



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.

SAE Executive Standards Committee Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2022 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

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
http://www.sae.org

SAE WEB ADDRESS:

Full document details: SAE AIR4548B

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

Documents

Document Sharing Hub

TABLE OF CONTENTS

1.	SCOPE.....	4
1.1	Purpose.....	4
1.2	Overview.....	4
2.	REFERENCES.....	4
2.1	Applicable Documents.....	4
2.1.1	SAE Publications.....	4
2.2	Definition of Symbols.....	4
2.3	Definitions.....	6
3.	REAL-TIME MODEL APPLICATIONS.....	9
3.1	Control Development Benches.....	9
3.1.1	Objectives for Control Benches.....	9
3.1.2	Requirements for Control Benches.....	10
3.2	Integrated Flight/Propulsion Control Evaluation Tools.....	11
3.2.1	Objectives for Engine Models in Flight/Propulsion Control Evaluation.....	11
3.2.2	Requirements for Engine Models in Flight/Propulsion Control Evaluation.....	11
3.3	Embedded Software for Flight Systems.....	11
3.3.1	Objectives for Embedded Engine Models.....	12
3.3.2	Requirements for Embedded Engine Models.....	12
3.4	System Model Within Engine Control.....	12
3.4.1	Objectives for System Models Within Controls.....	12
3.4.2	Requirements for System Models Within Controls.....	13
3.5	Engine/Control Models in Flight Simulators.....	14
3.5.1	Objectives for Propulsion Models in Simulators.....	14
3.5.2	Requirements for Propulsion Models in Simulators.....	16
4.	REAL-TIME ENGINE MODELING METHODOLOGIES.....	17
4.1	General Model Characteristics.....	17
4.1.1	Requirements Specification.....	17
4.1.2	Computational Requirements.....	17
4.1.3	Input/Output.....	18
4.1.4	Size Requirements.....	18
4.2	Tradeoff Between Speed, Accuracy, and Cost.....	18
4.2.1	Speed.....	18
4.2.2	Consistency and Bandwidth.....	19
4.2.3	Cost.....	19
4.2.4	Accuracy Versus Cost.....	19
4.2.5	Accuracy Versus Speed.....	20
4.2.6	Cost Versus Speed.....	20
4.2.7	Risk Factors.....	20
4.3	Aerothermodynamic Models.....	20
4.3.1	Mathematical Model Approach.....	20
4.3.2	Usage.....	22
4.3.3	Characteristics.....	22
4.3.4	Example Case.....	26
4.4	Piece-Wise Linear Models.....	26
4.4.1	Mathematical Model Approach.....	26
4.4.2	Usage.....	27
4.4.3	Characteristics.....	28
4.4.4	Example Case.....	29
4.5	Transfer Function Models.....	29
4.5.1	Mathematical Model Approach.....	30
4.5.2	Usage.....	31
4.5.3	Characteristics.....	32
4.5.4	Example Case.....	33
4.6	Selecting a Modeling Methodology for a Specific Application.....	33
4.6.1	Performance Measurements Used to Select Modeling Methodology.....	34

4.6.2	Real-Time Model Application Requirements.....	34
4.6.3	Modeling Methodology Performance Characteristics	34
4.6.4	Selecting Specific Modeling Methodology for a Specific Application.....	35
5.	INTEGRATION OF REAL-TIME ENGINE MODELS WITH OTHER SYSTEMS.....	35
6.	UTILITIES	36
6.1	Iteration Methods (Iterative Solvers).....	36
6.1.1	Bisection.....	36
6.1.2	Successive Substitutions	36
6.1.3	Newton Methods	36
6.1.4	Multivariable Iterations	37
6.2	Integration Methods	37
6.2.1	Explicit Integration.....	37
6.2.2	Implicit Integration	37
6.2.3	Trapezoidal Integration	38
6.3	Lead	38
6.4	Lag	39
6.5	Curve Data	39
6.5.1	Table Lookup	39
6.5.2	Polynomials.....	39
6.5.3	Splines	39
7.	NOTES.....	40
7.1	Revision Indicator.....	40
APPENDIX A	EXAMPLE OF AEROTHERMODYNAMIC REAL-TIME MODEL FOREWORD.....	41
APPENDIX B	EXAMPLE OF PIECEWISE LINEAR REAL-TIME MODEL FOREWORD	61
APPENDIX C	EXAMPLE OF TRANSFER FUNCTION REAL-TIME MODEL FOREWORD	69
Table 1	Real-time model performance requirements	34
Table 2	Modeling methodology performance characteristics	34

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., $G_i = G_i * (1 + \sum C_{i,j} Y_j)$. 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 (HMC) assembly. For example, if the HMC 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.

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

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).

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

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).

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

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).

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

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 $\pm 2\%$ 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 $\pm 5\%$ 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.