Time-Accurate Turbine Engine Simulation in a Parallel Computing Environment
Part 1 - Software Acceptance Test

by

M. A. Chappell,
D. J. Malloy, and
B. K. Feather
SverdrupTechnology, Inc., AEDC Group
Arnold Engineering Development Center
Arnold Air Force Base, Tennessee

For presentation at the
Second Annual ITEA High Performance Computing Workshop
Edgewood, MD
June 28 - July 1, 1999

Time-Accurate Turbine Engine Simulation in a Parallel Computing Environment Part I - Software Acceptance Test*

M. A. Chappell, D. J. Malloy, and B. K. Feather
Sverdrup Technology, Inc., AEDC Group
Arnold Engineering Development Center
Arnold Air Force Base, Tennessee

Abstract

The ability to rapidly execute a high-fidelity turbine engine simulation is essential in automating data validation processes during developmental engine testing. Further, execution of a simulation in real time enables validation of the data stream concurrently with data acquisition.

Component-level turbine engine simulations provide high-fidelity, steady-state, and time-accurate engine performance computations but are not typically applied in a real-time environment. An approach for distributing a component-level simulation task in a parallel computing environment in an effort to achieve real-time operation is the subject of this paper.

A turbine engine simulation was restructured to operate in a parallel computing environment and tested to quantify the impact on simulation fidelity and execution time. A software acceptance test was designed to measure the scalability and integrity of the approach. The results from the software acceptance test confirmed the viability of the parallel simulation approach and guided the formulation of refinements for future improvements. The approach, results of the software acceptance test, and direction of future work are summarized herein.

Introduction

Turbine engine testing at the Arnold Engineering Development Center (AEDC) is conducted to evaluate engine operation at a wide variety of power conditions and simulated altitude conditions. Thousands of sensors, each producing measurements at rates in excess of 100 samples per second, are typically installed in the engine and test facility to measure aerothermodynamic performance. Consequently, a typical 8-hr test can produce 500 million samples of aerothermodynamic performance data. The challenge is to ensure the validity of the data, monitor the condition of the engine, and to identify anomalies promptly.

The countless variations of steady-state and transient engine operation and the necessity to delineate between sensor anomalies and abnormal engine deterioration, combined with the large volume of data, overwhelm the capabilities of traditional data validation methods. Although traditional methods produce meaningful results, they are labor-intensive and time-consuming. Consequently, application of the methods is typically restricted to a fraction of the available data, which diminishes the ability to detect anomalous data and intermittent events.

An approach was developed for a fast, comprehensive, and automated data valida-

* The research reported herein was performed by the Arnold Engineering Development Center (AEDC), Air Force Materiel Command. Work and analysis for this research were performed by personnel of Sverdrup Technology, Inc., AEDC Group, technical services contractor for AEDC. Further reproduction is authorized to satisfy needs of the U. S. Government.

tion process. The process consists of several analysis and modeling methods which together provide a real-time model-based data validation tool. The methods include an event detection system, a rule-based expert system with more than 150 checks, and an engine model-based fault detection and diagnostic system (Fig. 1). The engine model-based element of the system relies on a component-level turbine engine model (CLM) which employs fundamental physical laws to relate measurements to each other. The CLM is an industry-accepted method[1-3] for simulating the aero-thermodynamic performance of turbine engines, but it is not typically used when real-time simulation is required. However, the CLM provides the level of accuracy and detail required for data validation and fault diagnosis. Other turbine engine modeling methods are available but may compromise accuracy and detail to increase execution speed.[4] Furthermore, the CLM is more easily adapted to frequent engine changes that are expected during development cycles.

The CLM must process each data sample at speeds that match or exceed the data sampling rate since it is an element of an automated process that monitors a data stream whose measurements vary continuously. Otherwise, a backlog of data samples grows while waiting for earlier data samples

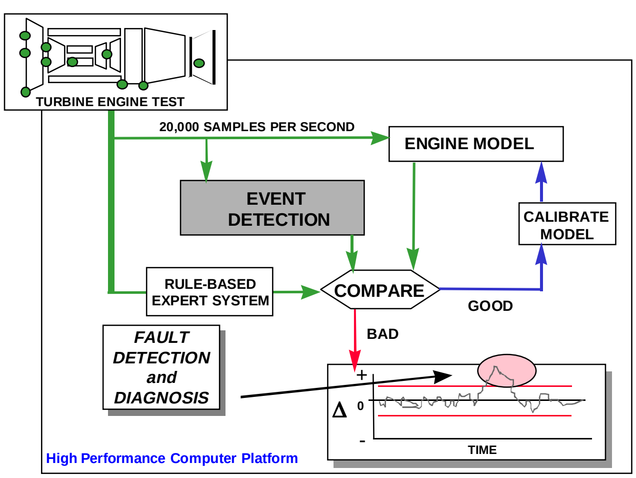

Fig. 1. Real-time model-based data validation system.

to be processed. A "real-time" processing rate is defined as a rate that matches the sampling rate. An additional requirement for a rate that exceeds the "real-time" rate addresses an offline need to process recorded data in a time span that is shorter than the duration of the test. For example, a 4-hr data tape, off-loaded from a test vehicle, can be processed in less than 4 hours.

The challenge is to implement a computing strategy that enables the CLM to execute in real time or less. Parallel computing was selected as a strategy to provide real-time execution. This strategy also can be scaled up, allowing extension to offline "faster than real-time" applications. A temporal decomposition of a time-accurate turbine engine CLM is proposed in which the CLM is replicated on multiple computer processors (CPU) within a high-performance multiple-CPU computer. Each CPU processes a set of time-synchronized data samples as other sets are processed on additional CPU's. Although spatial decomposition is effective for computational fluid dynamic (CFD) applications,[5] previous CLM work suggests that spatial decomposition of a CLM is ineffective.[6,7] Additionally, temporal decomposition of a CLM is more easily distributed to an increasing number of CPU's.

The objective of this work is to provide an automated, real-time, model-based test data validation computer code. The CLM is one element of an approach to permit comprehensive validation of test data. Each element, including the CLM, is subject to standard software development practices including Software Acceptance Testing (SAT), Alpha-testing, and Beta-testing prior to a production release.

The subject of this paper is the parallel implementation strategy for the time-accurate CLM. Results of the SAT and recommended refinements to the strategy are also discussed.

Component-Level Model Description

The component-level turbine engine model (CLM) provides a one-dimensional component-level time-accurate simulation applicable to arbitrary engine configurations. The CLM is capable of simulating off operating-line engine operation and utilizes widely accepted component-matching principles.[1-3] An engine control simulation may also be included as an additional program module.

The CLM combines physical relationships that govern engine operation with empirical relationships that describe individual component performance. The result is an adaptable model for which the effects of changes to engine attributes (e.g., components, configuration, and controls) are incorporated by making corresponding changes to the model attributes. Additionally, the component-matching approach quantifies the effects of engine changes, enabling a prediction capability for the data validation and fault diagnosis process.

The CLM is an assembly of components constrained to operate in unison to simulate the engine. An augmented turbofan engine, for example, may include a variable-geometry fan and compressor, combustor, high- and low-pressure turbines, fan bypass duct, mixer, augmentor, and variable exhaust nozzle (Fig. 2). The component models combine thermodynamic process equations with empirically determined component performance relationships to simulate component performance. An iterative technique is used to satisfy the implicit relationships that constrain the assembly to mass, momentum, and energy conservation principles. Measured engine control variables are used to govern model operation. The effects of rotor acceleration, heat transfer, and off-schedule variable geometry are included, providing a simulation of steady-state and transient engine operation ranging from engine starting conditions to maximum power. The CLM provides accurate simulation of operating temperatures, pressures, mass flows, and rotational speeds for each of the components illustrated in Fig. 2. Figure 3 illustrates comparisons between model calculations and test measurements for a turbine engine propulsion system.

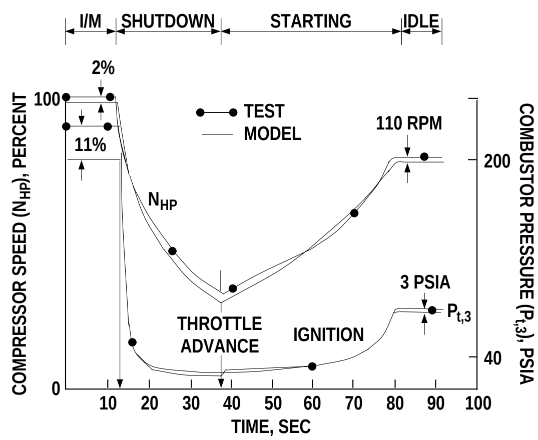

Fig. 3. Comparison of model calculations and test measurements for an augmented turbofan engine.

The primary time-dependent effect for the time-accurate CLM is the influence of rotor inertia on engine operation. Heat transfer

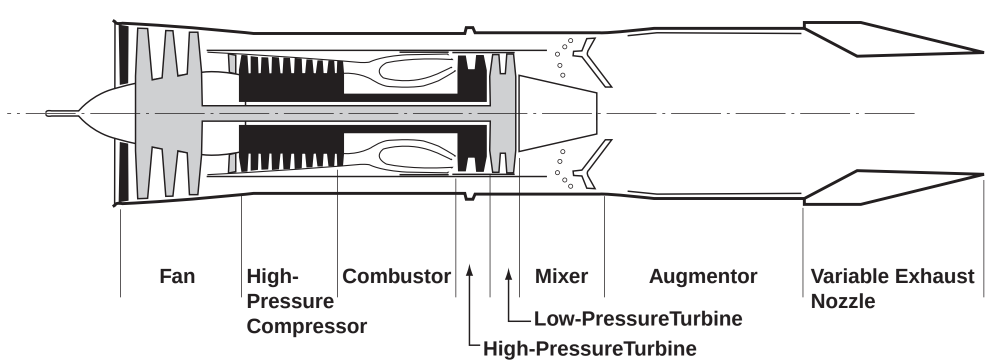

Fig. 2. Augmented turbofan engine.

between the gas path and the engine hardware is a significant but secondary effect. The physical effects of rotor inertia and heat transfer on engine operation are dependent on the time-related history of the affected parameters and, consequently, the simulation of these effects is dependent on previously computed values of the affected parameters. The dependence on previous values of the affected parameters imposes a serial character on the simulation computations. The serial character of the physical phenomena and their simulation, coupled with the serial character of numerical integration techniques employed by the CLM, present a challenge to the parallel computing strategy for the CLM.

CLM Operation in a Test Environment

The CLM, as used in a test environment, is implemented without an engine control simulation module. Avoiding a requirement for a control simulation module eliminates a potential source of error in the data validation process and simplifies the CLM. In the absence of a control simulation, selected test measurements are used as inputs to the CLM to prescribe its forcing functions.

Typically, measurements of the controlled variables such as fuel flow are used as inputs along with measurements of the environmental parameters such as inlet pressure and temperature. Figure 4 illustrates a typical set of pressure, temperature, speed, flow, horsepower extraction, and variable geometry measurements used as model inputs. Rotor acceleration computed from a time-history of rotor speed measurements is also input to the model together with the speed measurement.

Random temporal fluctuations in signals for a particular measurement are normal. Although normal, these variations are detrimental to the numerical stability of the model since the model is not designed to respond to rapid (>20 Hz) variations of operating conditions. Unstable model operation reduces fault detection and diagnosis effectiveness and leads to lengthy execution times; therefore, the measurements are filtered to minimize measurement noise while preserving the mean value of the measurement. Typical variations among fan speed measurements are shown in Fig. 5, along with filtered values used as input to the model.

The CLM uses time derivatives computed from the data for evaluation of transient effects such as rotor inertia dynamics and transient lag in the aerodynamic temperature measurements. The instantaneous rotor

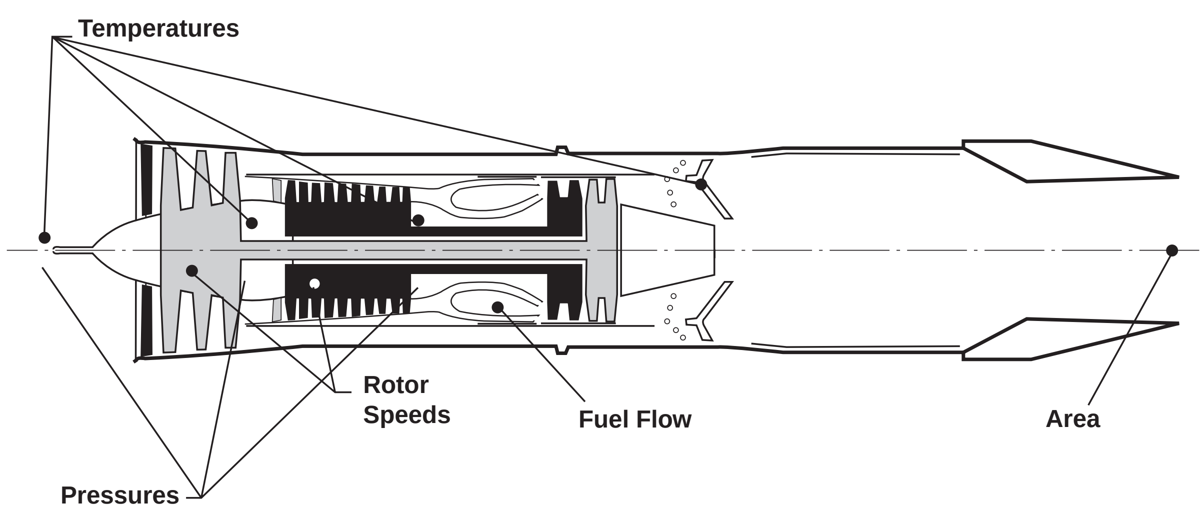

Fig. 4. Test measurements used as inputs to CLM.

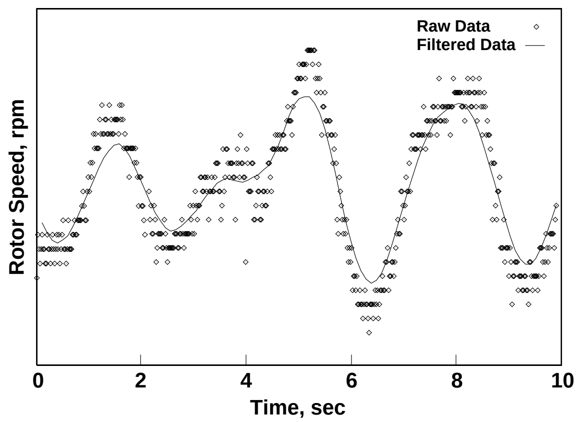

Fig. 5. Response of digital filtering techniques to rotor speed variations.

acceleration (time derivative of rotor speed), as determined by finite differences, is shown graphically in Fig. 6 as a function of time. The filtered acceleration, which is used as input to the model, is also shown.

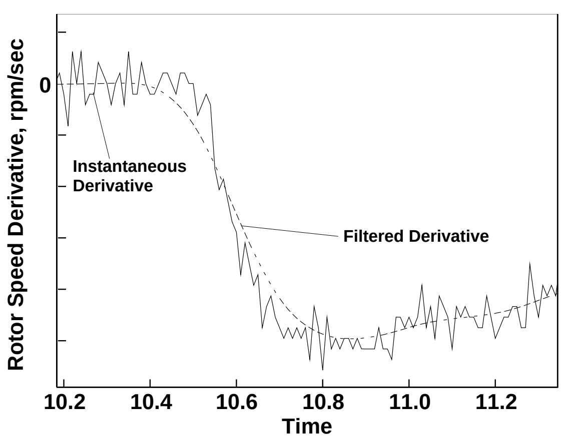

Fig. 6. Engine deceleration: rotor speed derivative as a function of time.

Parallel CLM Implementation

The time-decomposition of the time-accurate CLM enables a replicated worker approach for a parallel computing implementation of the CLM. The CLM code is replicated on each CPU of a parallel High Performance Computer (HPC). Although the results shown herein are from shared memory HPCs, the parallel implementation is also achievable on distributed architectures. All of the worker processors are controlled by a single master processor (Fig. 7). The master processor transmits a user-determined number of data samples to each worker, and the worker processes the samples with its own replication of the model. Consequently, the replicated worker must process noncontiguous data samples upon finishing one set of samples and starting the next set. Any impact of processing noncontiguous samples with the model must be eliminated or minimized for the time-decomposition replicated worker approach to succeed.

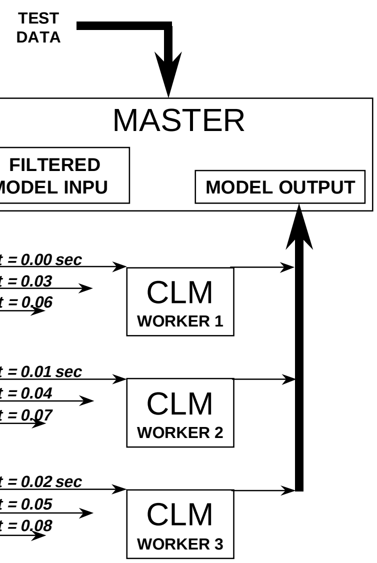

Fig. 7. Replicated-worker approach for time-decomposition of CLM (3 replicated workers).

Simulation of the time-dependent effects of rotor inertia and heat transfer on engine operation were addressed in two distinct approaches. The numerical integration of rotor acceleration, which requires contiguous samples in a conventional integration algorithm and provides the model's value of rotor speed, was replaced by the rotor speed measurement as described in the previous section. In contrast, the heat-transfer effects cannot be determined from the time-history of a single measurement because of a lack of related measurements.

Since the effects of heat transfer are secondary to the effects of rotor inertia, minor inaccuracies are tolerable. Therefore, an experiment was conducted to measure the impact of noncontiguous samples on heat-transfer modeling and to measure the effectiveness of modeling the history between noncontiguous samples. The results, shown in Fig. 8 for a lumped heat capacity model with an arbitrary time constant of 1.0 sec, indicate that the effects of noncontiguous processing increase as more CPUs are added since the addition of CPUs widens the gap between noncontiguous samples. Consequently, the achievable accuracy is inversely proportional to the number of CPUs. Additional research indicated these effects can be reduced by assuming a linear variation between noncontiguous samples, as seen in Fig. 9.

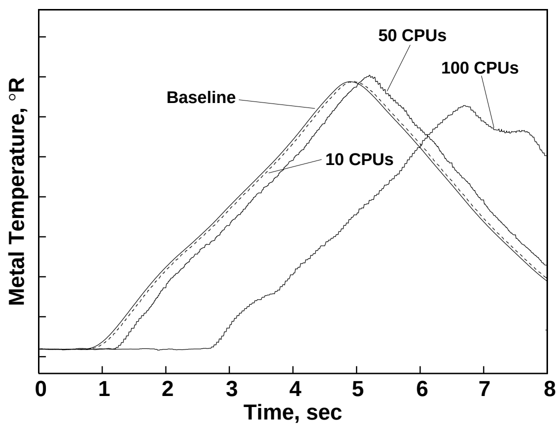

Fig. 8. Reduced accuracy of heat-transfer model with additional CPU's (stepwise variation between noncontiguous samples).

The approaches for including the effects of rotor acceleration and for stepwise variations (i.e., discontinuous variation between noncontiguous samples) in heat-transfer modeling were used during Software Acceptance Testing of the CLM.

Software Acceptance Testing

Software Acceptance Testing (SAT) consisted of two assessments: accuracy verification on multiple CPU configurations and execution speed measurements. Accuracy verification ensures that the results obtained

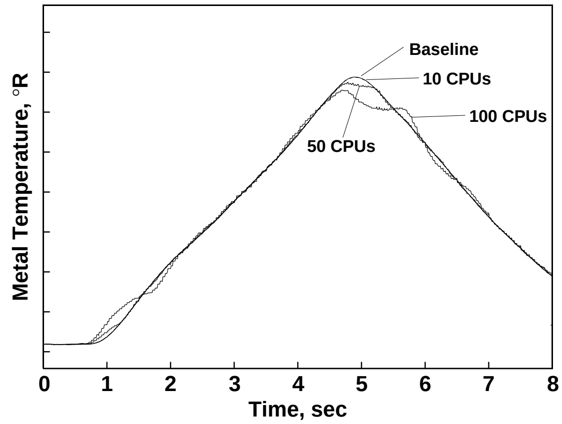

Fig. 9. Increased accuracy of heat-transfer model with additional CPU's (linear variation between noncontiguous samples).

from a variety of parallel configurations are the same, including a serial configuration and an increasing number of processors in parallel. The execution speed measurements provide an indication of the effectiveness of adding additional CPUs to the process to reduce execution time of the CLM.

Replication of results within a tolerance was required for a variety of multiple-CPU configurations. Differences in the results between the configurations were computed and analyzed. Acceptable tolerances were established for individual parameters and were based on the importance of the parameter in the fault detection process, the particular engine maneuver, and the precision of the computer. Typical comparisons between serial execution, which was considered the baseline, and multi-CPU configurations are shown in Figs. 10-13. The differences among the various configurations increase as the number of CPUs increases. This trend, if uncorrected, imposes a practical limit on the number of CPUs that should be employed. An approach to reduce the limitation in preparation for Alpha-testing is discussed in the next section.

Figures 10 and 11 show the differences in turbine exit temperature between a serial

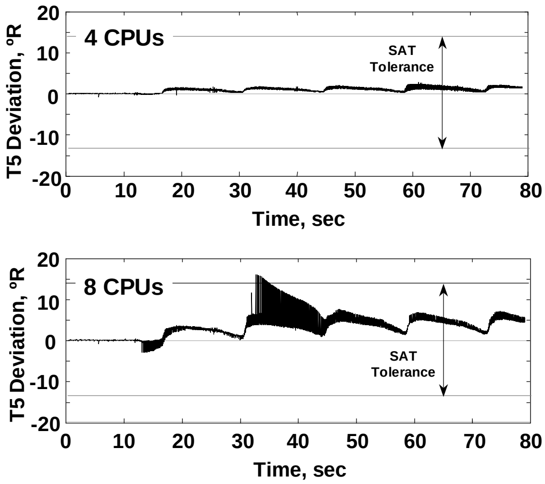

Fig. 10. SAT turbine exit temperature deviations with additional CPU's.

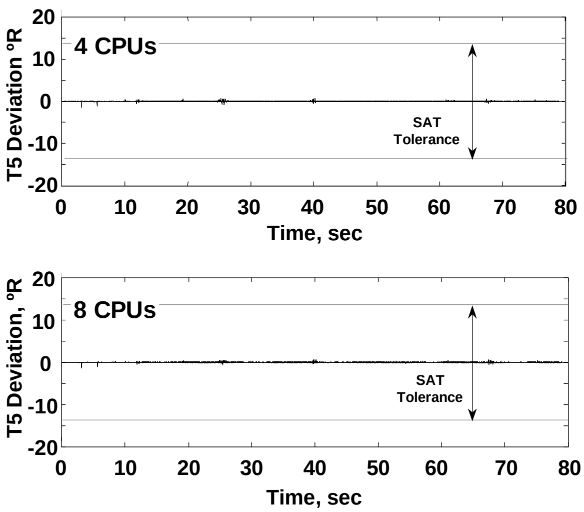

Fig. 11. SAT turbine exit temperature deviations with additional CPU's (excluding heat-transfer effects).

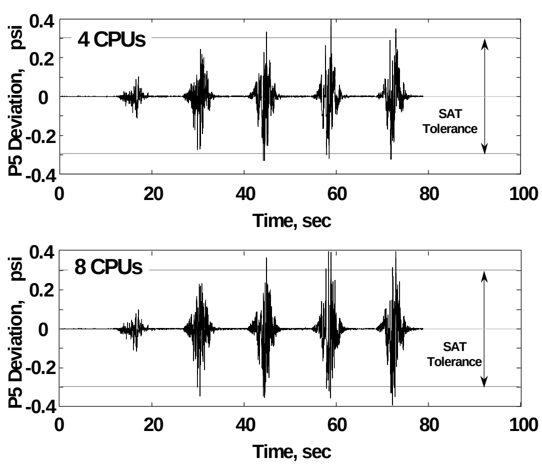

Fig. 12. SAT turbine exit pressure deviations with additional CPU's.

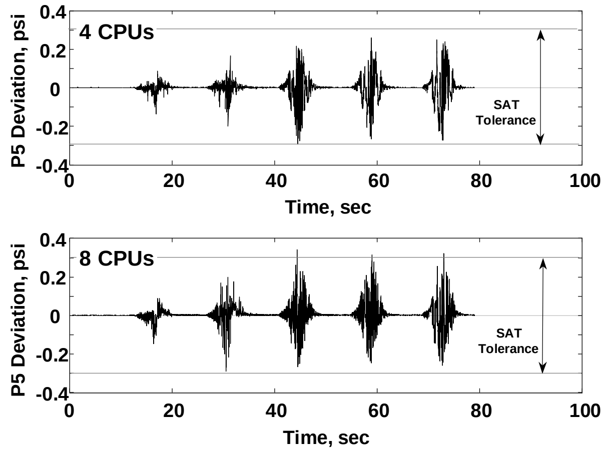

Fig. 13. SAT turbine exit temperature deviations with additional CPU's (excluding heat-transfer effects).

configuration and 4-CPU and 8-CPU configurations for an engine maneuver both with (Fig. 10) and without (Fig. 11) the effects of heat transfer. Turbine exit temperature (T5) was selected for illustration since rotor acceleration and heat transfer significantly affect it. The larger differences in Fig. 10 are attributed to the effects of processing noncontiguous samples for the heat-transfer model. However, despite the larger differences, the deviations are generally within SAT tolerances. Proposed improvements to the parallel CLM to compensate for these effects, along with variations in key performance parameters, will be discussed in the next section.

Figures 12 and 13 show the differences in turbine exit pressure between a serial configuration and 4-CPU and 8-CPU configurations for an engine maneuver both with (Fig. 12) and without (Fig. 13) the effects of heat transfer. Turbine exit pressure (P56) was selected for illustration since it is sensitive to model convergence tolerances. The slightly larger differences in Fig. 12 are attributed to the effects of processing noncontiguous samples for the heat-transfer model. However, despite the larger differences, the deviations are generally within SAT tolerances. Proposed improvements to the parallel CLM, including adjustments to the model convergence tolerances to compensate for these effects, will be discussed in the next section.

Execution speed was also measured during SAT of the CLM. Twelve distinct engine maneuvers were used during the test. Measurements were taken for serial execution of the CLM and 2-, 4-, 8-, and 12-CPU parallel configurations on two different HPCs, HPC-1 and HPC-2. Representative results are shown in Figs. 14-15 (HPC-1) and Figs. 16-17 (HPC-2). Figures 14 and 16 illustrate the speed increase factor relative to the serial execution time (i.e., speedup) achieved by employing additional CPUs. Figures 15 and 17 illustrate the speed increase factor relative to the duration of the engine maneuver as an indication of "real-time" capability. Variation in speedup among different engine maneuvers results from the impact of the maneuver on the number of iterations required for model convergence, i.e., severe maneuvers require more iterations.

A significant variation in the measurement of speedup was noted during processing on the HPC machines, especially for cases with 12 CPUs. The variation is apparently due to random blocking of program execution caused by memory and processor contention with other jobs in a time-sharing environment on the high-performance computing resources. The results shown in Figs. 14-17 are considered “worst case.” However, despite the contention, the assessment demonstrated an acceptable scalability character and the ability to achieve execution speeds that enable real-time CLM operation. Proposed improvements to the parallel CLM are discussed in the following section.

Proposed Improvements to Parallel CL

Three areas of improvement were identified as a result of SAT. The first relates to the impact on heat-transfer modeling of processing noncontiguous data samples. The second addresses model convergence tolerances and the associated number of iterations required for model convergence. The third improvement area addresses the situation in which the CLM is operated indepen-

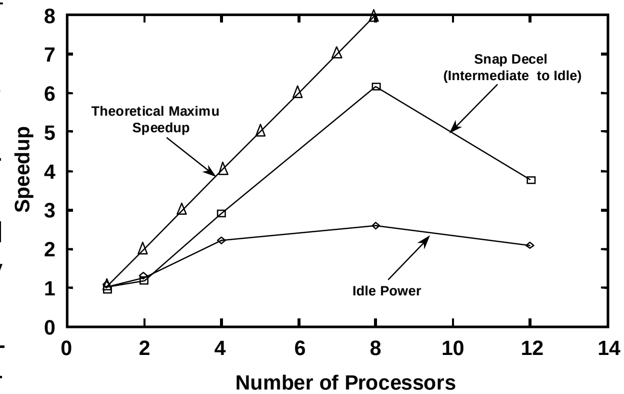

Fig. 14. Parallel processing effectiveness (HPC-1).

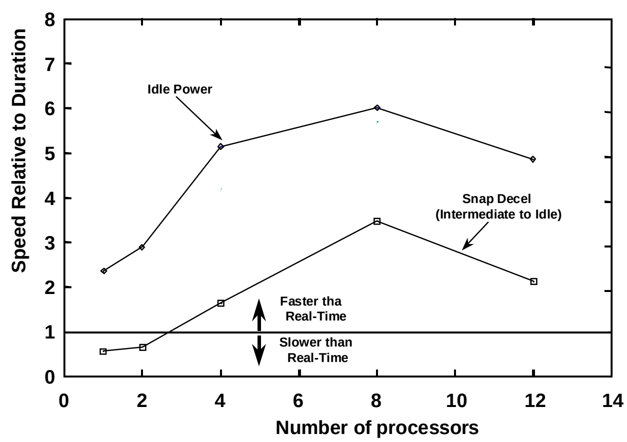

Fig. 15. Speed increase factor relative to duration of engine maneuver (HPC-1).

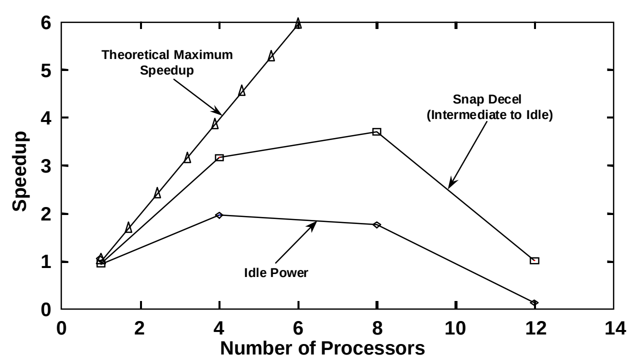

Fig. 16. Parallel processing effectiveness (HPC-2).

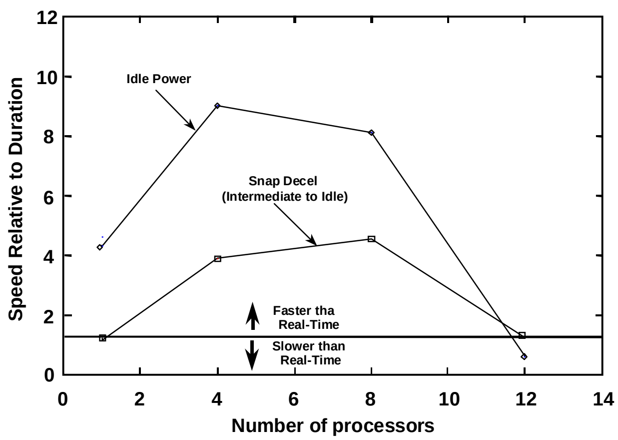

Fig. 17. Speed increase factor relative to duration of engine maneuver (HPC-2).

dently from the test environment and must rely on the integration of rotor acceleration to compute rotor speeds.

The impact of step-wise variations on the heat-transfer model that occur as a result of processing noncontiguous samples becomes more severe as more CPUs are applied to model operation. Research conducted during SAT indicated that introducing a simple linear model of the variations between the noncontiguous samples can reduce the effect. More sophisticated prediction algorithms would presumably provide additional reduction of the impact of processing noncontiguous samples, which eliminates the practical limit on the number of CPUs that can be applied effectively. A simple linear model will be tested during upcoming CLM development efforts.

The impact of tighter model convergence tolerances was noted during parallel model operation. Research conducted during SAT indicated that tighter convergence tolerances are required to reduce variations in the modeled parameters. A wider variation of model convergence tolerances will be evaluated during upcoming CLM development efforts, and scalable performance improvement and variations in computational time will be assessed.

Operation of the CLM in a test environment in which the rotor speeds are measured avoids the need for numerical integration in the CLM. However, numerical integration cannot be avoided for independent operation of the model. A synchronized implicit iteration approach is proposed for a test data-independent parallel operation of the CLM. In this approach all replicated workers perform an iteration and exchange current values of rotor speeds and accelerations, repeating until all replicated workers are complete (Fig. 18). The synchronized iteration approach is expected to speed up overall convergence while imposing a small penalty for sporadic occasions in which a worker must continue to iterate after completion until all workers are complete. This approach will also be tested during upcoming refinement efforts for the parallel implementation of the CLM.

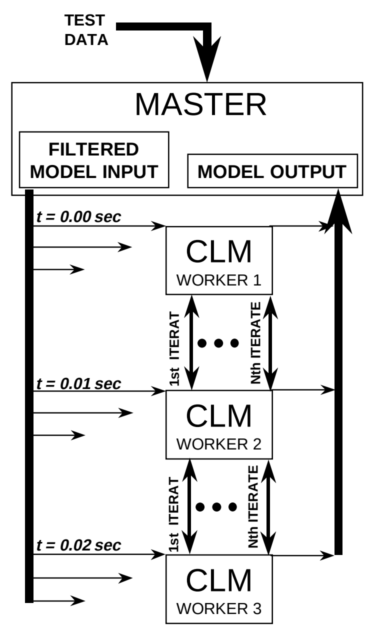

Fig. 18. Proposed synchronized iteration approach for CLM.

Summary

Component-level turbine engine simulations provide high-fidelity, time-accurate engine performance computations. An approach for distributing a component-level turbine engine simulation task in a parallel computing environment was implemented and subjected to Software Acceptance Testing (SAT).

Results of SAT indicated that the CLM can operate in real time and provide acceptable accuracy to serve as a critical element of a comprehensive, automated data validation process. SAT also revealed specific areas for improvement to both execution speed and accuracy for the parallel implementation of the component-level turbine engine model.

References

1. McKinney, J. S., "Simulation of Turbofan Engine, Part I. Description of Method and Balancing Technique," Air Force Aero-Propulsion Laboratory, AFAPC-TR-67-125-PT.-1, 1967.

2. Koenig, R. W. and Fishbach, L. H., "GENENG - A Program for Calculating Design and Off-Design Performance for Turbojet and Turbofan Engines," NASA TN D-6552, 1972.

3. Shapiro, S. R. and Caddy, M. J., "NEPCOMP - The Navy Engine Performance Program," American Society of Mechanical Engineers 74-GT-83, March 1974.

4. *Real-Time Modeling Methods for Gas Turbine Engine Performance*, Aerospace Information Report, AIR4548, Society of Automotive Engineers, Warrendale, PA, December 1995.

5. Steger, J. L., et al, "A Chimera Grid Scheme," American Society of Mechanical Engineers, Proceedings of the Applied Mechanics, Bioengineering, and Fluids Engineering Conference, Houston, TX, June 20-22, 1983.

6. *Parallel Processor Engine Model Program*, NASA-CR-174641, January 1984

7. Biegl, C., et al, "Automated, Real-Time Validation of Turbine Engine Test Data Using Explicit Parallelization on an Eight-Processor Pentium Platform," XIII International Symposium on Air-Breathing Engines, ISABE 97-7143, September 1997.