealing with the partials with respect to states, the other dealing with partials with respect to inputs. The full set of differential equations can then be written in matrix notation as in Equation 6:

$$\dot{X} = A(\Delta X) + B(\Delta U) \quad \text{(Eq. 6)}$$

where:

$\dot{X}$ = vector of state derivatives

A = matrix of partial derivatives relating state derivatives to states

B = matrix of partial derivatives relating state derivatives to inputs

ΔX = vector of differences between the actual and base point values of the states. The base points are values of the states about which the partial derivatives are defined.

ΔU = difference between the actual and base point values of the input parameters

Likewise, outputs are calculated as functions of the states and input parameters. In matrix notation (see Equation 7):

$$\Delta Y = C(\Delta X) + D(\Delta U) \quad \text{(Eq. 7)}$$

where:

ΔY = vector of differences between the actual and base point values of the outputs. The base points are values of the outputs about which the partial derivatives were defined.

C = matrix of partial derivatives relating outputs to states

D = matrix of partial derivatives relating outputs to inputs

ΔX and ΔU are as defined in Equation 6.

The partial derivatives (A, B, C, and D matrices) that are used in the piece-wise linear model are usually derived from an aerodynamic/thermodynamic simulation of the engine, consisting of detailed component representations which characterize transient performance.

4.4.1.3 Form of Equations (Control Model)

While the engine model represents aerothermodynamic processes and can be linearized about a point, electronic digital controls cannot be treated in the same manner. The control is itself a computer program and is not subject to physical laws describable by differential equations. In general, when a complete propulsion system model is required (i.e., engine with control), the control model will be a separate entity.

4.4.2 Usage

There are basically two types of piece- wise linear state-space models: Predictor models and Observer models. Each has unique applications and can be characterized in terms of model inputs and model outputs. Predictor models are the most common piece-wise linear models. They take as inputs, in addition to engine boundary conditions (i.e., engine face pressure, temperature, and ambient conditions), a set of control effectors and output a set of engine parameters. Predictor models may have extended inputs. Extended inputs are generally "tuning" parameters such as efficiency and flow shifts.

Predictor models can stand alone or be combined with an engine control system to form an overall propulsion system model. Stand alone predictor models are used for control design, control benches and as embedded models for flight systems. In addition, since piece-wise linear representation of the engine's aerothermodynamic process is the basis of these models, they are easily adapted to control design and development where a piece-wise linear model of the controlled process is required. When combined with a control, the overall propulsion system models are used for flight control design, flight system evaluation, and flight simulators.

Observer models are generally used as embedded models. They are derived from the corresponding predictor model (with extended inputs) by rearranging the calculations, and often incorporating Kalman filtering techniques, to accept as inputs measurable engine parameters and output the extended outputs as well as other engine parameters. They can be used as part of performance optimization, synthesizing unobservable engine parameters, and detecting failures.

Piece-wise linear models offer several advantages over other real-time engine models. The primary advantage is that with the state-space matrices, the model can be used for a wide range of applications and there are many numerical techniques available for analysis and design. Some of the applications have already been mentioned. In addition, the models are readily generated from nonlinear aerothermodynamic simulations at relatively low cost and can be scaled to represent specific engines by modifying the base point data.

The major limitation of piece-wise linear models is that they function well only for normal engine operation and are not adaptable to extremely nonlinear processes. Generally, separate calculations are incorporated to represent severe abnormal engine behavior such as engine stall or nonnormal operating regions such as engine starting.

4.4.3 Characteristics

4.4.3.1 Size

Typically the engine model code size is on the order of 1 MB without data. The amount of engine data is highly dependent on accuracy requirements and the intended applications. A typical model would use 12 to 14 base points and would require approximately 10 MB of storage. However, if special requirements are necessary both the model size and data size will increase. If a combined propulsion system model is required, the size and complexity of the control simulation must be considered.

Normally the control model size is considerably larger than the engine model and may require up to 5 times the execution time of the engine model. It is possible to reduce the control model complexity to meet size and run time constraints at the loss of control model fidelity.

4.4.3.2 Consistency

Generally, the piece-wise linear models will generate steady-state gross thrust and airflow which are within 6% of the reference aerothermodynamic model from which the piece-wise linear model is derived. When higher fidelity is required, additional base points can be used to attain almost any accuracy but at the expense of added storage and computational time.

4.4.3.3 Tradeoff Between Accuracy, Speed, and Cost

Execution time varies widely depending on the accuracy requirements, programming language, compiler efficiency, target processor, and number of inputs/outputs. Since the model is basically a set of matrices, the execution time will vary with the number of states, inputs and outputs according to the number of multiplications to evaluate the derivatives and outputs. Thus, execution time is roughly proportional to Equation 8:

$$(\text{Execution Time}) = (\text{Number of States} + \text{outputs}) \cdot (\text{Number of states} + \text{inputs}) \quad \text{(Eq. 8)}$$

As an example, a model with three states, six inputs, and 20 outputs will require approximately twice the execution time of a model three states, five inputs, and 10 outputs. A secondary effect on computer time is the number of base points.

4.4.3.4 Versatility

The piece-wise linear models can be scaled to represent specific test engines simply by modifying the base point data. Scaling to different thrust/airflow sizes is possible provided the basic engine design (i.e., bypass ratio, fan pressure ratio, compressor pressure ratio) is constant. This involves modifying the base point data and analytically modifying the state derivative partials to account for polar moment of inertia and thermal time constants.

4.4.3.5 Modes of Operation

Generally the piece-wise linear models allow time-varying (transient) operation or steady-state initialization at any flight condition for engine operation between idle and maximum augmentation. When combined with an engine control, the steady-state initialization mode initializes the engine model and control to specified power lever angle (PLA). In special cases, an additional iteration loop on PLA to initialize at a specified thrust can be incorporated.

The piece-wise linear state-space models generally do not simulate severe abnormal engine operation such as stalls, augmentor blow-outs, or engine flame outs. They can be used to simulate certain control failures and/or sensor failures so long as the failures do not result in rapid, widely varying model inputs. If necessary, special nonlinear computations can be added to account for abnormal operation, but at the expense of model complexity, size, and execution time.

If special modes, such as start and/or shutdown are required, these calculations are generally incorporated as nonlinear representations which add to the model complexity, size, and execution time.

4.4.3.6 Dynamic Bandwidth

Dynamic bandwidth of the engine model is generally up to 5 Hz (i.e., 3 dB deviation from the reference aerothermodynamic model) for rotor speeds and engine pressures to a sinusoidal fuel flow. For a complete propulsion model, dynamic bandwidth comparisons are not made since the engine bandwidth is greater than the control bandwidth.

4.4.3.7 Input/Output

In general a predictor-type engine model will require flight condition information (P2, T2, PAMB). Depending on the application, other inputs would normally include fuel flow, jet area, variable geometry, and in the case of extended inputs, efficiency and flow shifts. If the engine model is coupled with a control model, then PLA would normally be input and the control model would supply inputs to the engine model.

Normal outputs include engine rotor speeds, pressures, temperatures, and flow rates. It is relatively simple to reduce the number of outputs to increase execution speed, but it is difficult to increase the number of outputs as this requires generation of additional partial derivatives.

4.4.3.8 Data Origin

Data necessary for a piece-wise linear state-space model consists of base point values and the state-space matrices. The base points and matrices are usually defined at multiple engine power conditions. The data is normally obtained from a detailed aerothermodynamic simulation. However, base point information can be obtained directly from engine test data.

4.4.3.9 Time Step Margins

Generally, the time-domain response of the piece-wise linear models are relatively insensitive to time step for normal throttle transients. However, when combined with an engine control model, actual closed-loop stability margin is affected by increasing the time step. Generally the engine alone can operate with time steps of 100 ms, and a combined engine/control model can operate at 50 ms with some loss in transient fidelity.

4.4.4 Example Case

Included in Appendix B is a sample model using the piece-wise linear approach. The simplified aerothermo model (see 4.3) was linearized at multiple points to produce the piece-wise linear model. Note that the volume dynamics are not part of the linear model but are the same as in the simplified aerothermo simulation. In general, high frequency dynamics or extremely nonlinear representations are handled as separate calculations instead of being part of the state-space model.

4.5 Transfer Function Models

Like piece-wise linear state-space models, the engine processes are not represented on a component level, but rather, are represented in terms of overall engine response. They are derived from nonlinear aerothermo models and are normally driven by engine control functions simulated separately from the engine model.

In contrast, Transfer Function models represent the engine's aerothermo processes at steady-state with a set of base characteristics, and transiently with simple “leads” and “lags” applied to the base characteristics as they respond to control inputs. This architecture allows a reasonable amount of tunability to the steady-state performance levels and shapes, but is somewhat difficult to tune to specific transient events.

4.5.1 Mathematical Model Approach

4.5.1.1 Theory

The gas generator model is segregated into three parts, the steady-state baselines, the subsystem offsets, and the transient characteristics.

The steady-state baselines are represented by nonlinear functions which are the steady-state operating lines of the engine. These are usually curves of corrected output parameters versus corrected fuel flow and flight Mach number. These curves would typically be generated using a detailed aerothermo model along a nominal flight path with nominal or fixed subsystem inputs. Additionally, the baselines can be derived from, or scaled, based on developmental or flight test results.

Subsystems (i.e., surge bleed, customer bleed, variable stators, etc.) are modeled via the application of offsets to the steady-state baselines. The offsets are functions of appropriate corrected input or output parameters (e.g., fuel flow or rotor speed) and flight Mach number. The offsets are obtained by observing the effect of each subsystem on the output parameters. The amount of offset used for a subsystem is dictated by the deviation of the control model command from the nominal or fixed subsystem input assumed in the steady-state baselines.

The transient characteristics of the engine are represented by transfer functions. A transfer function filters the change to an input variable and transfers it to an output variable to create its response. The transfer function consists of a ratio of two polynomials in the Laplace operator “s” which can be expressed as a ratio of factored roots. The denominator of a transfer function is called the "characteristic function" of the physical system and determines all the possible kinds of natural motion the system can have. The characteristic function is often referred to as the "lag" of the system. The numerator of the transfer function determines which part of the characteristic function will be dominant in the engine parameter response. The transfer function is implemented in the form of lead/lag pairs. The roots of the polynomials determine the lead and lag time constants. While a unique set of time constants can be generated for each engine parameter, in practice parameters will often share the same coefficients if their dynamic response characteristics are similar.

The coefficients (time constants) of the transfer functions vary as a function of a match parameter (e.g., corrected rotor speed). The coefficients are obtained from the partial derivative matrices of a state space model which are algebraically manipulated to generate an equivalent transfer function representation. The equivalent transfer function polynomials are determined by finding the set of coefficients which best approximate the frequency response of a parameter. In the state space representation the relationship between the states, inputs, and outputs is explicitly represented by the partial derivatives. In the transfer function representation, however, a parameter’s response to all state and input perturbations is lumped into a lead/lag transfer function with a single input (usually fuel flow). The steady-state value of a parameter is scheduled as a function of fuel flow then passed through a set of lead/lag functions to determine the parameter’s transient response.

4.5.1.2 Form of Equations (Steady-State)

Representation of corrected output parameters as a function of fuel flow and flight Mach number provide definition of steady-state baselines.

4.5.1.3 Form of Equations (Subsystem Operation Effects)

Representation of output parameter offsets as a function of fuel flow and flight Mach number provide definition of subsystem operation effects.

4.5.1.4 Form of Equations (Transient)

The transient characteristics are represented by applying lead/lag transfer functions to the offset-adjusted steady-state baselines (see Equation 9).

Transfer Function:

$$\frac{Y}{X} = \frac{\tau_{lead} S + 1}{\tau_{lag} S + 1} \tag{Eq. 9}$$

Numerical Algorithm:

$$Y_t = X_t\left(\frac{\tau_{lead}}{\tau_{lag}}\right) + \left(1 - \frac{\tau_{lead}}{\tau_{lag}}\right)\left[Y_{t-\Delta t} + (X_t - Y_{t-\Delta t})\left(1 - e^{\left(\frac{\Delta t}{\tau_{lag}}\right)}\right)\right]$$

where:

τ_{lead} = lead time constant

τ_{lag} = lag time constant

Δt = time increment

Y = output

X = input

s = laplace operator

Since the denominator of the transfer function represents the characteristic function of the system, it is often the practice to represent the individual contributions of a systems' state variables individually in the denominator. Each first order lag would then be accompanied by its associated lead. Equation 9 (Transfer Function) would then be expanded to (see Equation 10):

$$\frac{Y}{X} = \frac{(\tau_{lead,XNL} s + 1)(\tau_{lead,XNH} s + 1)(\tau_{lead,TM} s + 1)}{(\tau_{lag,XNL} s + 1)(\tau_{lag,XNH} s + 1)(\tau_{lag,TM} s + 1)} \tag{Eq. 10}$$

For a system with two rotor speeds and a metal temperature as state variables. Since certain lead/lag pairs will cancel, this general case can often be simplified.

4.5.2 Usage

Transfer function models have applications similar to those described for piece-wise linear models (see 4.4), in that they are used to predict real time performance of engines when accompanied by a control or control simulation. They have some advantages to other types of models, particularly in the ease of updating the model performance. Since the base characteristics are simply steady-state operating lines, they can be easily updated to match performance levels for a specific engine (for example, demonstrated flight test performance versus pre-test prediction). Additionally, since the subsystem characteristics are in the form of deltas rather than partial derivatives, they are easier to calibrate to data. Also, transfer function models are easy to initialize to any flight condition and input condition because the steady-state performance is explicitly defined.

Relative to piece-wise linear models, transfer function models tend to be smaller in size and in the number of calculations required. As such, they can be implemented on smaller and more economical computers. However, this type of design makes the models difficult to tune dynamically, since the dynamic response of an engine parameter to all input and state variables is lumped into its time constants, rather than broken out into single-input to single-output relationships as in partial derivatives. This tends to force reliance on tuning the reference aerothermo model to the desired events and regenerating the coefficients. For many applications, this would not be a problem, but should be considered before choosing this type of model.

4.5.3 Characteristics

4.5.3.1 Size

Transfer function models are perceived to be smaller in size, in terms of the amount of data and number of calculations required, compared to piece-wise linear models. In fact, a rigorously defined transfer function model will be similar in size. For example, a piece-wise linear model that has four states, four inputs, and six additional outputs would require 14 steady-state baselines, one each for the states, inputs and outputs, 16 “A” matrices, 16 “B” matrices, 24 “C” matrices and 24 “D” matrices. This amounts to a total of 94 data elements to represent the engine’s response. A comparable transfer function model would also require 14 steady-state baselines. In addition, the model would require 40 subsystem offsets (10 states and outputs times four inputs), four lag time constants, and 40 lead time constants for a total of 98 data elements. In this example the two models would be similar in size.

In practice, however, the transfer function model is simplified because the methods used to generate the model are less automated and more time consuming than for a state space model. For example, the piece-wise linear model tends to rely on automated calculation of the partial derivative coefficients and therefore each coefficient is included in the model no matter how insignificant. The transfer function model relies on nonautomated generation of subsystem offsets. The experienced modeler will not generate the least significant deltas to save time. In addition, even though the leads and lags are automatically generated, many lead/lag combinations are judged to cancel each other and are therefore eliminated resulting in fewer data elements and calculations. In general, transfer function models will be about 25 to 33% smaller than piece-wise linear models.

4.5.3.2 Consistency

Transfer function models, like piece-wise linear models, are consistent with their reference data base within 5 to 10%. Higher fidelity is achievable at the expense of additional storage and computational time, which makes coordination with the customer of the product essential prior to construction of these models.

4.5.3.3 Tradeoff Between Accuracy, Speed, and Cost

As with other real-time models, transfer function models can achieve high accuracy at the expense of speed and cost. To achieve comparable accuracy with a piece-wise linear model, the transfer function model would tend to have the same speed with higher cost. The speed would be comparable due to similar amounts of data and code. The higher cost would be associated with the generation of the subsystem offsets and with the tuning of the time constants to match dynamic response. In practice, transfer function models tend to be less accurate, faster, and similar in cost to piece-wise linear models. If the application does not require the highest fidelity and computer hardware constraints limit the storage and execution time of the model, a simplified transfer function model can be generated. In some applications the baselines are generated with scheduled subsystem effects so the offsets can be eliminated. This allows the model to be generated at lower cost and to execute at a faster rate.

4.5.3.4 Versatility

Transfer function models are scalable to new performance levels by modifying the baseline steady-state curves to reflect the updated performance. If changes to the engine include changes to rotor inertia and heat transfer effects, analytical modifications of the time constants is very difficult. For these types of changes, the time constants are typically regenerated using a re-calibrated aerothermo dynamic simulation. For this reason, coordination with the customer regarding intended use and fidelity requirements is extremely important.

4.5.3.5 Modes of Operation

The transfer function model can operate as a transient model or a steady-state model, with or without a closed loop control simulation. Many special operational modes and failure modes can be modeled using the basic transfer function methodology (including startup, shutdown, etc.), although many times nonlinear methods (i.e., torque integration for starting) are used for expediency. The addition of special modes of operation, particularly those requiring nonlinear methods, increase the complexity, size, and execution time of the model.

The transfer function models generally do not simulate severe abnormal engine operation such as stalls, augmentor blow-outs, or engine flame outs. The transfer function models can be used to simulate certain control failures and/or sensor failures so long as the failures do not result in rapid, widely varying model inputs. If necessary, special nonlinear computations can be added to account for abnormal operation, but at the expense of model complexity, size, and execution time.

4.5.3.6 Dynamic Bandwidth

Dynamic bandwidth of the transfer function model is generally up to 5 Hz (i.e., 3 dB deviation from the reference aerothermo model) for rotor speeds and pressures in response to a sinusoidal fuel flow input. For a complete propulsion system model, dynamic bandwidth comparisons are not made since the engine bandwidth is greater than control bandwidth.

4.5.3.7 Input/Output

Inputs required for the transfer function model include environmental conditions (e.g., P2, T2, PAMB), engine control variables (e.g., fuel flow, exhaust nozzle area, stator position), and external power requirements (e.g., compressor bleed air, power extraction, starter power input).

The transfer function model has the same output capability and limitations as a Piece-Wise Linear model. As in the Piece-Wise Linear model, adding additional outputs to the transfer function model requires generation of additional dynamic elements. However, in actual practice, it is often possible to represent an additional output parameter with existing lead/lag pairs, provided that the additional output's response closely resembles the existing parameter's response.

4.5.3.8 Data Origin

Data necessary for a transfer function model consists of base point values, deltas to the base point values, and the lead and lag time constants. The data is usually defined at multiple engine power conditions and is normally obtained from a detailed aerothermodynamic simulation. The base points and deltas can also be readily obtained directly from engine test data and often these values are updated while the time constants remain unchanged.

4.5.4 Example Case

A real time model of a turbojet engine has been developed for demonstration and educational purposes. It has been combined with a simple control so that it might be exercised. The FORTRAN code of the engine simulation and sample input and output are included in Appendix C.

4.6 Selecting a Modeling Methodology for a Specific Application

This section describes some general guidelines for matching application requirements to model characteristics. First, performance measurement criteria are described. Then, specific application requirements and model performance characteristics are reviewed. Finally, the application requirements are combined with the performance characteristics to describe industry trends in using models.

Choosing the correct modeling methodology for a specific application is really an optimizing process trading off modeling performance characteristics against the application requirements. This trade must take place - one cannot have the highest fidelity representation and at the same time have the fastest and most compact model. Simplifications that result in a reduction in execution time normally degrade the fidelity of the model. Flight simulators are classic examples of the trade between speed and accuracy - they typically use either transfer function or piece-wise linear models to meet execution rate and versatility requirements. By placing a higher priority on update rate, Flight Simulators accept reduced accuracy and consistency; this is considered an acceptable trade. The trades that are made when choosing the best model usually come down to a balance between model fidelity and update rate requirements; models with higher levels of fidelity, in general, require more computer resources to run real-time.

4.6.1 Performance Measurements Used to Select Modeling Methodology

The following four performance measurements can be used to help select the best modeling methodology for a specific application:

a. Consistency: Refers to the simulation's ability to match data

b. Versatility: Refers to the ease of adjusting the simulation to match data, and its ability to simulate abnormal events such as shaft failures or locked rotors

c. Bandwidth: Defines the range of frequencies for which the simulation is accurate

d. Execution Rate (speed): Dictates how much computer resources are required for the simulation

4.6.2 Real-Time Model Application Requirements

To choose the correct modeling method, the application requirements must be understood. Table 1 shows the general trend in requirements for specific applications. Sometimes it is difficult to characterize a requirement (e.g., versatility for embedded control applications). Some embedded controls require the simulation to be continuously updated to match data (e.g., sensor models), whereas other embedded control applications require little, if any, versatility.

Table 1 - Real-time model performance requirements

Application	Consistency	Versatility	Bandwidth	Speed
Control Bench	medium	medium	medium	low
Embedded Flight System	medium	low	low	medium
Flight Simulator	medium	high	low	high
Flight Propulsion Control	high	medium	high	high
Embedded Control	high-medium	high-low	medium	high

The application requirements are described as high, medium, and low. High requirements should be thought of as stringent. For example, a high bandwidth requirement means that the model must accurately simulate the frequency response of the engine and a medium rating means that less emphasis is placed on frequency response than other requirements. An example of this is flight simulators that have accuracy as a medium requirement, thus they typically use models that give up some accuracy to achieve fast execution times. A low requirement rating means that little emphasis is placed on this requirement - shortfalls in simulation performance with a low requirement rating are normally acceptable or have been traded to get better performance in another category.

4.6.3 Modeling Methodology Performance Characteristics

The following is an overview of the performance characteristics of each model type.

The model performance characteristics are rated as high, medium, and low and are listed in Table 2. These ratings reflect how well the model meets requirements in a particular area. For example, aerothermo models are relatively high in consistency so they receive a high rating in the consistency category. Exceptions to the ratings listed in Table 2 may exist; for example, in general, a piece-wise linear model executes faster than an aerothermo model, however, a simplified aerothermo model may execute faster than a complex piece-wise linear model.

Table 2 - Modeling methodology performance characteristics

Model Type	Consistency	Versatility	Bandwidth	Speed
Aerothermo	high	low	high	low
Aerothermo – Reduced Balance Structure	high	high	high	low/medium
Transfer Function	low	low	low	medium
Piece-Wise Linear	medium	medium	medium	medium

4.6.4 Selecting Specific Modeling Methodology for a Specific Application

Selecting between the three types of models for a specific application requires a clear understanding of the intended use of the model. Several issues should be considered:

a. Is the engine in an early design phase?

b. What is the degree of modeling complexity?

c. Will the model need to be calibrated to flight test results?

d. Will the model be used in an embedded control or flight propulsion control?

e. Will the model be used in an embedded flight system?

f. Will the model be used for flight simulation?

g. Will the model be used for control bench work?

h. How many models will be produced in a year?

i. What computer equipment will host the model?

j. Is there an automated system available to produce dynamics?

k. Will the model be used for more than one purpose?

It is essential to have discussions between the supplier and the customer to establish the best selection for each application. In general, however, applications with high modeling complexity, frequent changes in engine design and a need to compute many thermodynamic variables are best served by aerothermo models. Less complicated systems or systems that need to be calibrated to flight test results (including self-calibrating systems) are best served by piece-wise linear models. When there is no automated system available to generate system dynamics, transfer function models can be created manually very quickly and accurately, using minimal computer resources

5. INTEGRATION OF REAL-TIME ENGINE MODELS WITH OTHER SYSTEMS

A wide range of real-time engine model usage and capabilities is evident within this document. Effective communication between customer and model supplier is essential in producing and maintaining a real-time engine model that satisfies the specific requirements for each application.

Preliminary coordination is essential between customer and supplier to establish a mutual understanding of the model usage and integration requirements. In addition, the initial coordination also serves to establish common expectations for emerging integration requirements and model limitations. Follow-up coordination is essential to identify a match between the model application and the modeling methodology and to further refine requirements as they relate to model consistency, bandwidth, speed, versatility, and computing platform. A continuing and constant exchange between customer and supplier ensures that changing and emerging requirements are addressed.

Among the integration requirements for the real-time engine model is the ability to interface with other real or simulated sub-systems as part of an overall system. The interfaces must be defined and the functionality of the engine model as it relates to the other sub-systems must be identified. Typical examples include interfaces with the airframe (applied forces and moments), with electrical and mechanical loads served by the engine, with sensor simulations, and with control systems. The interfaces should be identified during preliminary coordination between customer and supplier.

Requirements for special engine model capabilities must also be coordinated during preliminary coordination between customer and supplier. Special capabilities include engine abnormal operation and failure mode simulation.

Effective communication and coordination between customer and supplier is essential in satisfactorily matching model capabilities and limitations with firm and emerging model requirements.

6. UTILITIES

6.1 Iteration Methods (Iterative Solvers)

Iterations in real-time models should be avoided whenever possible. The reason is that iterations require more execution time and a poorly designed iteration can fail to converge or even make the model unstable. However, there are cases where a well-designed iteration is acceptable.

If iterations cannot be avoided entirely, then it is best to attempt to design the program flow such that only one parameter requires closure. This is a single variable iteration. When two or more parameters are interrelated and require closure a multivariable iteration is necessary. Single variable iteration schemes can be designed which are robust and suitable for real-time applications. The classes are: bisection, successive substitutions, and Newton-type methods. Multivariable iterations are typically Newton-type methods.

6.1.1 Bisection

If the problem can be formulated such that lower and upper bounds can be established for the unknown parameter then a bisection iteration can be applied. The functional form is shown in Equation 11:

$$X = \frac{(X_{lower} + X_{upper})}{2} \qquad \text{(Eq. 11)}$$

where a new estimate of the unknown is calculated as the mid point between the upper and lower bounds. The function is evaluated with the new estimate and the bounds reset depending on the sign of the function.

Bisection iteration has advantages over other methods. One is, assuming that an initial bracket can be established, the method will always find a solution. Another desirable property is that since the range will be divided in half at each pass, the number of passes to achieve the required accuracy can be calculated in advance. Thus, an iteration with a fixed number of passes that cannot fail can be designed. This is the recommended method for real-time applications.

6.1.2 Successive Substitutions

If the problem can be formulated such that the unknown parameter can be calculated as a function of itself then successive substitutions may be applied. The functional form is shown in Equation 12:

$$X = F(X) \qquad \text{(Eq. 12)}$$

where a new estimate of the unknown is calculated from the previous estimate. There are two problems associated with successive substitutions which the program developer must consider. First, for convergence, the magnitude of the derivative of F(X) with respect to X must be less than unity. Functionally (see Equation 13):

$$\left|\frac{dF(X)}{dX}\right| < 1 \qquad \text{(Eq. 13)}$$

If this criteria is violated, a relaxation constant can be applied. A second problem is that the rate of convergence is not generally known and the number of passes required must be established by trial and error. However, there are many examples of successive substitution which provide fast, reliable convergence.

6.1.3 Newton Methods

Newton methods involve calculating the function derivative (or estimating the derivative from function evaluations). They have the advantage of rapid convergence when performing correctly. However, there is no guarantee of convergence. Newton methods are not normally recommended for real-time applications and should be considered as a special circumstance method where the rate of convergence and failure rate has been established empirically.

6.1.4 Multivariable Iterations

Multivariable iteration schemes are generally a multidimensional extension of the Newton methods described. Many methods have been devised to reduce failure rates including the use of pre-determined or analytically-derived partial derivatives.

The single-pass cycle balance avoids the iterative nature of typical Newton methods. The single-pass cycle balance is a procedure which makes a single attempt to improve the cycle guess parameters (e.g., bypass ratio, component operating lines) as a function of the continuity errors after each time step. Since tolerances cannot be imposed, it should be demonstrated that the single-pass procedure does not produce unacceptable differences from a multipass procedure (with error tolerances) prior to selecting the single-pass procedure. Computations of errors and guesses should occur between the actuator update and the cycle calculation so that flow-area changes which affect errors will affect the cycle in the current time-step as they would in a multipass procedure.

A variation of the single-pass cycle balance is a fixed-number-pass cycle balance. The number of iteration passes per time step is pre-selected to ensure a fixed execution time. Typically 2 to 4 passes are used, depending on the processor speed and engine response rate. This approach is often called “Truncated Newton.” The truncation does not normally cause any loss in the basic numerical stability of the iterative method. Obviously this approach requires higher processor speed for a given execution time, but it has the advantage of requiring less work to create the real-time model from the normal performance cycle program and it usually confers greater accuracy than the single-pass cycle balance. With the limited number of passes per time step there is no opportunity to recalculate the Jacobian matrix of partial derivatives. However, the calculation of a few passes at each time step does provide data, which can be used to adjust the elements of the Jacobian matrix. This is done using the Broyden Secant formula. When there are rapid changes in the model, such as handling bleed valve movement, it may be necessary to apply a damping factor to the changes calculated by the Broyden formula. The Broyden formula is:

$$J = J + \frac{b(y - s \cdot J)s^T}{s^T s} \quad \text{(Eq. 14)}$$

where:

J = Jacobian matrix

b = damping factor

y = vector of changes in the error function between consecutive passes through the iteration loop

s = vector of changes in the variables between the same consecutive passes

6.2 Integration Methods

6.2.1 Explicit Integration

Explicit numerical integration gives the value of the parameter θ at the next time step (t+Δt) based on the derivative of θ at time = t and the size of the time step (see Equation 15):

$$\Theta_{t+\Delta t} = \Theta_t + \left(\frac{d\Theta}{dt}\right)_t \cdot \Delta t \quad \text{(Eq. 15)}$$

Equation 15 gives the simple Euler integration method. Various more complex integration schemes (e.g., Runge-Kutta) may also be used.

6.2.2 Implicit Integration

Implicit numerical integration gives the value of the parameter θ at the next time step (t+Δt) based on the derivative of θ at time = t+Δt and the size of the time step (see Equation 16):

$$\Theta_{t+\Delta t} = \Theta_t + \left(\frac{d\Theta}{dt}\right)_{t+\Delta t} \cdot \Delta t \qquad \text{(Eq. 16)}$$

$\theta_{t+\Delta t}$

6.2.3 Trapezoidal Integration

Trapezoidal integration is an implicit numerical integration which gives the value of the parameter θ at the next time step ($t+\Delta t$) based on the average of the derivative at both time step "t" and "$t+\Delta t$" along with the size of the time step (see Equation 17):

$$\Theta_{t+\Delta t} = \Theta_t + \left[\frac{\left(\frac{d\Theta}{dt}\right)_t + \left(\frac{d\Theta}{dt}\right)_{t+\Delta t}}{2}\right] \cdot \Delta t \qquad \text{(Eq. 17)}$$

For most systems, it is necessary to guess a parameter at $t+\Delta t$ and iterate until Equation 17 is solved.

6.3 Lead

A first order lead transfer function in the time domain is represented by Equation 18:

$$Y_n = \frac{X_n - X_{n-1}\left(e^{\frac{-\Delta t}{\tau_{lead}}}\right) - Y_{n-1}\left[\frac{\tau_{lead}}{\Delta t}\left(1 - e^{\frac{-\Delta t}{\tau_{lead}}}\right) - e^{\frac{-\Delta t}{\tau_{lead}}}\right]}{1 - \frac{\tau_{lead}}{\Delta t}\left(1 - e^{\frac{-\Delta t}{\tau_{lead}}}\right)} \qquad \text{(Eq. 18)}$$

where:

τ_{lead} = lead time constant

X_n = input to the transfer function

X_{n-1} = past value of the input

Y_n = output

Y_{n-1} = past value of the output

Δt = time step

6.4 Lag

A first order lag transfer function in the time domain is represented by Equation 19:

$$Y_n = Y_{n-1} e^{\frac{-\Delta t}{\tau_{lag}}} + X_n \left[1 - \frac{\tau_{lag}}{\Delta t} \left(1 - e^{\frac{-\Delta t}{\tau_{lag}}} \right) \right] + X_{n-1} \left[\frac{\tau_{lag}}{\Delta t} \left(1 - e^{\frac{-\Delta t}{\tau_{lag}}} \right) - e^{\frac{-\Delta t}{\tau_{lag}}} \right] \quad \text{(Eq. 19)}$$

where:

τ_{lag} = lag time constant

X_n = input to the transfer function

X_{n-1} = past value of the input

Y_n = output

Y_{n-1} = past value of the output

Δt = time step

6.5 Curve Data

6.5.1 Table Lookup

Table lookups are the most commonly used and flexible means of entering data into real-time models. The type of table selected can have a significant effect on the speed and accuracy of the model.

The most commonly used table method is a simple sequential search. The table index is saved dynamically as the model is executed. The advantage to this type of table is the speed as the table pointer is usually near the correct position. This type of table is typically used for time dependent tables.

Binary searching is another table method often used. The advantage of binary search is that the time required is always the same for the table interrogation which is important attribute in real-time models.

Direct lookup tables are inherently the fastest method. Opposed to other techniques, direct lookup requires no trial and error searching, but an exact relationship between the input and table position is required.

Linear interpolation is normally used for all tables but higher degrees of interpolation can be used for more accuracy at a sacrifice in speed. Higher degrees of interpolation is not recommended unless table size is a major constraint. Normally more points should be added to the table for greater accuracy rather than using higher degrees of interpolation.

6.5.2 Polynomials

Polynomials can provide an accurate, high speed method of providing data to models. The problem with polynomials is their lack of flexibility. When the data needs to be modified, the polynomials have to be regenerated. Sometimes the polynomial requires high degrees and multiple sections to “fit” the data. Polynomials are recommended for univariate functions that are unlikely to change (e.g., gas properties), but should be avoided for model-dependent data.

6.5.3 Splines

Splines are a useful compromise between the flexibility of table lookup and the accuracy of polynomials. Splines are particularly useful for bivariate functions that require more accuracy than table lookups and where table size is a consideration.

7. NOTES

7.1 Revision Indicator

A change bar (l) located in the left margin is for the convenience of the user in locating areas where technical revisions, not editorial changes, have been made to the previous issue of this document. An (R) symbol to the left of the document title indicates a complete revision of the document, including technical revisions. Change bars and (R) are not used in original publications, nor in documents that contain editorial changes only.

PREPARED BY SAE COMMITTEE S-15, GAS TURBINE PERFORMANCE SIMULATION NOMENCLATURE AND INTERFACES

APPENDIX A - EXAMPLE OF AEROTHERMODYNAMIC REAL-TIME MODEL

FOREWORD

An example of an aerothermo real-time model is contained in this section. A FORTRAN listing of the model, and sample input and output from the model are included.

A.1 OPERATING INSTRUCTIONS AND GENERAL DESCRIPTION

A.1.1 Introduction

The GEXX program is an adaptation of a more detailed transient component level engine model currently in use within GE Aircraft Engines. The GEXX program is being released for evaluation and review to illustrate the form, speed and capability of real time cycle models. This paper gives a brief description of the engine model and provides instructions for running the GEXX program.

A.1.2 Model Description (See Figure A1)

The GEXX program is a transient computer model which simulates the core of a single spool nonafterburning turbojet engine. In addition to modeling the basic performance of this engine, the program also demonstrates the use of heat soak, bleed, volume dynamics, variable stators and variable area. The model contains both chargeable and nonchargeable cooling flows. The program does not model the engine inlet, nor does it perform thrust or stall margin calculations.

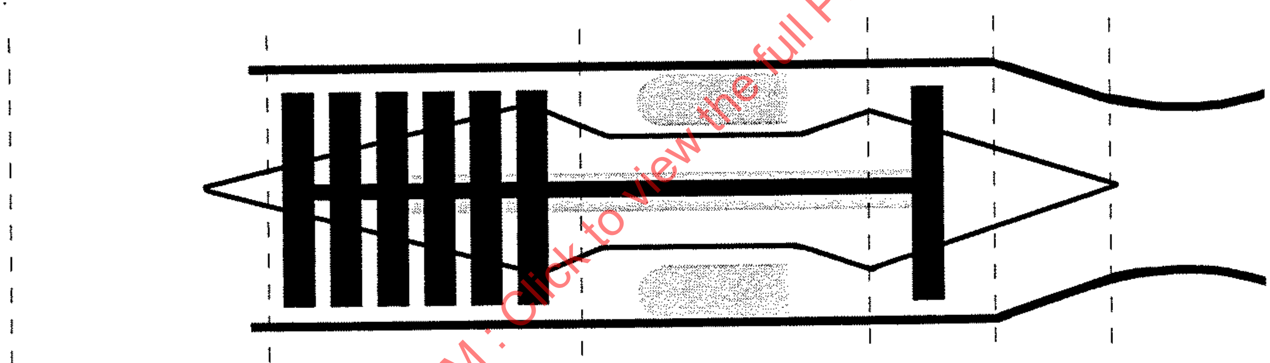

Figure A1 - Sketch of single-spool turbojet

In its present form, the program consists of a main driver program, a block data routine and approximately eight subroutines. The subroutines loosely correspond to the Figure A2. The main program calls the zero time balance, manages the time steps, invokes the model and prints program output. The CLM subroutine performs the engine simulation, calling the various component routines or performing the sections of applicable code. The CNTRL subroutine is a generic control model which provides fuel flow and nozzle area values.

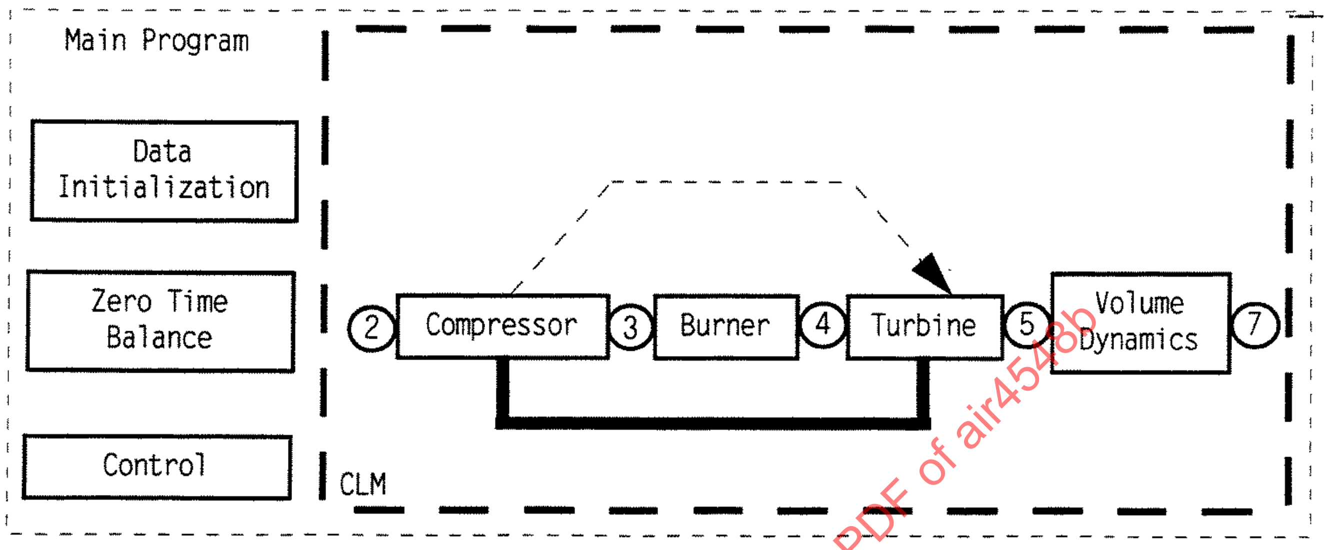

Figure A2 - Diagram of GEXX program

Both the compressor and turbine routines utilize maps to model component performance. Burner calculations are done within the body of the CLM routine. Volume dynamics for each time step are performed by a separate routine which is called near the end of the model.

A.1.3 Running the Model

The GEXX program is written in FORTRAN and utilizes the NAMELIST function for program input. The available program inputs are as follows.

TIME	Start Time of Study
FTIME	End Time of Study
PLA	Power Lever Angle
ZP2	Input Total Pressure at Compressor Inlet
ZT2	Input Total Temperature at Compressor Inlet
ZPAMB	Input Ambient Pressure
ZWB3	Input Customer Bleed

Power lever angle (PLA) can be varied from 70 to 100 in the model. PLA controls the core speed and acts as a user specified speed demand. Customer bleed (ZWB3) is input in percent of core flow and can vary from 0. to 0.1. The input values of ZP2, ZT2 and ZPAMB are also limited as follows.

ZP2	1.0 to 36.0 psia
ZT2	390 to 710 °R
ZPAMB	1.0 to 15.0 psia

The program uses a time step of 0.02 seconds. Program execution is terminated by entering an FTIME value less than the current time value (i.e., FTIME = -1.)

A.2 AEROTHERMO MODEL LISTING (SEE TABLE A1)

Table A1

```
PROGRAM MAIN
C########################################################################
C
C  DESCRIPTION
C    THIS IS THE MAIN PROGRAM FOR THE REAL TIME MODEL
C
C  SECURITY        UNCLASSIFIED
C
C  RESTRICTIONS    GE CLASS 1
C
C  COPYRIGHT (C)   1991 BY GE AIRCRAFT ENGINES
C
C########################################################################
C
      LOGICAL FADCIC
C
      COMMON /CLM1/GHHG,TIME,DT,PLA,ZPAMB,ZP2,ZT2,ZVSV2,ZWFM,
     &  ZA8,ZSUW2,ZSUL2,ZSUW41,ZSUL41,ZWB3,XNH,P5QG,XNHDOT,
     &  PNRM2,P3Q2,TMHPC,TMPC,TMHPT,YTFFH,YP8,HL41RG,FADCIC,
     &  W2R,HRM41,XNRM41
C
      COMMON /AIR/ W2,W3,W4,W5,P3,P4,P5,T3,T4,T5
C
C  ESTABLISH INPUT NAMELIST AND SET DEFAULT VALUES
C
      NAMELIST /INPUT/ TIME,FTIME,PLA,ZP2,ZT2,ZPAMB,ZWB3
C
      DATA FTIME / 1.0 /
C
    5 PRINT INPUT
      READ INPUT
C
C  CHECK FOR VALID INPUTS AND CORRECT AS NECESSARY
C
      IF(PLA.GT.100.) THEN
        WRITE(6,10) PLA
        PLA=100.
      END IF
C
      IF(PLA.LT.70.) THEN
        WRITE(6,20) PLA
        PLA=70.
      END IF
C
      IF(ZP2.GT.36.) THEN
        WRITE(6,30) ZP2
        ZP2=36.
      END IF
C
      IF(ZP2.LT.1.) THEN
        WRITE(6,40) ZP2
        ZP2=1.
      END IF
C
      IF(ZT2.GT.710.) THEN
        WRITE(6,50) ZT2
        ZT2=710.
      END IF
C
      IF(ZT2.LT.390.) THEN
        WRITE(6,60) ZT2
        ZT2=390.
      END IF
C
      IF(ZPAMB.GT.15.) THEN
        WRITE(6,70) ZPAMB
        ZPAMB=100.
      END IF
C
      IF(ZPAMB.LT.1.) THEN
        WRITE(6,80) ZPAMB
        ZPAMB=1.
      END IF
C
      IF(ZWB3.GT.0.1) THEN
```

```
          WRITE(6,90) ZWB3
          ZWB3=0.1
        END IF
C
        IF(ZWB3.LT.0.) THEN
          WRITE(6,100) ZWB3
          ZWB3=0.
        END IF
C
        IF(FTIME.LT.TIME) STOP
        IF(TIME.EQ.0.) CALL TIME0
C
         WRITE(6,190)
  190 FORMAT(6X,'TIME',3X,'PLA',4X,'PNRM',4X,'P3Q2',4X,'ZWFM',
     & 4X,'ZA8',4X,'ZVSV2',4X,'XNH',4X,'XNHDOT',5X,'P3',5X,'W2')
C
        WRITE(6,300)TIME,PLA,PNRM2,P3Q2,ZWFM,ZA8,ZVSV2,XNH,
     & XNHDOT,P3,W2
C
        ITER=0
C
  200 CALL CLM
C
        CALL CNTRL(PLA,DT,ZP2,ZT2,XNH,XNHDOT,P3,P3Q2D,
     &  ZWFM,ZA8,ZVSV2)
C
        IF(MOD(ITER,10).EQ.0) THEN
          WRITE(6,300)TIME,PLA,PNRM2,P3Q2,ZWFM,ZA8,ZVSV2,XNH,
     &  XNHDOT,P3,W2
        ENDIF
C
        ITER=ITER+1
        TIME=TIME+DT
        IF(TIME.LT.FTIME) GOTO 200
        GOTO 5
C
   10 FORMAT(1X,'NSI 0301 - INPUT PLA VALUE OF ',F7.2,' EXCEEDS MAX',
     &  ' ALLOWABLE VALUE OF 100.',/,12X,'PLA HAS BEEN SET TO 100.')
   20 FORMAT(1X,'NSI 0302 - INPUT PLA VALUE OF ',F7.2,' IS LESS THAN',
     & ' MIN ALLOWABLE VALUE OF 70.',/,12X,'PLA HAS BEEN SET TO 70.')
   30 FORMAT(1X,'NSI 0501 - INPUT ZP2 VALUE OF ',F7.2,' EXCEEDS MAX',
     & ' ALLOWABLE VALUE OF 36.',/,12X,'ZP2 HAS BEEN SET TO 36.')
   40 FORMAT(1X,'NSI 0502 - INPUT ZP2 VALUE OF ',F7.2,' IS LESS THAN',
     & ' MIN ALLOWABLE VALUE OF 1.',/,12X,'ZP2 HAS BEEN SET TO 1.')
   50 FORMAT(1X,'NSI 0503 - INPUT ZT2 VALUE OF ',F7.2,' EXCEEDS MAX',
     & ' ALLOWABLE VALUE OF 710.',/,12X,'ZT2 HAS BEEN SET TO 710.')
   60 FORMAT(1X,'NSI 0504 - INPUT ZT2 VALUE OF ',F7.2,' IS LESS THAN',
     & ' MIN ALLOWABLE VALUE OF 390.',/,12X,'ZT2 HAS BEEN SET TO 390.')
   70 FORMAT(1X,'NSI 0505 - INPUT ZPAMB VALUE OF ',F7.2,' EXCEEDS MAX',
     & ' ALLOWABLE VALUE OF 15.',/,12X,'ZPAMB HAS BEEN SET TO 15.')
   80 FORMAT(1X,'NSI 0506 - INPUT ZPAMB VALUE OF ',F7.2,' IS LESS THAN',
     & ' MIN ALLOWABLE VALUE OF 1.',/,12X,'ZPAMB HAS BEEN SET TO 1.')
   90 FORMAT(1X,'NSI 0401 - INPUT ZWB3 VALUE OF ',F7.2,' EXCEEDS MAX',
     & ' ALLOWABLE VALUE OF .10.',/,12X,'ZWB3 HAS BEEN SET TO .10.')
  100 FORMAT(1X,'NSI 0402 - INPUT ZWB3 VALUE OF ',F7.2,' IS LESS THAN',
     & ' MIN ALLOWABLE VALUE OF 0.',/,12X,'ZWB3 HAS BEEN SET TO 0.')
C
  300 FORMAT(3X,2F7.1,5F8.2,2F8.1,2F8.2)
C
C
        END
```

```
      BLOCKDATA BDCLM1
C#########################################################################
C
C  DESCRIPTION
C    ESTABLISH DATA FOR COMMON BLOCKS
C
C  SECURITY        UNCLASSIFIED
C
C  RESTRICTIONS    GE CLASS 1
C
C  COPYRIGHT (C)   1991 BY GE AIRCRAFT ENGINES
C
C#########################################################################
C
C
      LOGICAL FADCIC
C
      COMMON /CLM1/GHHG,TIME,DT,PLA,ZPAMB,ZP2,ZT2,ZVSV2,ZWFM,
     &  ZA8,ZSUW2,ZSUL2,ZSUW41,ZSUL41,ZWB3,XNH,P5QG,XNHDOT,
     &  PNRM2,P3Q2,TMHPC,TMPC,TMHPT,YTFFH,YP8,HL41RG,FADCIC,
     &  W2R,HRM41,XNRM41
C
      COMMON /AIR/W2,W3,W4,W5,P3,P4,P5,T3,T4,T5
C
      DATA PLA,ZP2,ZT2,ZPAMB,ZWB3 /100.,14.696,518.67,14.696,0./
      DATA TIME,DT,ZSUW2,ZSUL2,ZSUW41,ZSUL41/0.,0.020000,4*1./
C
      DATA GHHG,ZVSV2,ZWFM,ZA8       /4*0./
      DATA XNH,P5QG,XNHDOT           /3*0./
      DATA PNRM2,P3Q2,TMHPC,TMPC     /4*0./
      DATA TMHPT,YTFFH,YP8,HL41RG    /4*0./
      DATA FADCIC                   /.TRUE./
      DATA W2R,HRM41,XNRM41          /3*0./
C
      DATA W2,W3,W4,W5,P3,P4,P5,T3,T4,T5 / 10*0. /
C
C
      END
```

```
      SUBROUTINE CLM
C#######################################################################
C
C  DESCRIPTION
C    THIS ROUTINE CONTROLS THE EXECUTION ORDER OF THE INDIVIDUAL
C    ROUTINES
C
C  SECURITY       UNCLASSIFIED
C
C  RESTRICTIONS   GE CLASS 1
C
C  COPYRIGHT (C)  1991 BY GE AIRCRAFT ENGINES
C
C#######################################################################
C
      SAVE
      LOGICAL FADCIC
C
      COMMON /CLM1/GHHG,TIME,DT,PLA,ZPAMB,ZP2,ZT2,ZVSV2,ZWFM,
     &  ZA8,ZSUW2,ZSUL2,ZSUW41,ZSUL41,ZWB3,XNH,P5QG,XNHDOT,
     &  PNRM2,P3Q2,TMHPC,TMPC,TMHPT,YTFFH,YP8,HL41RG,FADCIC,
     &  W2R,HRM41,XNRM41
C
      COMMON /AIR/W2,W3,W4,W5,P3,P4,P5,T3,T4,T5
C
      DATA WRBH41,XJHP,CF8 / 18.578, 1.87, .963/
      DATA WCL41Q,WCL5Q / .095, .0523 /
C
      RT2=SQRT(ZT2)
      PNRM2=XNH*.195454/RT2
      H2=((ZT2*7.42102E-9-7.94082E-6)*ZT2+.241948)*ZT2-.451
C
      RNI2=ZP2*(ZT2+198.72)/(0.0392*ZT2*ZT2)
C
C  COMPRESSOR MAP
C
      CALL CMP2X(GHHG,PNRM2,ZVSV2,ZDVSVH,RNI2,ZSUW2,ZSUL2,P3Q2,
     &  W2RM,HRM2,SM2)
C
      W2R=.97002*W2RM
      H3D2=HRM2*ZT2
      H3=H2+H3D2
      W2=W2R*ZP2/RT2*1.5497
      PS2=ZP2-2.81E-7*W2*W2*ZT2/ZP2
      T3=((-H3*1.04605E-6-5.27271E-5)*H3+4.23409)*H3-3.01028
C
C  COMPRESSOR POWER CONSUMPTION
C
      W3=W2
      PW2=W3*H3D2
C
      P3=P3Q2*ZP2
      PS3=P3-6.5406E-4*W3*W3*T3/P3
C
C  CDP BLEED
C
      WC35P=WCL5Q*W2
      WC3040=WCL41Q*W2
      W36=W3-(WC35P+WC3040)-ZWB3
C
C  MAIN BURNER
C
      WFC=ZWFM*.000277778
      FAR4=WFC/W36
      T3D4=76202.*FAR4*(1.0-8.12*FAR4)
      P4=P3-5.4853E-4*W36*W36*T3/P3
      W4=W36+WFC
C
C  BURNER HEAT SOAK
C
      T4A=T3+T3D4
      TPC=0.5*(T3+T4A)
      DMPC=TPC-TMPC
      QPC=W36*DMPC*0.0146
      T4=T4A-DMPC*.04794
C
C  RECALCULATE ENTHALPY AFTER HEAT SOAK
```

```
C
      H4J=((-T4*1.42037E-9+1.91495E-5)*T4+.216687)*T4+6.342
      SHJFT=1.+.000285*(T4+2000.)*FAR4
      H4=SHJFT*H4J
C
      P5=P5QG*ZP2*W2RM
C
C  HIGH PRESSURE TURBINE
C
      P41=P4
      W41=W4+WC3040
      H41=(W4*H4+WC3040*H3)/W41
      FAR41=WFC/(W41-WFC)
      STFHJ=1.-.000795*(H41+900.)*FAR41
      H41J=STFHJ*H41
      T41=((H41J*3.33344E-7-1.057431E-3)*H41J+4.44213)*H41J
     &-14.62035
      RT41=SQRT(T41)
      TFF41=W41*RT41/P41
      P4Q5=P4/P5
C
C  TURBINE MAP
C
      CALL TRB41X(P4Q5,XNH,T41,HL41RG,FADCIC,WRB41,HRM41,
     &  HL41RP,XNRM41)
C
      YTFFH=(WRB41-TFF41)/WRBH41
      YL41R=HL41RP-HL41RG
      W5P=W41
      H5D41=HRM41*T41
      H5P=H41-H5D41
      W5=W5P+WC35P
      H5=(W5P*H5P+WC35P*H3)/W5
      FAR5=WFC/(W5-WFC)
      STFHJ=1.-.000795*(H5+900.)*FAR5
      H5J=STFHJ*H5
      T5=((H5J*3.33342E-7-1.057431E-3)*H5J+4.42213)*H5J
     &-14.62035
C
C  SHAFT POWER BALANCE
C
      PW41=W41*H5D41
      PWACCH=XNH*XNH*(1.0569E-7+XNH*5.5456E-12)
      PWACCH=0.3*PWACCH
      YSHPH=PW41-PW2-PWACCH
      XNHDOT=70960.*YSHPH/(XJHP*XNH)
C
C  CALL VOLUME DYNAMICS FOR STATION 7 PROPERTIES
C
      DO 100 K=1,10
        P7D5=-4.3438E-6*W5*W5*T5/P5
C
        CALL VD7(H5,P5,P7D5,W7,H7,P7,W5)
C
        FAR7=WFC/(W7-WFC)
        STFHJ=1.-.000795*(H7+900.)*FAR7
        H7J=STFHJ*H7
        T7=((H7J*3.33342E-7-1.057431E-3)*H7J+4.42213)*H7J-14.62035
C
C  NOZZLE
C
        P7QAM=P7/ZPAMB
        CF8=.02251*(P7QAM-1.)+.90
        A8=CF8*ZA8
        FP8C=.5342-.000006*T7
        FPQFPC=W7*SQRT(T7)/A8/P7/FP8C
        FPS=P7QAM*FPQFPC
        FPSH=1.895-1.625E-5*T7
        PTAREQ=FPS
        IF(FPS.LE.FPSH) PTAREQ=1.0-0.0227138*FPS+.26*FPS*FPS
        YP8=(P7QAM-PTAREQ)/PTAREQ
        W7=W7*(1.+.8*YP8)
C
  100 CONTINUE
C
C  ITERATION EQUATIONS
C
```

```
      GHHG=GHHG-1.9*YTFFH
      HL41RG=HL41RG+.8*YL41R
      P5QG=P5/(ZP2*W2RM)
C
C   SPEED UPDATED
C
      XNH=XNH+XNHDOT*DT
C
C   METAL TEMPERATURE UPDATED
C
      TMPC=TMPC+QPC*DT*0.069
C
C
      RETURN
      END
```

```
      SUBROUTINE CNTRL(PLA,DT,P2,T2,XNH,XNHDOT,P3,PQD,WFM,A8,VSV)
C#######################################################################
C
C  DESCRIPTION
C    GE16 HPC VSV SCHEDULE EFFECTS AND (NOMINAL) VSV SCHEDULE
C
C  SECURITY        UNCLASSIFIED
C
C  RESTRICTIONS    GE CLASS 1
C
C  COPYRIGHT (C)   1991 BY GE AIRCRAFT ENGINES
C
C#######################################################################
C
      SAVE
      DIMENSION TAB(3),APQD(-2:11)
C
      DATA CVS3,CVS4,CVS6/75.,100.,20./
      DATA TAB / 52.5, 2.5,-21.5/
      DATA APQD/0.,10.,11.,1.,1.105,1.211,1.316,1.422,1.527,1.916,
     &   2.53,3.44,4.74,6.22/
C
      PNRM=XNH*.195454/SQRT(T2)
C
C  USE MAP NOMINAL VSV ANGLE FOR CONTROL
C
      XNRU=AMAX1(CVS3,PNRM)
      XNRU=AMIN1(XNRU,CVS4)
      DEL=(XNRU-CVS3)/CVS6
      IX=DEL
      BOTT=TAB(IX+1)
      VSVNOM=BOTT+(DEL-IX)*(TAB(IX+2)-BOTT)
      VSV=VSVNOM
C
C  SPEED DEMAND
C
      PNRMD=PLA
      PNRME=MIN(.2,MAX(-.2,(PNRMD-PNRM)*.01))
      PNRME=PNRME-.01*PNRM*XNHDOT/XNH
      PNRME=MIN(.2,MAX(-.2,PNRME))
C
C  PRESSURE RATIO DEMAND
C
      PQ=P3/P2
      PQD=FCT(PNRM,APQD)
      PQE=(PQD-PQ)/6.22
C
C  FUEL FEED RATE
C
      EWFED=(PNRME+PQE)*11652.
      RMV=AMAX1(-0.75,AMIN1(0.75,.004*EWFED))
      DWFED=WFM*DT*RMV
      WFM=MIN(3500.,MAX(200.,WFM+DWFED))
C
C  A8 FEED RATE
C
      EA8FED=(PNRME+PQE)*11652.
      RA8=AMAX1(-.75,AMIN1(.75,.003*EA8FED))
      DA8=A8*DT*RA8
      A8=MIN(150.,MAX(60.,A8+DA8))
C
C
      RETURN
      END
```

```
      SUBROUTINE CMP2X(GH,PNRM,ZVSV,DVSV,RNI,ZSUW,ZSUL,PQB,
     &WRB,HRM,SMM)
C########################################################################
C
C  DESCRIPTION
C     PARAMETRIC COMPRESSOR.
C     DESIGN POINT CONDITIONS ARE:
C       PQBD=6.0,PNRMD=100.0,GHLLD=3.394,WQMD=37.0,UTRBD=1196.0,
C       WRBD=40.0,SMD=25.0%
C
C     TO MODIFY FOR OTHER DESIGN CONDITIONS:
C     1.  MACH NO AND LOSS SLOPES ARE INDEPENDENT OF PRESSURE
C         RATIO.  HOWEVER,GHLLD MUST BE REPLACED.
C     2.  MACH NO INTERCEPT CAN BE MULTIPLIED BY (PQBD/6.0)**1/6
C         (APPROXIMATION ONLY)
C     3.  BACKBONE CURVES INCLUDING AVQUQD MUST BE REPLACED
C
C  SECURITY       UNCLASSIFIED
C
C  RESTRICTIONS   GE CLASS 1
C
C  COPYRIGHT (C)  1991 BY GE AIRCRAFT ENGINES
C
C########################################################################
C
      SAVE
      DIMENSION AWQN(15),AXMLL(15),AXMSL(15),AVQUQD(15)
      DIMENSION ASSP1(13)
      DIMENSION ASSP2(13)
      DIMENSION AGHLL(15),AGLSL(15),AXLSSL(13),APBS(13)
      DIMENSION TAB(3)
C
      DATA RDPNRM,PNRML,PNRMH/0.1,0.0,140.0/
      DATA GHLLD,WRBD,SWR/3.3940,40.0,1.0/
C
      DATA AWQN/.213,.213,.213,.225,.230,.240,.250,.2714,.3075,
     &  .3555,.4000,.3882,.3694,.3492,.3320/
      DATA AXMLL/0.,.11,.22,.33,.44,.55,.66,.745,.820,.8750,
     &  .9150,.9400,.9800,1.0,1.0/
      DATA AXMSL/0.,.10,.20,.30,.40,.50,.60,.730,.910,1.08,
     &  1.00,.500,.200,.100,.100/
      DATA AVQUQD/.47,.47,.47,.4825,.4950,.5180,.5520,.6120,
     &  .7080,.8450,1.0,.9833,.9500,.9167,.8833/
      DATA AGHLL/2.05,2.07,2.102,2.162,2.220,2.3614,2.591,
     &  2.905,3.0724,3.2392,3.3940,3.2494,3.1051,2.9569,2.8087/
      DATA AGLSL/1.230,.776,.6756,.6230,.5926,.5738,.5522,.5178,
     &  .4332,.4147,.4819,.6450,.7969,.8817,.9665/
      DATA AXLSSL/0.,.06,.125,.202,.310,.409,.830,1.460,1.790,
     &  1.880,1.90,1.90,1.90/
      DATA APBS/1.0,1.121,1.436,1.984,2.654,3.380,4.174,4.989,
     &  5.824,6.654,7.500,8.402,9.319/
      DATA ASSP1/.0057,.0057,.0057,.0057,.0057,.0057,.0057,
     &  .0057,.0057,.0057,.0049,.0045,.0045/
      DATA ASSP2/0.,0.,0.,0.,0.,0.,0.,.003,.004,.0057,.0078,
     &  .0091,.0091/
C
      DATA CR1,CR2/.32,.1488/
C
C  GE16 HPC VSV SCHEDULE EFFECTS AND (NOMINAL) VSV SCHEDULE
C
      DATA CVS1,CVS2/-3.5,52.5/
      DATA CVS3,CVS4,CVS6/75.,100.,20./
      DATA CVS7/1./
      DATA CVS8,CVS9/75.,100./
      DATA CVS10,CVS11,CVS12/.0121662,-3.71251,249.7/
      DATA CVS13,CVS14/-.675E-4,-.0054/
      DATA CVS15,CVS16,CVS116/57.,100.,75./
      DATA CVS17,CVS18,CVS19,CVS20/0.,0.,.8,-35./
      DATA CVS119,CVS120/-1.47,135.25/
      DATA CVS21/.0012/
C
C  THIS NOMINAL VSV SCHEDULE IS FOR REFERENCE ONLY
C
      DATA TAB / 52.5, 2.5,-21.5/
C
C  COMPUTE SCALARS WHICH MODEL THE EFFECTS OF
C  OFF-SCHEDULE VSV OPERATION
```

```
C
C  FIRST LIMIT THE (INPUT) VSV ANGLE TO WITHIN
C  THE PROPER RANGE
C
      VSV=AMAX1(CVS1,ZVSV)
      VSV=AMIN1(VSV,CVS2)
C
C  COMPUTE THE NOMINAL VSV ANGLE FOR THIS SPEED
C
      XNRU=AMAX1(CVS3,PNRM)
      XNRU=AMIN1(XNRU,CVS4)
      DEL=(XNRU-CVS3)/CVS6
      IX=DEL
      BOTT=TAB(IX+1)
      VSVNOM=BOTT+(DEL-IX)*(TAB(IX+2)-BOTT)
C
C  COMPUTE THE AMOUNT THAT THE INLET GUIDE
C  VANES ARE OFF THE SCHEDULE (USING THE VSV
C  ANGLE)
C
      DVPI=VSV-VSVNOM
      DVSV=DVPI
C
C  COMPUTE THE FRACTIONAL CHANGE IN WQN AND GHR
C
      PNRU=AMAX1(CVS8,PNRM)
      PNRU=AMIN1(PNRU,CVS9)
      OFFSET=(CVS10*PNRU+CVS11)*PNRU+CVS12
      DUM=DVPI+OFFSET
      SVSVW=((CVS13*DUM+CVS14)*DUM+1.)/((CVS13*OFFSET+
     &CVS14)*OFFSET+1.)
C
C  COMPUTE THE FRACTIONAL CHANGE IN GLSL
C
      PNRU=AMAX1(CVS15,PNRM)
      IF(PNRU.GT.CVS116)GO TO 30
      OFFSET=((CVS17*PNRU+CVS18)*PNRU+CVS19)*PNRU+CVS20
      GO TO 40
   30 CONTINUE
      PNRU=AMIN1(PNRU,CVS16)
      OFFSET=CVS119*PNRU+CVS120
   40 CONTINUE
      DUM=DVPI+OFFSET
      SLOSS=(CVS21*DUM*DUM+1.)/(CVS21*OFFSET*OFFSET+1.)
      SVSVL=SLOSS*SVSVW
C
C  END OF VSV SCALAR CALCULATION
C
C  FLOW CALCULATION STARTS HERE
C
      DEL=(PNRM-PNRML)*RDPNRM
      IX=DEL
      IX=MAX0(IX,0)
      IX=MIN0(IX,13)
      IXP1=IX+1
      IXP2=IX+2
      BOTT=AWQN(IXP1)
      FSQD=DEL-IX
      WQN=BOTT+FSQD*(AWQN(IXP2)-BOTT)
      WRLL=WQN*PNRM
      BOTT=AXMLL(IXP1)
      XMLL=BOTT+FSQD*(AXMLL(IXP2)-BOTT)
      BOTT=AXMSL(IXP1)
      XMSL=BOTT+FSQD*(AXMSL(IXP2)-BOTT)
      XMSL=XMSL/GHLLD
      XMJ=XMLL-XMSL*GH
      XMJ=AMIN1(1.0,XMJ)
      WRB=(1.0+.2*XMJ*XMJ)
      WRB=(1.0+.2*XMLL*XMLL)/WRB
      WRB=WRLL*WRB*WRB*WRB*XMJ/XMLL
C
      WRB=SWR*WRB
C
C  CALCULATION OF REYNOLDS NUMBER SCALARS ON FLOW AND LOSS
C
      DRN=CR1*(1.-1./RNI)/WRB
      DRN=AMAX1(DRN,-.3)
```

```
      SRWB=1.+DRN
      SRGLSL=1.-DRN/CR2
      SRWB=AMIN1(1.,SRWB)
      SRGLSL=AMAX1(1.,SRGLSL)
C
C  END OF REYNOLDS NUMBER SCALAR CALCULATION
C
      WRB=ZSUW*SVSVW*SRWB*WRB
C
C  FLOW CALCULATION ENDS
C
C  LOSS CALCULATION STARTS HERE
C
      BOTT=AGHLL(IXP1)
      GHLL=BOTT+FSQD*(AGHLL(IXP2)-BOTT)
      BOTT=AGLSL(IXP1)
      GLSL=BOTT+FSQD*(AGLSL(IXP2)-BOTT)
      GLSL=ZSUL*SVSVL*SRGLSL*GLSL
      BOTT=AVQUQD(IXP1)
      VQUQD=BOTT+FSQD*(AVQUQD(IXP2)-BOTT)
      DEL=(VQUQD-0.0)*10.
      IX=DEL
      IX=MAX0(IX,0)
      IX=MIN0(IX,11)
      IXP1=IX+1
      IXP2=IX+2
      BOTT=AXLSSL(IXP1)
      FSQD=DEL-IX
      XLSSL=BOTT+FSQD*(AXLSSL(IXP2)-BOTT)
      IF(GH.GT.0.0)XLSSL=XLSSL/GHLLD
      GLSD=XLSSL*GH*GH
      GHR=SVSVW*(GHLL+GH)
      GHRI=GHR-GLSL-GLSD
C
C  PRESSURE RATIO AND WORK CALCULATION START HERE
C
      GUTR=0.000022956*PNRM*PNRM
      TQ=GUTR*GHR+1.0
      TQI=GUTR*GHRI+1.0
      TQIRT=.45*TQI+.55
      TQIRT1=TQI/TQIRT
      TQIRT=0.5*(TQIRT+TQIRT1)
      TQIRT1=TQI/TQIRT
      TQIRT=0.5*(TQIRT+TQIRT1)
      PQB=TQI*TQI*TQI*TQIRT
      HRM=GHR*GUTR*.239953
C
C
      RETURN
      END
```

```
      SUBROUTINE TIME0
C#######################################################################
C
C  DESCRIPTION
C    THIS ROUTINE PERFORMS A ZERO TIME BALANCE OF THE MODEL
C
C  SECURITY        UNCLASSIFIED
C
C  RESTRICTIONS    GE CLASS 1
C
C  COPYRIGHT (C)   1991 BY GE AIRCRAFT ENGINES
C
C#######################################################################
C
      SAVE
      LOGICAL FADCIC
C
      COMMON /CLM1/GHHG,TIME,DT,PLA,ZPAMB,ZP2,ZT2,ZVSV2,ZWFM,
     &  ZA8,ZSUW2,ZSUL2,ZSUW41,ZSUL41,ZWB3,XNH,P5QG,XNHDOT,
     &  PNRM2,P3Q2,TMHPC,TMPC,TMHPT,YTFFH,YP8,HL41RG,FADCIC,
     &  W2R,HRM41,XNRM41
C
      COMMON /AIR/ W2,W3,W4,W5,P3,P4,P5,T3,T4,T5
C
      DIMENSION TAB(3),AWFQP(-2:6),AA8(-2:6),AGHHG(-2:6),AP5QG(-2:6),
     & ATMPC(-2:6),AHLRG(-2:6)
C
      DATA CVS3,CVS4,CVS6/75.,100.,20./
      DATA TAB /    52.5,   2.5,-21.5/
      DATA AWFQP/50.,10.,6.,17.0,26.2,40.3,61.4,90.9,119./
      DATA AA8/50.,10.,6.,128.,86.,79.,80.,87.6,102./
      DATA AGHHG/50.,10.,6.,.491,.472,.346,.280,.210,.0791/
      DATA AP5QG/50.,10.,6.,.0948,.0814,.0715,.0649,.0589,.0510/
      DATA AHLRG/50.,10.,6.,.00269,.00362,.00453,.00519,.00572,.00691/
      DATA ATMPC/50.,10.,6.,946.,1040.,1129.,1218.,1329.,1423./
C
      PNRM2=PLA
      XNHD=PNRM2*SQRT(ZT2)/.195454
      XNH=XNHD
      ZA8=FCT(PLA,AA8)
      ZWFM=ZP2*FCT(PLA,AWFQP)
      GHHG=FCT(PLA,AGHHG)
      P5QG=FCT(PLA,AP5QG)
      TMPC=FCT(PLA,ATMPC)
      HL41RG=FCT(PLA,AHLRG)
C
C  USE MAP NOMINAL VSV ANGLE FOR CONTROL
C
      XNRU=MIN(CVS4,MAX(CVS3,PNRM2))
      DEL=(XNRU-CVS3)/CVS6
      IX=DEL
      BOTT=TAB(IX+1)
      VSVNOM=BOTT+(DEL-IX)*(TAB(IX+2)-BOTT)
      ZVSV2=VSVNOM
C
   10 CONTINUE
      XNDOT1=XNHDOT
C
      CALL CLM
C
      XNH=XNHD
C
      CALL CNTRL(PLA,DT,ZP2,ZT2,XNH,XNHDOT,P3,PQD,ZWFM,ZA8,
     & ZVSV2)
C
      IF(ABS(XNHDOT-XNDOT1).LT.1.)GOTO 20
      ICNT=ICNT+1
      IF(ICNT.EQ.100)THEN
        WRITE(06,100)ICNT
  100   FORMAT('ZERO TIME BALANCE NOT ACHIEVED AFTER ',I3,
     &    ' ITERATIONS')
        STOP
      ENDIF
      GOTO 10
   20 CONTINUE
      FADCIC=.FALSE.
C                            C
                                   RETURN
                                   END
```

```
      SUBROUTINE TRB41X(P4Q42,XN41,T41,HL41RG,FADCIC,WRB41,H41R,
     &  HL41RP,XNRM41)
C##########################################################################
C
C   DESCRIPTION
C     THIS ROUTINE PERFORMS THE TURBINE MAP EVALUATION
C
C   SECURITY        UNCLASSIFIED
C
C   RESTRICTIONS    GE CLASS 1
C
C   COPYRIGHT (C)   1991 BY GE AIRCRAFT ENGINES
C
C##########################################################################
C
      SAVE
      LOGICAL FADCIC
C
      DIMENSION NX(5),IX(5)
      DIMENSION AWRH41(0: 30),AGNL41(0: 32),AGRL41(0: 32),
     & AGSP41(0: 32),AGSN41(0: 32)
C
C   TRB41X CONSTANTS
C
      DATA RADP41,DGMN41,DGMH41/        0.98441660E-06,     0.00143472,
     &    2902.97100000/
C
C   TRB41X CURVES
C
      DATA (AWRH41 (I),I=  0, 30)/          15.00000000,
     &     26.40000000,     52.80000000,     79.20000000,    105.60000000,
     &    132.00000000,    158.40000000,    184.80000000,    211.20000000,
     &    237.60000000,    264.00000000,    290.40000000,    316.80000000,
     &    343.20000000,    369.60000000,    396.00000000,     19.69800000,
     &     19.69800000,     19.69800000,     19.60200000,     19.31500000,
     &     19.05700000,     18.85600000,     18.71200000,     18.62100000,
     &     18.57800000,     18.57600000,     18.61000000,     18.67400000,
     &     18.76200000,     18.86900000/
      DATA (AGNL41 (I),I=  0, 32)/          16.00000000,
     &      0.00000000,      0.00100000,      0.01000000,      0.02000000,
     &      0.03000000,      0.04000000,      0.05000000,      0.05400000,
     &      0.06000000,      0.07000000,      0.08000000,      0.09000000,
     &      0.10000000,      0.15000000,      0.20000000,      0.24000000,
     &      0.40740000,      0.40890000,      0.42360000,      0.44430000,
     &      0.47210000,      0.51270000,      0.58390000,      0.63060000,
     &      0.70070000,      0.81750000,      0.93420000,      1.05100000,
     &      1.16780000,      1.75170000,      2.33560000,      2.80270000/
      DATA (AGRL41 (I),I=  0, 32)/          16.00000000,
     &      0.00000000,      0.00100000,      0.01000000,      0.02000000,
     &      0.03000000,      0.04000000,      0.05000000,      0.05400000,
     &      0.06000000,      0.07000000,      0.08000000,      0.09000000,
     &      0.10000000,      0.15000000,      0.20000000,      0.24000000,
     &      0.00000000,      1.67600000,     15.49200000,     29.31300000,
     &     41.54800000,     53.25200000,     65.66700000,     70.53300000,
     &    100.59200000,    180.57500000,    272.83900000,    390.43100000,
     &    538.05000000,   2008.80000000,   6877.75200000, 27036.08000000/
      DATA (AGSP41 (I),I=  0, 32)/          16.00000000,
     &      0.00000000,      0.00100000,      0.01000000,      0.02000000,
     &      0.03000000,      0.04000000,      0.05000000,      0.05400000,
     &      0.06000000,      0.07000000,      0.08000000,      0.09000000,
     &      0.10000000,      0.15000000,      0.20000000,      0.24000000,
     &      1.31175200,      1.31289900,      1.31993800,      1.31842800,
     &      1.30205800,      1.26031400,      1.16515400,      1.10061000,
     &      1.03193500,      0.96089570,      0.92548600,      0.92691050,
     &      0.97274430,      0.97274430,      0.97274430,      0.97274430/
      DATA (AGSN41 (I),I=  0, 32)/          16.00000000,
     &      0.00000000,      0.00100000,      0.01000000,      0.02000000,
     &      0.03000000,      0.04000000,      0.05000000,      0.05400000,
     &      0.06000000,      0.07000000,      0.08000000,      0.09000000,
     &      0.10000000,      0.15000000,      0.20000000,      0.24000000,
     &      1.31175200,      1.31289900,      1.31993800,      1.31842800,
     &      1.30205800,      1.26031400,      1.16515400,      1.10061000,
     &      1.03193500,      0.96089570,      0.92548600,      0.92691050,
     &      0.97274430,      0.97274430,      0.97274430,      0.97274430/
C
C   CALCULATE IDEAL AND ACTUAL WORK
C
```

```
      DPQP=P4Q42-1.
      HIRB=(((-.0011005*DPQP+.0086553)*DPQP-.028989)*DPQP+
     &.065426)*DPQP
      FCTR=(.342711E-5*T41-.0091286)*(1.22222+DPQP)+1.0039
      H41IR=HIRB*FCTR
      H41R=MAX(0.0,H41IR-HL41RG)
C
C  MAP INDEPENDENT VARIABLES
C
      XN41R=XN41/SQRT(T41)
      HRM41=H41R
      XNRM41=XN41R
      XNRS41=XNRM41*XNRM41
C
C  CALCULATE MAP FLOW
C
      GMNH41=0.9258201*XNRM41
      GMN41=DGMN41*XNRS41+DGMH41*HRM41
      CALL CRVX1(XNRM41,AWRH41,NX(1),FADCIC,IX(1),WRMH41)
      GMN41=AMIN1(GMN41,GMNH41)
      XMJ41=GMN41/XNRM41
      TSQTP=6.-XMJ41*XMJ41
      WRB41=.00794072*TSQTP*TSQTP*TSQTP*XMJ41*WRMH41
C
C  CALCULATE MAP LOSS
C
      CALL CRVX1(HRM41,AGNL41,NX(2),FADCIC,IX(2),GHNL41)
      CALL CRVX1(HRM41,AGRL41,NX(3),FADCIC,IX(3),GHRL41)
      UPRS41=RADP41*XNRS41
      GHN41=HRM41/(UPRS41)
      DGN41=GHN41-GHNL41
      DUM=DGN41*DGN41
C
      IF(DGN41.GE.0)THEN
        CALL CRVX1(HRM41,AGSP41,NX(4),FADCIC,IX(4),GLS41)
      ELSE
        CALL CRVX1(HRM41,AGSN41,NX(5),FADCIC,IX(5),GLS41)
      ENDIF
C
      GLS41=DGN41*DGN41*GLS41
      HRL41=(GLS41*UPRS41*(0.27423152-HRM41-GHRL41*7.9893239E-5)
     & +GHRL41*7.9893239E-5)
C
C  ACTUAL LOSS
C
      HL41RP=H41IR*(1.-(H41R/(H41R+HRL41)))
C
C
      RETURN
      END
```

```
      SUBROUTINE CRVX1 (XN,A,NX,LSWTCH,IX,ZN)
C########################################################################
C
C  DESCRIPTION
C    ONE DIMENSIONAL UNEQUALLY SPACED CURVE EVALUATION WITHOUT
C    EXTRAPOLATION.
C
C  SECURITY        UNCLASSIFIED
C
C  RESTRICTIONS    GE CLASS 1
C
C  COPYRIGHT (C)   1991 BY GE AIRCRAFT ENGINES
C
C########################################################################
C
      LOGICAL LSWTCH
      DIMENSION A (0:*)
C
      IF (LSWTCH) THEN
          NX = A (0)
          X = AMAX1 (AMIN1 (XN, A (NX)), A (1))
          IL = 1
          IH = NX
          IP = 1
   10     IX = (IL + IH) / 2
          IF (IX .NE. IP) THEN
              IP = IX
              IF (A (IX) .LT. X) THEN
                  IL = IP
              ELSE
                  IH = IP
              END IF
              GO TO 10
          END IF
          IX1 = IX + NX
          ZN = A (IX) - A (IX + 1)
          ZN = A (IX1) + (X - A (IX)) * (A (IX1) - A (IX1 + 1)) / ZN
          RETURN
      ELSE
          X = AMAX1 (AMIN1 (XN, A (NX)), A (1))
          IF (A (IX + 1) .LT. X) THEN
              IX = IX + 1
          ELSE
              IF (A (IX) .GT. X) IX = IX - 1
          END IF
          IX1 = IX + NX
          ZN = A (IX) - A (IX + 1)
          ZN = A (IX1) + (X - A (IX)) * (A (IX1) - A (IX1 + 1)) / ZN
          RETURN
      END IF
C
      END
      REAL FUNCTION FCT(X,A)
C########################################################################
C
C  DESCRIPTION
C    ONE DIMENSIONAL EQUALLY SPACED CURVE EVALUATION WITHOUT
C    EXTRAPOLATION.
C
C  SECURITY        UNCLASSIFIED
C
C  RESTRICTIONS    GE CLASS 1
C
C  COPYRIGHT (C)   1991 BY GE AIRCRAFT ENGINES
C
C########################################################################
C
      REAL A(-2:*),DEL,X
      INTEGER IX
      IF(X.GT.A(-2))THEN
        DEL=(X-A(-2))/A(-1)+1.
        IF(DEL.LT.A(0))THEN
          IX=DEL
          FCT=A(IX)+(DEL-AINT(DEL))*(A(IX+1)-A(IX))
        ELSE
          FCT=A(IFIX(A(0)))
        ENDIF
                              ELSE
                                FCT=A(1)
                              ENDIF
                              RETURN
                              END
```

```
      SUBROUTINE VD7(H1,P1X,DP1X,W3,H2,P2,W2)
C########################################################################
C
C  DESCRIPTION
C    VOLUME DYNAMICS SUBROUTINE THAT CALCULATES THE PRESSURE AND
C    ENTHALPY OF THE VOLUME CENTER (P7,H7) AND THE INLET PRESSURE (P5)
C    REQUIRED TO MATCH THE INLET MASS FLOWRATE THAT IS SET BY THE TURBINE
C    SINCE THE TIME STEP (DT) IS APPROXIMATELY AN ORDER OF MAGNITUDE
C    LARGER THAN USUAL FOR THIS VOLUME DYNAMICS LOGIC THE VOLUME SIZE (V2)
C    REQUIRED FOR STABILITY IS LARGE.
C
C  SECURITY        UNCLASSIFIED
C
C  RESTRICTIONS    GE CLASS 1
C
C  COPYRIGHT (C)   1991 BY GE AIRCRAFT ENGINES
C
C########################################################################
C
      SAVE
      LOGICAL LTIME0
C
      DIMENSION GAMMA(-2:7)
C
      DATA GAMMA/0.,100.,7.,1.401453,1.390191,1.366160,1.344462,
     & 1.328095,1.315496,1.305037/
      DATA CMOM,CCONT/386.04,9337.92/
      DATA DT,AQL,V2 / 0.002, 7.5, 3000./
      DATA LTIME0/.TRUE./
C
      P1=P1X+DP1X
C
C  FIRST PASS THROUGH INITIALIZE PROPERTIES AT VOLUME CENTER AND EXIT
C
      IF(LTIME0)THEN
        LTIME0=.FALSE.
        P2=P1
        H2=H1
        W3=W2
        CM=0.5*CMOM*DT*AQL
        CCE=DT*CCONT/V2
      ENDIF
C
      GM=FCT(H1,GAMMA)
      QGM=1./GM
      CE=(GM-1.)*CCE
      DCM=CM*CE
      WHN3=W3*H2
      DP1=P1-P2
      DP1ACT=CE*(W2*H1-WHN3)
      P1X=P1X+DP1ACT-DP1
C
      CALL VDPHXX(DP1ACT,H1,W3,WHN3,W2,H2,P2,CM,CE,DCM,QGM)
C
C
      RETURN
      END
      SUBROUTINE VDPHXX(DP,H1,WN3,WHN3,W2,H2,P2,CM,CE,DCM,QGM)
C########################################################################
C
C  DESCRIPTION
C    BASIC VOLUME DYNAMICS SUBROUTINE
C
C  SECURITY        UNCLASSIFIED
C
C  RESTRICTIONS    GE CLASS 1
C
C  COPYRIGHT (C)   1991 BY GE AIRCRAFT ENGINES
C
C########################################################################
C
      SAVE
C
      WN2=W2+DCM*WHN3+CM*DP
      IF(WN2)10,20,30
   10 WN2=WN2/(1.+DCM*H2)
   20 WHN2=WN2*H2

                GO TO 40
             30 WN2=WN2/(1.+DCM*H1)
                WHN2=WN2*H1
             40 H2=H2+CE*H2*(WHN2-WHN3-QGM*H2*(WN2-WN3))/P2
                P2=P2+CE*(WHN2-WHN3)
                W2=WN2+WN2-W2
          C
          C
                RETURN
                END
```

A.3 SAMPLE INPUT

The program was compiled using the Lahey F77L Version 4.10 FORTRAN compiler and the OPTILINK linker. The executable is called RTM.EXE. The program is run by entering the following command:

A.3.1 RTM

The format of the input data is slightly different than standard file based NAMELIST input since the program is expecting input from the screen. To run the program using the sample data file, type the following command:

A.3.2 RTM < INPUT.DAT (See Table A2)

Table A2 - Listing of Input.Dat

```
PLA=100. &END
FTIME=10., PLA=70. &END
FTIME=20., PLA=100. &END
FTIME=-1 &END
```

A.4 SAMPLE OUTPUT

Table A3

```
 &INPUT TIME=0.000000,FTIME=1.00000,PLA=100.000,ZP2=14.6960,ZT2=518.670,
ZPAMB=14.6960,ZWB3=0.000000 &END
     TIME    PLA    PNRM    P3Q2    ZWFM     ZA8    ZVSV2     XNH    XNHDOT     P3      W2
      0.0  100.0  100.00    6.22 1748.55  102.01   -3.50 11652.0     -0.1   91.43   38.64
      0.0  100.0  100.00    6.22 1758.36  102.47   -3.49 11650.6    -70.9   91.40   38.64
      0.2  100.0   99.99    6.22 1749.83  102.62   -3.49 11651.2      4.9   91.39   38.64
      0.4  100.0   99.99    6.22 1749.64  102.61   -3.49 11651.3      0.7   91.40   38.64
      0.6  100.0  100.00    6.22 1749.73  102.61   -3.49 11651.5      0.5   91.40   38.64
      0.8  100.0  100.00    6.22 1749.75  102.61   -3.50 11651.6      0.4   91.40   38.64
      1.0  100.0  100.00    6.22 1749.77  102.61   -3.50 11651.6      0.4   91.40   38.64
 &INPUT TIME=1.02000,FTIME=1.00000,PLA=100.000,ZP2=14.6960,ZT2=518.670,
ZPAMB=14.6960,ZWB3=0.000000 &END
     TIME    PLA    PNRM    P3Q2    ZWFM     ZA8    ZVSV2     XNH    XNHDOT     P3      W2
      1.0   70.0  100.00    6.22 1749.77  102.61   -3.50 11651.6      0.4   91.40   38.64
      1.0   70.0  100.00    6.22 1723.52  101.07   -3.50 11651.6      0.4   91.40   38.64
      1.2   70.0   98.77    5.90 1481.76   86.90   -1.70 11477.2  -1580.5   86.64   37.71
      1.4   70.0   95.32    5.43 1438.70   75.89    2.70 11060.3  -2299.1   79.87   34.77
      1.6   70.0   91.52    4.95 1267.15   71.62   12.12 10621.1  -2160.6   72.68   31.61
      1.8   70.0   87.91    4.44 1108.45   68.87   21.08 10203.6  -1957.0   65.32   28.73
      2.0   70.0   84.81    4.06 1007.67   67.48   28.67  9849.5  -1619.1   59.60   26.29
      2.2   70.0   82.26    3.74  905.44   66.94   34.92  9558.5  -1329.4   54.96   24.48
      2.4   70.0   80.16    3.48  808.33   67.11   40.06  9318.8  -1100.7   51.14   23.13
      2.6   70.0   78.40    3.29  764.34   67.03   44.41  9116.3   -940.3   48.35   22.06
      2.8   70.0   76.91    3.16  746.16   67.18   48.05  8946.2   -779.6   46.38   21.13
      3.0   70.0   75.68    3.05  730.02   67.59   51.07  8805.8   -642.0   44.79   20.39
      3.2   70.0   74.67    2.96  713.63   68.22   52.50  8690.3   -528.2   43.49   19.81
      3.4   70.0   73.84    2.89  697.42   69.00   52.50  8595.3   -434.6   42.41   19.35
      3.6   70.0   73.16    2.82  682.01   69.87   52.50  8517.1   -357.7   41.51   19.00
      3.8   70.0   72.59    2.77  667.79   70.78   52.50  8452.7   -294.4   40.77   18.71
      4.0   70.0   72.13    2.73  654.97   71.69   52.50  8399.7   -242.3   40.15   18.49
      4.2   70.0   71.75    2.70  643.61   72.58   52.50  8356.1   -199.3   39.63   18.31
      4.4   70.0   71.43    2.67  633.67   73.42   52.50  8320.3   -163.9   39.21   18.17
      4.6   70.0   71.18    2.64  625.07   74.20   52.50  8290.8   -134.8   38.85   18.06
      4.8   70.0   70.96    2.62  617.68   74.91   52.50  8266.5   -110.7   38.56   17.96
      5.0   70.0   70.79    2.61  611.37   75.56   52.50  8246.6    -90.9   38.32   17.89
      5.2   70.0   70.65    2.59  606.03   76.12   52.50  8230.3    -74.6   38.11   17.83
      5.4   70.0   70.53    2.58  601.52   76.62   52.50  8216.9    -61.2   37.95   17.78
      5.6   70.0   70.43    2.57  597.73   77.05   52.50  8205.9    -50.2   37.81   17.74
      5.8   70.0   70.35    2.57  594.56   77.42   52.50  8196.9    -41.1   37.70   17.71
      6.0   70.0   70.29    2.56  591.91   77.74   52.50  8189.5    -33.7   37.61   17.68
      6.2   70.0   70.24    2.55  589.72   78.01   52.50  8183.5    -27.6   37.53   17.66
      6.4   70.0   70.19    2.55  587.91   78.23   52.50  8178.5    -22.5   37.47   17.65
      6.6   70.0   70.16    2.55  586.41   78.42   52.50  8174.5    -18.4   37.41   17.63
      6.8   70.0   70.13    2.54  585.18   78.58   52.50  8171.2    -15.1   37.37   17.62
      7.0   70.0   70.11    2.54  584.17   78.71   52.50  8168.4    -12.3   37.34   17.61
      7.2   70.0   70.09    2.54  583.34   78.82   52.50  8166.2    -10.1   37.31   17.61
      7.4   70.0   70.07    2.54  582.67   78.91   52.50  8164.4     -8.2   37.28   17.60
      7.6   70.0   70.06    2.54  582.13   78.99   52.50  8163.0     -6.7   37.27   17.59
      7.8   70.0   70.05    2.53  581.68   79.05   52.50  8161.7     -5.5   37.25   17.59
      8.0   70.0   70.04    2.53  581.33   79.10   52.50  8160.8     -4.5   37.24   17.59
      8.2   70.0   70.03    2.53  581.05   79.14   52.50  8160.0     -3.7   37.23   17.58
      8.4   70.0   70.03    2.53  580.82   79.17   52.50  8159.3     -3.0   37.22   17.58
      8.6   70.0   70.02    2.53  580.64   79.20   52.50  8158.8     -2.4   37.21   17.58
      8.8   70.0   70.02    2.53  580.50   79.23   52.50  8158.3     -2.0   37.21   17.58
      9.0   70.0   70.01    2.53  580.40   79.24   52.50  8158.0     -1.6   37.20   17.58
      9.2   70.0   70.01    2.53  580.32   79.26   52.50  8157.7     -1.3   37.20   17.58
      9.4   70.0   70.01    2.53  580.26   79.27   52.50  8157.4     -1.1   37.19   17.58
      9.6   70.0   70.01    2.53  580.21   79.28   52.50  8157.2     -0.9   37.19   17.58
      9.8   70.0   70.01    2.53  580.19   79.29   52.50  8157.1     -0.7   37.19   17.58
 &INPUT TIME=10.0000,FTIME=10.0000,PLA=70.0000,ZP2=14.6960,ZT2=518.670,
ZPAMB=14.6960,ZWB3=0.000000 &END
     TIME    PLA    PNRM    P3Q2    ZWFM     ZA8    ZVSV2     XNH    XNHDOT     P3      W2
     10.0  100.0   70.00    2.53  580.17   79.30   52.50  8157.0     -0.6   37.19   17.57
     10.0  100.0   70.00    2.53  588.88   80.49   52.50  8157.0     -0.6   37.19   17.57
     10.2  100.0   70.39    2.58  683.41   93.41   52.50  8211.6    510.7   37.91   17.67
     10.4  100.0   71.67    2.70  793.13  108.40   52.50  8371.1   1035.3   39.73   18.20
     10.6  100.0   73.88    2.92  920.46  125.81   52.50  8641.1   1618.6   42.89   19.23
     10.8  100.0   77.17    3.25 1038.41  144.57   46.13  9036.1   2239.4   47.82   20.93
     11.0  100.0   81.08    3.60 1012.14  148.23   36.34  9492.3   2226.8   52.96   23.67
     11.2  100.0   84.63    3.98 1050.69  129.19   27.62  9898.6   1864.3   58.43   26.35
     11.4  100.0   87.53    4.35 1162.37  111.19   20.55 10228.2   1461.2   63.90   28.55
     11.6  100.0   89.77    4.69 1300.99  102.92   15.09 10482.6   1147.2   68.93   30.26
     11.8  100.0   91.57    4.96 1411.02   99.92   10.68 10688.0    935.1   72.91   31.63
     12.0  100.0   93.04    5.19 1497.05   98.78    7.07 10856.2    770.2   76.28   32.77
```

12.2	100.0	94.25	5.38	1555.57	98.64	4.09	10995.1	636.2	79.00	33.75
12.4	100.0	95.26	5.53	1597.27	98.87	2.08	11109.8	525.3	81.20	34.58
12.6	100.0	96.08	5.65	1628.95	99.46	1.11	11204.1	435.8	83.01	35.27
12.8	100.0	96.77	5.75	1652.55	99.91	0.31	11282.3	358.0	84.47	35.85
13.0	100.0	97.33	5.83	1671.92	100.30	-0.36	11346.9	295.5	85.69	36.33
13.2	100.0	97.80	5.90	1687.35	100.65	-0.91	11400.1	244.0	86.69	36.73
13.4	100.0	98.18	5.95	1699.71	100.96	-1.36	11444.1	201.4	87.51	37.06
13.6	100.0	98.50	6.00	1709.65	101.22	-1.73	11480.4	166.3	88.20	37.34
13.8	100.0	98.76	6.04	1717.69	101.45	-2.04	11510.4	137.2	88.76	37.57
14.0	100.0	98.98	6.07	1724.22	101.64	-2.30	11535.1	113.3	89.22	37.75
14.2	100.0	99.16	6.10	1729.53	101.81	-2.51	11555.5	93.5	89.60	37.91
14.4	100.0	99.30	6.12	1733.86	101.94	-2.68	11572.4	77.2	89.92	38.04
14.6	100.0	99.43	6.14	1737.40	102.06	-2.82	11586.3	63.7	90.18	38.14
14.8	100.0	99.53	6.15	1740.29	102.15	-2.94	11597.8	52.6	90.39	38.23
15.0	100.0	99.61	6.16	1742.66	102.23	-3.04	11607.3	43.4	90.57	38.30
15.2	100.0	99.68	6.17	1744.60	102.30	-3.12	11615.1	35.8	90.72	38.36
15.4	100.0	99.73	6.18	1746.20	102.36	-3.19	11621.5	29.6	90.84	38.41
15.6	100.0	99.78	6.19	1747.50	102.40	-3.24	11626.9	24.4	90.94	38.45
15.8	100.0	99.82	6.19	1748.57	102.44	-3.29	11631.2	20.1	91.02	38.49
16.0	100.0	99.85	6.20	1749.44	102.47	-3.32	11634.9	16.6	91.09	38.51
16.2	100.0	99.88	6.20	1750.16	102.50	-3.35	11637.9	13.7	91.14	38.54
16.4	100.0	99.90	6.21	1750.75	102.52	-3.38	11640.3	11.3	91.19	38.55
16.6	100.0	99.92	6.21	1751.23	102.54	-3.40	11642.4	9.3	91.23	38.57
16.8	100.0	99.93	6.21	1751.62	102.55	-3.42	11644.1	7.7	91.26	38.58
17.0	100.0	99.94	6.21	1751.93	102.56	-3.43	11645.5	6.4	91.29	38.59
17.2	100.0	99.95	6.21	1752.19	102.57	-3.44	11646.6	5.3	91.31	38.60
17.4	100.0	99.96	6.21	1752.40	102.58	-3.45	11647.5	4.3	91.33	38.61
17.6	100.0	99.97	6.22	1752.57	102.59	-3.46	11648.3	3.6	91.34	38.62
17.8	100.0	99.97	6.22	1752.70	102.59	-3.47	11649.0	3.0	91.35	38.62
18.0	100.0	99.98	6.22	1752.81	102.60	-3.47	11649.5	2.4	91.36	38.62
18.2	100.0	99.98	6.22	1752.90	102.60	-3.48	11649.9	2.0	91.37	38.63
18.4	100.0	99.99	6.22	1752.96	102.61	-3.48	11650.3	1.7	91.38	38.63
18.6	100.0	99.99	6.22	1753.01	102.61	-3.49	11650.6	1.4	91.38	38.63
18.8	100.0	99.99	6.22	1753.05	102.61	-3.49	11650.9	1.1	91.39	38.63
19.0	100.0	99.99	6.22	1753.08	102.61	-3.49	11651.1	0.9	91.39	38.64
19.2	100.0	99.99	6.22	1753.10	102.61	-3.49	11651.2	0.8	91.39	38.64
19.4	100.0	99.99	6.22	1753.10	102.62	-3.49	11651.4	0.6	91.40	38.64
19.6	100.0	100.00	6.22	1753.11	102.62	-3.49	11651.5	0.5	91.40	38.64
19.8	100.0	100.00	6.22	1753.11	102.62	-3.50	11651.6	0.4	91.40	38.64

```
&INPUT TIME=20.0003,FTIME=20.0000,PLA=100.000,ZP2=14.6960,ZT2=518.670,
ZPAMB=14.6960,ZWB3=0.000000 &END
```

APPENDIX B - EXAMPLE OF PIECEWISE LINEAR REAL-TIME MODEL

FOREWORD

An example of a real-time model based on a piecewise-linear approach is contained in this section. A FORTRAN listing of the model, sample input, and sample output from the model are included.

B.1 PIECEWISE LINEAR MODEL LISTING

Table B1 is a listing of the piecewise linear representation of the simplified aerothermo model. The piecewise linear model consists of two subroutines to be used in conjunction with the simplified aerothermo model. Subroutine CLM has been modified to remove the aerothermo calculations and to call SVM001 which contains the piecewise linear model. All input and output formats are the same as the aerothermodynamic model.

Table B1

```
      SUBROUTINE CLM
C########################################################################
C
C  DESCRIPTION
C
C    MODIFIED SUBROUTINE CLM FROM PROGRAM GEXX FOR STATE VARIABLE MODEL
C
C    HISTORY:
C
C       FEB   1991, SUBROUTINE CLM, GEXX VERSION 1, GENERAL ELECTRIC
C       APRIL 1992, MODIFIED FOR SVM, PRATT & WHITNEY
C
C########################################################################
      SAVE
      LOGICAL FADCIC
C
      COMMON /CLM1/GHHG,TIME,DT,PLA,ZPAMB,ZP2,ZT2,ZVSV2,ZWFM,
     &  ZA8,ZSUW2,ZSUL2,ZSUW41,ZSUL41,ZWB3,XNH,P5QG,XNHDOT,
     &  PNRM2,P3Q2,TMHPC,TMPC,TMHPT,YTFFH,YP8,HL41RG,FADCIC,
     &  W2R,HRM41,XNRM41
C
      COMMON /AIR/W2,W3,W4,W5,P3,P4,P5,T3,T4,T5
C
      DATA WRBH41,XJHP,CF8 / 18.578, 1.87,  .963/
      DATA WCL41Q,WCL5Q / .095,  .0523 /
C
      RT2=SQRT(ZT2)
      PNRM2=XNH*.195454/RT2
C
C STATE VARIABLE MODEL OF GAS TURBINE
C
      CALL SVM001(H5,W2RM,WFC,QPC)
C
C  CALL VOLUME DYNAMICS FOR STATION 7 PROPERTIES
C
      DO 100 K=1,10
        P7D5=-4.3438E-6*W5*W5*T5/P5
C
        CALL VD7(H5,P5,P7D5,W7,H7,P7,W5)
C
        FAR7=WFC/(W7-WFC)
        STFHJ=1.-.000795*(H7+900.)*FAR7
        H7J=STFHJ*H7
        T7=((H7J*3.33342E-7-1.057431E-3)*H7J+4.42213)*H7J-14.62035
C
C  NOZZLE
C
        P7QAM=P7/ZPAMB
        CF8=.02251*(P7QAM-1.)+.90
        A8=CF8*ZA8
        FP8C=.5342-.000006*T7
        FPQFPC=W7*SQRT(T7)/A8/P7/FP8C
        FPS=P7QAM*FPQFPC
```

```
        FPSH=1.895-1.625E-5*T7
        PTAREQ=FPS
        IF(FPS.LE.FPSH) PTAREQ=1.0-0.0227138*FPS+.26*FPS*FPS
        YP8=(P7QAM-PTAREQ)/PTAREQ
        W7=W7*(1.+.8*YP8)
C
  100 CONTINUE
C
      P5QG=P5/(ZP2*W2RM)
C
C  SPEED UPDATED
C
      XNH=XNH+XNHDOT*DT
C
C  METAL TEMPERATURE UPDATED
C
      TMPC=TMPC+QPC*DT*0.069
      RETURN
      END


      SUBROUTINE SVM001 (H5,W2RM,WFC,QPC)
C***********************************************************************
C  SUBROUTINE SVM001 -  PIECEWISE LINEAR STATE VARIABLE MODEL
C***********************************************************************
C
C INPUTS THROUTGH COMMON CLM1
C
C   ZP2    = ENGINE FACE TOTAL PRESSURE                     (PSIA    )
C   ZT2    = ENGINE FACE TOTAL TEMPERATURE                  (DEG.  R )
C   PNRM2  = PERCENT CORRECTED ROTOR SPEED                  (PERCENT)
C   ZWFM   = CORRECTED FUEL FLOW                            (LBM/HR)
C   P5QG   = THROTTLING PARAMETER = P5/(P2*W2R)             (SEC/LBM)
C   ZVSV2  = VARIABLE VANE ANGLE                            (DEGREES)
C   ZWB3   = PERCENT COMPRESSOR DISCHARGE BLEED             (DEC. FRAC.)
C   TMPC   = METAL TEMPERATURE                              (DEG. R)
C
C OUTPUTS THROUGH COMMON CLM1
C
C   XNHDOT = ROTOR SPEED DERIVATIVE                         (RPM/SEC)
C   W2R    = CORRECTED COMPRESSOR AIRFLOW                   (LBM/SEC)
C   P3Q2   = COMPRESSOR PRESSURE RATION                     (D'LESS )
C
C OUTPUTS THROUGH COMMON AIR
C
C   W2     = PHYSICAL COMPRESSOR AIRFLOW                    (LBM/SEC)
C   P3     = COMPRESSOR EXIT TOTAL PRESSURE                 (PSIA    )
C   T3     = COMPRESSOR EXIT TOTAL TEMPERATURE              (DEG. R )
C   W5     = TURBINE EXIT FLOW                              (LBM/SEC)
C   P5     = TURBINE EXIT TOTAL PRESSURE                    (PSIA    )
C   T5     = TURBINE DISCHARGE TOTAL TEMPERATURE            (DEG. R )
C
C
C OUTPUTS THROUGH CALL LIST
C
C   H5     = TURBINE EXIT ENTHALPY                          (  ??    )
C   W2RM   = MAP CORRECTED AIRFLOW                          (LBM/SEC)
C   WFC    = FUEL FLOW                                      (LBM/SEC)
C   QPC    = HEAT TRANSFER TO METAL                         (  ??    )
C
C
C HISTORY:
C
C   APRIL 1994, PRATT & WHITNEY, DEVELOPED FOR SAE S15 COMMITTEE
C
C COMMENTS:
C
C   1.   THIS SVM IS A PIECEWISE LINEAR REPRESENTATION OF THE CORE
C        ENGINE PORTION OF THE GEXX SIMPLIFIED AEROTHERMODYNAMIC
C        MODEL.  THE VOLUME DYNMAICS ARE NOT INCLUDED IN THE LINEAR
C        REPRESENTATION.
C
C   2.   THE MODEL USES 13 BASE POINTS SCHEDULED WITH PERCENT CORRECTED
C        SPEED.  LINEAR MODEL VARIABLES ARE:
C
C            STATES:  ROTOR SPEED
C                     METAL TEMPERATURE
```

```
C
C          INPUTS:  CORRECTED FUEL FLOW
C                   THROTTLING PARAMETER   ( P5 / (P2 * W2R) )
C                   VARABLE VANE ANGLE
C                   PERCENT DISCHARGE BLEED
C
C          OUTPUTS: CORRECTED COMPRESSOR FLOW
C                   COMPRESSOR EXIT PRESSURE
C                   COMPRESSOR EXIT TEMPERATURE
C                   TURBINE EXIT TEMPERATURE
C
C***********************************************************************
      LOGICAL FADCIC
C
      COMMON /CLM1/GHHG,TIME,DT,PLA,ZPAMB,ZP2,ZT2,ZVSV2,ZWFM,
     &  ZA8,ZSUW2,ZSUL2,ZSUW41,ZSUL41,ZWB3,XNH,P5QG,XNHDOT,
     &  PNRM2,P3Q2,TMHPC,TMPC,TMHPT,YTFFH,YP8,HL41RG,FADCIC,
     &  W2R,HRM41,XNRM41
C
      COMMON /AIR/W2,W3,W4,W5,P3,P4,P5,T3,T4,T5
C**********************************************************************
C                  BASE POINT AND MATRIX DATA
C**********************************************************************
      DIMENSION X2BASE(13), U1BASE(13), U2BASE(13), U3BASE(13),
     *          U4BASE(13), Y1BASE(13), Y2BASE(13), Y3BASE(13),
     *          Y4BASE(13)
      DIMENSION A12MAT(13), A22MAT(13), B11MAT(13), B12MAT(13),
     *          B13MAT(13), B14MAT(13), B21MAT(13), B22MAT(13),
     *          B23MAT(13), B24MAT(13), C12MAT(13), C22MAT(13),
     *          C32MAT(13), C42MAT(13), D11MAT(13), D21MAT(13),
     *          D31MAT(13), D41MAT(13), D12MAT(13), D22MAT(13),
     *          D32MAT(13), D42MAT(13), D13MAT(13), D23MAT(13),
     *          D33MAT(13), D43MAT(13), D14MAT(13), D24MAT(13),
     *          D34MAT(13), D44MAT(13)
      DATA X2BASE /  1.0301E+03,  1.0608E+03,  1.0901E+03,  1.1179E+03,
     *  1.1442E+03,  1.1690E+03,  1.1923E+03,  1.2141E+03,  1.2344E+03,
     *  1.2531E+03,  1.2703E+03,  1.2859E+03,  1.2999E+03/
      DATA U1BASE /  2.8318E+02,  3.1536E+02,  3.5124E+02,  3.9019E+02,
     *  4.3153E+02,  4.7462E+02,  5.1881E+02,  5.6344E+02,  6.0786E+02,
     *  6.5141E+02,  6.9344E+02,  7.3331E+02,  7.7034E+02/
      DATA U2BASE /  7.5864E-02,  7.4314E-02,  7.2703E-02,  7.1027E-02,
     *  6.9285E-02,  6.7474E-02,  6.5591E-02,  6.3633E-02,  6.1599E-02,
     *  5.9485E-02,  5.7288E-02,  5.5006E-02,  5.2637E-02/
      DATA U3BASE /  5.2500E+01,  5.2500E+01,  5.2500E+01,  4.6250E+01,
     *  4.0000E+01,  3.3750E+01,  2.7500E+01,  2.1250E+01,  1.5000E+01,
     *  8.7500E+00,  2.5000E+00, -5.0000E-01, -3.5000E+00/
      DATA U4BASE /  2.0000E-02,  2.0000E-02,  2.0000E-02,  2.0000E-02,
     *  2.0000E-02,  2.0000E-02,  2.0000E-02,  2.0000E-02,  2.0000E-02,
     *  2.0000E-02,  2.0000E-02,  2.0000E-02,  2.0000E-02/
      DATA Y1BASE /  1.7062E+01,  1.8219E+01,  1.9528E+01,  2.0975E+01,
     *  2.2549E+01,  2.4239E+01,  2.6032E+01,  2.7916E+01,  2.9881E+01,
     *  3.1913E+01,  3.4002E+01,  3.6135E+01,  3.8301E+01/
      DATA Y2BASE /  2.5464E+00,  2.7406E+00,  2.9621E+00,  3.2089E+00,
     *  3.4788E+00,  3.7700E+00,  4.0802E+00,  4.4075E+00,  4.7498E+00,
     *  5.1051E+00,  5.4714E+00,  5.8466E+00,  6.2286E+00/
      DATA Y3BASE /  1.3765E+00,  1.4073E+00,  1.4392E+00,  1.4722E+00,
     *  1.5060E+00,  1.5408E+00,  1.5762E+00,  1.6123E+00,  1.6490E+00,
     *  1.6861E+00,  1.7236E+00,  1.7614E+00,  1.7994E+00/
      DATA Y4BASE /  2.4651E+00,  2.5308E+00,  2.5894E+00,  2.6408E+00,
     *  2.6853E+00,  2.7228E+00,  2.7533E+00,  2.7770E+00,  2.7938E+00,
     *  2.8039E+00,  2.8072E+00,  2.8039E+00,  2.7939E+00/
      DATA A12MAT /  7.3487E-02,  7.6874E-02,  8.2183E-02,  8.9115E-02,
     *  9.7363E-02,  1.0663E-01,  1.1661E-01,  1.2700E-01,  1.3749E-01,
     *  1.4780E-01,  1.5760E-01,  1.6661E-01,  1.7451E-01/
      DATA A22MAT / -6.9680E-03, -7.4410E-03, -7.9752E-03, -8.5661E-03,
     * -9.2089E-03, -9.8990E-03, -1.0632E-02, -1.1403E-02, -1.2207E-02,
     * -1.3039E-02, -1.3896E-02, -1.4772E-02, -1.5663E-02/
      DATA B11MAT /  3.8091E+00,  3.7500E+00,  3.7253E+00,  3.7281E+00,
     *  3.7517E+00,  3.7892E+00,  3.8339E+00,  3.8791E+00,  3.9178E+00,
     *  3.9434E+00,  3.9491E+00,  3.9280E+00,  3.8735E+00/
      DATA B12MAT / -2.6554E+04, -2.9524E+04, -3.2929E+04, -3.6797E+04,
     * -4.1157E+04, -4.6038E+04, -5.1468E+04, -5.7476E+04, -6.4091E+04,
     * -7.1340E+04, -7.1340E+04, -7.1340E+04, -7.1340E+04/
      DATA B13MAT /  4.8123E+01,  4.5840E+01,  4.3773E+01,  4.1980E+01,
     *  4.0517E+01,  3.9442E+01,  3.8811E+01,  3.8680E+01,  3.9108E+01,
     *  4.0150E+01,  4.1864E+01,  4.4307E+01,  4.7535E+01/
      DATA B14MAT / -2.7172E+02, -2.7282E+02, -2.7750E+02, -2.8461E+02,
```