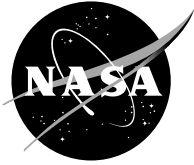


NASA/TM—2003-212543



Development of a Turbofan Engine Simulation in a Graphical Simulation Environment

Khary I. Parker and Ten-Heui Guo
Glenn Research Center, Cleveland, Ohio

August 2003

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

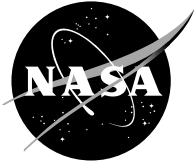
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA Access Help Desk at 301-621-0134
- Telephone the NASA Access Help Desk at 301-621-0390
- Write to:
NASA Access Help Desk
NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

NASA/TM—2003-212543



Development of a Turbofan Engine Simulation in a Graphical Simulation Environment

Khary I. Parker and Ten-Heui Guo
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center

August 2003

Acknowledgments

The authors would like to thank Kevin J. Melcher of NASA Glenn Research Center for his development of the model comparison script used in the validation process. The authors would also like to thank the following people from NASA Glenn Research Center for their helpful input in the preparation of this paper:
Dr. Sanjay Garg, George Kopasakis, Jonathan Litt, and Donald L. Simon.

Trade names or manufacturers' names are used in this report for identification only. This usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at <http://gltrs.grc.nasa.gov>

Development of a Turbofan Engine Simulation in a Graphical Simulation Environment

Khary I. Parker and Ten-Heui Guo
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Abstract

This paper presents the development of a generic component level model of a turbofan engine simulation with a digital controller, in an advanced graphical simulation environment. The goal of this effort is to develop and demonstrate a flexible simulation platform for future research in propulsion system control and diagnostic technology. A previously validated FORTRAN-based model of a modern, high-performance, military-type turbofan engine is being used to validate the platform development. The implementation process required the development of various innovative procedures, which are discussed in the paper. Open-loop and closed-loop comparisons are made between the two simulations. Future enhancements that are to be made to the modular engine simulation are summarized.

Nomenclature

<i>a14</i> :	Forward VABI area actuator (in ²)
<i>a16</i> :	Aft VABI area actuator (in ²)
<i>a8</i> :	Nozzle throat area actuator (in ²)
<i>a9</i> :	Nozzle exit area actuator (in ²)
AB:	Afterburner
<i>bpr14g</i> :	Forward duct bypass ratio
CAD:	Controller and Actuator Dynamics
CLM:	Component Level Model
HPC:	High-Pressure Compressor
HPT:	High-Pressure Turbine
IGV:	Inlet Guide Vane
LPT:	Low-Pressure Turbine
MAPSS:	Modular Aero-Propulsion System Simulation
<i>p2</i> :	Fan inlet pressure (psia)
<i>p27</i> :	HPC pressure (psia)

PI:	Proportional-plus-Integral
PLA:	Power Lever Angle (degree)
<i>pnrm2</i> :	Percent corrected fan spool speed
<i>ps3</i> :	HPC exit static pressure (psia)
<i>ps15</i> :	Bypass duct static pressure at mixer (psia)
<i>ps56</i> :	LPT exit static pressure (psia)
<i>sm2</i> :	Fan Stall Margin
<i>stp2</i> :	Fan variable inlet stator vane actuator
<i>stp27</i> :	HPC stator vane actuator
<i>stp27d</i> :	Booster tip stator vane actuator
<i>t2</i> :	Fan inlet temperature (°R)
<i>t27</i> :	HPC temperature (°R)
<i>t3</i> :	HPC exit temperature (°R)
<i>t56</i> :	LPT exit temperature (°R)
<i>t5b</i> :	LPT blade temperature (°R)
<i>tmpe</i> :	Average core metal temperature aft of burner.
VABI:	Variable Area Bypass Injector.
<i>w14</i> :	Forward duct flow rate (slugs/s)
<i>w2</i> :	Fan exit flow rate (slugs/s)
<i>wf36</i> :	Burner fuel flow actuator (pph)
<i>wf6</i> :	Afterburner fuel flow actuator (pph)
<i>xnh</i> :	High-pressure spool speed (RPM)
<i>xnl</i> :	Low-pressure spool speed (RPM)

Introduction

In recent years, interest in intelligent engine technology with advanced active component control has increased. This interest is being pursued in areas such as life extending control, which reduces the effect of engine component damage and extends engine life while maintaining optimal component performance [1]; and model-based control, which integrates a

real-time engine model into the control architecture [2]. The computer engine models used in these and other types of control studies are component level models (CLM), a model that links individual component models of state space and non-linear algebraic equations, written in a computer language such as FORTRAN. The difficulty faced with performing control studies on FORTRAN-based models is that the FORTRAN platform is not supported with control design and analysis tools, and there is no means to do real-time control implementation. It is desirable to have a simulation environment that is straightforward, has modular graphical components and allows easy access to health, control, and engine parameters through a graphical user interface. Such a tool should also provide the ability to convert a control design into real-time code. Because of the current technology in advanced modeling software, such as MATLAB[®] [3] and Simulink[®] [4], such an engine model, represented in a graphical simulation environment, has the capability to become an extremely powerful tool in control and diagnostic system development.

A generic turbofan engine simulation is being implemented as a modular aero-propulsion system simulation (MAPSS) using Simulink[®], a computer aided control design and simulation package that allows graphical representation of dynamical systems in a block diagram form. The goal of this effort is to develop and demonstrate a flexible turbofan engine simulation platform. This platform will allow easy access to health, engine, and control parameters, as well as a quick way to test control and diagnostic systems.

This paper focuses on the development of the MAPSS environment and its validation using a FORTRAN engine simulation. The engine model to be simulated is described first. Next, the paper discusses the approach and validation methods used for implementation, including example comparisons between the two simulations. Finally, the paper concludes with a summary of the validation results and future work to be done.

Engine Model Description

A previously validated FORTRAN-based model of a modern high-pressure ratio, dual-spool, low bypass, military-type engine, with a digital controller, is being implemented in MAPSS. This engine model (seen in Figure 1) is a nonlinear, low bandwidth, transient, performance model similar to the type used in various areas of intelligent engine control research such as model-based control [2] and nonlinear performance seeking control [5].

The FORTRAN simulation has an updating rate of 50 Hz, which is the rate needed by its controller. The engine model is represented as a CLM, along with accompanying maps that model component characteristics [5]. The components of this engine model consist of a single stage high-pressure ratio fan with variable inlet stator vanes, booster with independent hub and tip stator vanes, high-pressure mixed flow compressor, double-annular combustor, high- and low-pressure turbines, afterburner, and nozzle components. Also included are forward blocker doors and an aft variable area bypass injector (VABI), giving the engine variable cycle capability [5]. The CLM uses low-pressure and high-pressure rotor speeds, as well as average hot section temperature (i.e. metal temperature) as state variables. The integration of these state variable occurs within a `while` loop in order to achieve mass, energy, and momentum balance within a certain tolerance.

Along with a diagram of the implemented engine model, Figure 1 also lists the actuators and sensor variables used by the controller. The actuators, listed above the engine diagram, are for the fan variable inlet stator vanes (*stp2*), forward blocker door area positioning (*a14*), booster tip stator vanes (*stp27d*), high-pressure compressor and booster hub stators vanes (*stp27*), burner fuel flow (*wf36*), aft VABI area positioning (*a16*), afterburner fuel flow (*wf6*), and nozzle throat and exit area positioning (*a8* and *a9*, respectively). There are many sensors placed throughout the engine, as shown in Figure 1 below engine diagram. The sensors used by the controller in this model measure fan inlet temperature (*t2*) and pressure (*p2*),

low-pressure spool speed (*xnl*), high-pressure compressor inlet temperature (*t27*) and pressure (*p27*), high-pressure spool speed (*xnh*), high-pressure compressor exit temperature (*t3*) and static pressure (*ps3*), low-pressure turbine blade temperature (*t5b*), bypass duct static pressure at the mixing plane (*ps15*), low-pressure turbine exit temperature (*t56*) and static pressure (*ps56*).

This FORTRAN simulation also has a digital controller model which runs at a fixed frame rate of 50 Hz for sampling sensor inputs and issuing actuator commands. This controller is represented in Figure 2. Inputs used by the controller are power lever angle (PLA), Mach number, altitude, and sensor outputs from the CLM. The controller uses an open loop scheduler to control the stator vane actuators. Proportional-plus-integral (PI) control is used to regulate actuator demands for the burner fuel flow (*wf36*), nozzle exit area (*a8*), and aft VABI exit area (*a16*) through five modes: high, medium, and low PLA power settings, fan stall margin (*sm2*) and overspeed mode. Each mode has its own set of gains for PI control. The PLA is also used in control logic to position *a14*. The actuator demands set the corresponding actuator positions used by the CLM (see Figure 3). The CLM then feeds the controller calculated and measured sensor values in order to re-compute actuator demands.

The goal of this paper is to explain the process used to implement these functions and features in the MAPSS environment, and thereby obtain a new, powerful development tool for control and diagnostic systems. To explain this process, the next section gives a description of the MAPSS environment, followed by a discussion on the implementation approach and validation procedures that were employed to develop MAPSS.

MAPSS Description

The Modular Aero-Propulsion System Simulation (MAPSS) is a nonlinear, non-real time system composed of the “Controller and Actuator Dynamics” (CAD) and “Component Level Model” (CLM) modules. MAPSS is

capable of aiding in the same areas of research as the FORTRAN model with increased flexibility for control systems development, as discussed before.

The controller in the CAD module emulates the functionality of a digital controller, which has a typical update rate of 50 Hz. The CLM module simulates the dynamics of the engine components and uses an update rate of 2500 Hz, which is needed to iterate for solutions for mass-energy balancing among system components. The actuators in the CAD module use the same sampling rate as the CLM. The method used to obtain the mass-energy balance will be discussed in the next section.

The graphical user interface (GUI) accompanying MAPSS is designed to simplify the user’s interaction with the model. Figure 3 shows the interaction of the GUI with the CAD and CLM modules. The main GUI panel (Figure 4) manages the input/output information from the simulation. The GUI is used to input simulation parameters, such as PLA, Mach number, and altitude, and output parameters from both the controller and CLM. In addition, a sub-GUI allows changes to controller, engine, and health parameter constants (see Figure 5), as well as other parameters that may be varied. After running the model, a summary of simulation output operating conditions is sent to a display in the main GUI. Output that is created by the model, along with user defined input information is stored in an output structure array (a MATLAB[®] data type) that may be saved as a MAT-file and reloaded into the GUI or accessed independently.

The components simulated in the MAPSS CLM (Figure 6) are the fan, booster, high-pressure compressor (HPC), burner, high- and low-pressure turbines (HPT and LPT), mixer, afterburner (AB), and nozzle. The fan inlet temperature and pressure are external inputs calculated from user-defined operating conditions. High- and low-pressure turbines are simulated to extract work from the hot section to provide power to the fan/compressor section. The CLM also models the forward blocker doors, the VABI, the flow through the bypass

duct, and the bleed flows. The bypass duct module determines the pressure, temperature, enthalpy, and flow rate up to the point where the bypass and core flows mix, known as the mixing plane. The CLM does not include an inlet model because that is typically not considered part of the engine. In addition to these parameters, the compressor bleeds and cooling flows are simulated. The three state variables used in the FORTRAN-based model—low-pressure rotor speed, high-pressure rotor speed, and the block metal temperature—are also used in MAPSS CLM. The overall engine component system requires iteration steps to ensure a balance of mass flows and energy.

The CAD module contains the controller and the actuator dynamics sub-modules. The controller sub-module (Figure 2) uses calculated and simulated sensed output values from the CLM to determine if the desired set points have been reached. The set points are based on the user defined operating conditions of PLA, altitude, and Mach number. The controller uses maps to schedule the actuator demands for the fan, booster, and HPC stator positions using the engine's operating conditions. To obtain the actuator demands for $wf36$, $a8$, $a16$, the controller has five different modes, each of which is a proportional-plus-integral (PI) control action regulator. The High power mode handles CLM operation for $PLA \geq 42.5^\circ$. The Medium power mode handles CLM operation for $PLA \in (32.5^\circ, 37.5^\circ]$. The Low power mode handles CLM operation for $PLA \leq 32.5^\circ$. As the controller transitions between the three power modes, percentages of each regulator output, based on the current value of PLA, fan stall margin, and xnl , are added together to obtain the final actuator demand. This is a technique known as blending. In addition to the power modes there is the Fan Stall Margin mode controls the fan in the stall region, and the Overspeed mode limits the speed of xnl . These last two modes may be turned on and off upon request.

The actuator dynamics sub-module (Figure 7) simulates the dynamics of the torque motors, servomechanisms and feedback to determine actuator positions for the fan, HPC, booster,

guide vanes; nozzle, mixer, and VABI areas; and the fuel flow actuator. Lags are used to simulate actuator dead-time or phase shift. These lags are also captured in MAPSS, even though their precise nature has not been explored.

Implementation Approach and Validation

Before development of this environment commenced, a fundamental understanding of the FORTRAN engine component models, and the information passed between them was obtained. This knowledge was used in implementing the engine simulation in a block diagram format. A script written in MATLAB[®], an object-oriented software language used for algorithm development, modeling, and data analysis, was used to take data generated by the FORTRAN model, and compare it to data generated by the block diagrams. The block diagram depicting the engine components in MAPSS can be seen in Figure 8. This section explains the development approach and validation of MAPSS then shows example comparisons between it and the FORTRAN model.

As seen in Figure 9, the values that are used to determine the fan exit parameters are calculated in the intermediate subsystem block diagrams. The subsystems contain algebraic equations and maps, which are used to characterize the behavior of the fan component. For instance, the “IGV Scalars” subsystem uses maps to determine how the fan inlet guide vane setting affects the output of the fan, and the “Corr. Airflow w/ Re # Scalars” uses a combination of maps and algebraic equations to determine the corrected flow and Reynolds number. Generally, in developing the subsystems within each component model, special attention was paid to highlighting the flow of information. The general organization of the subsystems in this and other component models follows the organization of equations in the FORTRAN code. However, to simplify the illustration of the information flow, the order of execution of some of the equations was changed. This would not have an affect on the results produced by each component.

The validation process consisted of two steps. First, there was an open-loop comparison of each block diagram component with the FORTRAN model, then a comparison between the MAPSS and FORTRAN closed-loop models. Each completed block diagram, for both intermediate blocks and component blocks, was validated before it was linked to another subsystem. The MATLAB[®] script that was used for model comparison supplied FORTRAN input variables from the subroutines to the block diagram and produced comparison plots for each output variable. The plots displayed raw data from the FORTRAN model and block diagram, as well as the percent error between the two, which was used to determine the validity of the block diagrams. Since the state space equations of the FORTRAN routine were exactly represented with proper integration, comparison utilizing frequency response was not deemed necessary and the time-domain error criterion was sufficient means of comparison.

As an example of the comparison plots, Figure 10 shows the response of $pnrm2$ and $w2$ for both models. The operating conditions for this test consisted of a step increase in PLA from 21 to 30, at sea-level, with a Mach number of 0. From these figures, it can be seen that the largest difference occurs during the transition from 21 to 30 (i.e., the transient region). There is very little difference (less than a tenth of a percent) in both steady state regions. Generally, if the error was less than one percent the block diagram was considered valid. However, if a block diagram had an error that was higher than one percent, it was examined to determine the reason for the difference. During validation, it was acceptable to have an error around one percent during the transient, just as long as the steady state error before and after was less than one percent. According to the plots shown in Figure 10, there is good matching of the fan component, therefore, it is considered a valid block. These steps of block diagram building and validation were conducted repeatedly until the CLM and CAD modules were fully modeled.

The CLM and CAD modules were integrated (i.e. connected) to form a closed-loop model. In order to properly integrate these two modules,

the CLM had to be balanced in mass and energy. A challenge encountered during implementation was developing a scheme that would present a balanced model to the CAD module. In the FORTRAN simulation, as stated before, a while loop is used to ensure the selected mass-flow, energy, and work of the turbines is balanced out before the next simulation time step of 0.02 seconds. For MAPSS, an equivalent mechanism would be to feed these parameters back into the CLM (i.e. create an algebraic loop). However, an algebraic loop can prevent the MAPSS CLM from finding a convergent solution without the parameters being updated. A solution to this problem was to construct an iteration block that adjusts a vector of parameters, obtained from the CLM, in order to drive a norm of component error values to zero, and then delay the iteration of the block by one simulation time step (40 milliseconds). This allowed the components to achieve an energy balance within the same tolerances as the FORTRAN components, meaning that MAPSS CLM and the FORTRAN CLM are running at effectively the same sample rate. The adjustable vector consists of mass-flow, energy, and work parameters that are used to update the CLM at each time step, but delayed by one time step to avoid an algebraic loop. The error values are derived from turbine losses, and booster and HPC mass flow rates, as well as bypass, mixer, and nozzle throat pressure. They are calculated as the difference between the estimated value, obtained from component maps, and their calculated values. To ensure the balanced condition is met before the next control step, the iteration block and the CLM, as stated before, operate at a higher frequency than the controller.

Figure 11 shows a comparison of state variables between the FORTRAN and MAPSS closed-loop simulations, for two PLA step up conditions. The first is a small step up increase from 21 to 22 degrees and the second is a larger step up increase, from 21 to 50 degrees. Mach number and altitude remained the same as in the fan component test. Figures 11(a) and 11(b) show that there is very little difference (less than one percent) between MAPSS and the FORTRAN simulation. The only caveat to this result is that, for both models, it was found that

the bypass ratio at the bypass duct entrance (known as *bpr14g*) became negative during the transient (see Figure 12). This result manifests itself for $PLA > 35$. This occurs because reverse flow is allowed to occur at the bypass duct entrance (known as *w14*) during the transient. Although, through testing this phenomenon does not seem to cause any adverse effects in the performance of MAPSS, it is undesirable to have reverse flow anywhere in the model because it may cause instabilities to occur among the neighboring components. Therefore, the measurements for *bpr14g* are now prevented from becoming negative in the MAPSS CLM (as shown in Figure 12), which causes the differences between the two simulations shown in Figure 13.

Summary of Results

This paper presents progress on the development of a generic turbofan engine simulation in Simulink[®]. The goal of this effort was to develop and demonstrate a flexible simulation platform for future research in propulsion system control and diagnostic technology. A FORTRAN-based model of a modern, high-performance, military-type turbofan engine was used as the test case for the platform development. The focus of this paper was on the development of the MAPSS environment and its validation using the FORTRAN-based engine model. The block diagrams in MAPSS were developed from the state space and non-linear algebraic equations in the FORTRAN model. The validation procedure compared the block diagram component models against the FORTRAN component models, using a MATLAB[®] script, and then compared the MAPSS and FORTRAN closed-loop models. It was found through open-loop and closed-loop comparison, for a variety of operating conditions, that there was less than one percent difference between the two simulations when looking at the individual components, the controller models, and the overall models.

There was, however, a change made in the MAPSS CLM to prevent the bypass ratio from becoming negative. Future improvements to MAPSS include improvements to the GUI and the implementation of a commercial, high-bypass turbofan engine in the MAPSS environment.

References

1. Caplin, J., Ray, A, "Life Extending Control of Gas Turbine Engines for Aircraft Propulsion," 36th IEEE Conference on Control and Decision, San Diego, CA, Dec. 1997.
2. Adibhatla, S., Gastineau, Z., "Tracking Filter Selection and Control Mode Selection for Model Based Control," AIAA 94-3204, 30th Joint Propulsion Conference and Exhibit, Indianapolis, IN, June 1994.
3. MATLAB, "Getting Started with MATLAB," The MathWorks, Inc., 1996.
4. Simulink, "Using Simulink," The MathWorks, Inc., 2000.
5. Adibhatla, S., Johnson, K.L., "Evaluation of Nonlinear PSC Algorithm on a Variable Cycle Engine," AIAA 93-2077, 29th Joint Propulsion Conference and Exhibit, Monterey, CA, June 1993.
6. Watts, S.R., Garg, S., "A Comparison of Multivariable Control Design Techniques for a Turbofan Engine Control," ASME 95-GT-258, International Gas Turbine and Aeroengine Congress and Exposition, Houston, TX, June 1995.

APPENDIX A: Figures

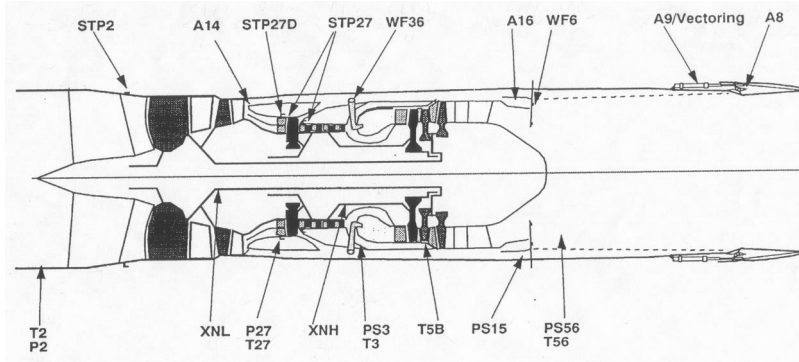


Figure 1: Schematic of turbofan engine model with labeled actuators (above) and sensors (below) [2].

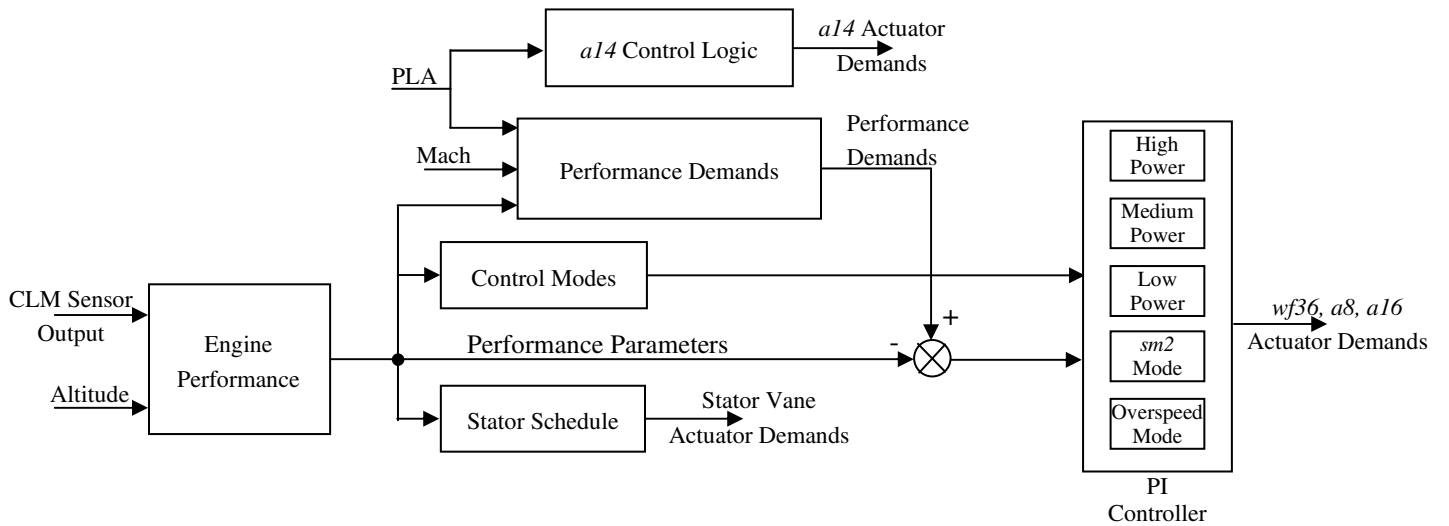


Figure 2: Representation of the Digital Controller Module.

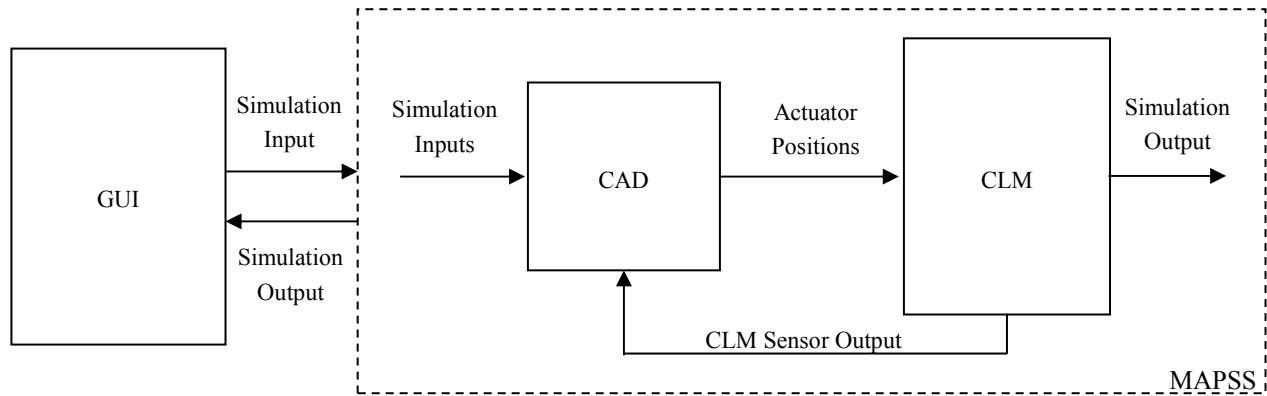


Figure 3: MAPSS Module Interaction.

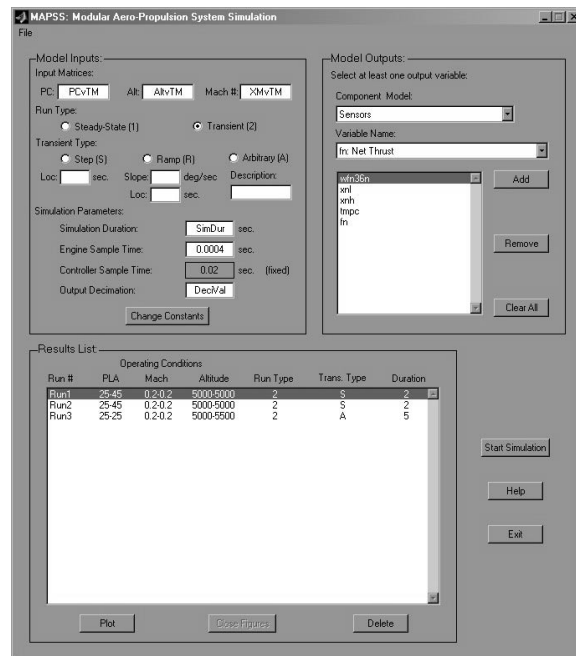


Figure 4: MAPSS GUI.

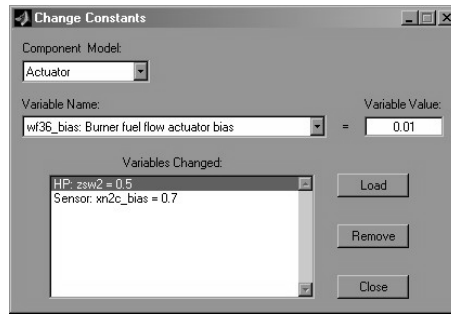


Figure 5: MAPSS Change Constants GUI.

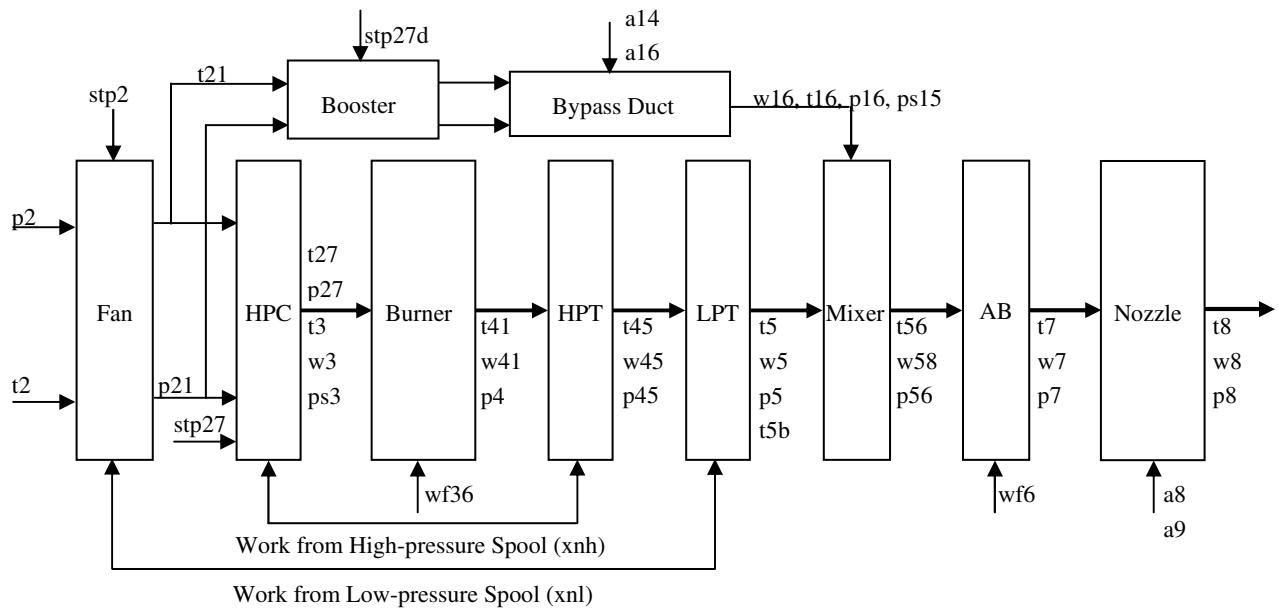


Figure 6: Representation of Engine Components

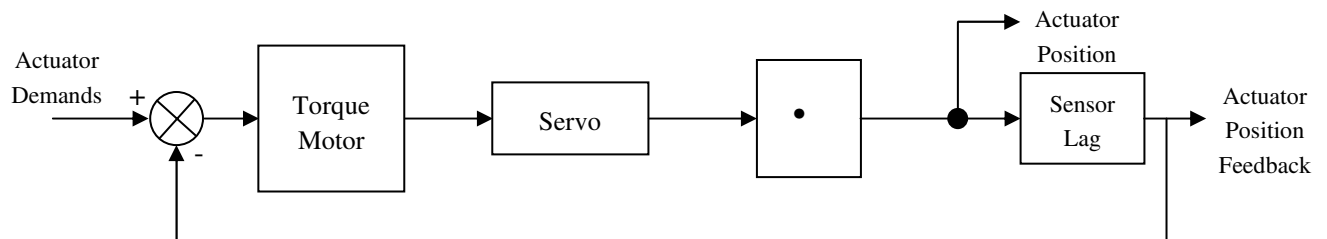


Figure 7: Actuator Dynamics Module.

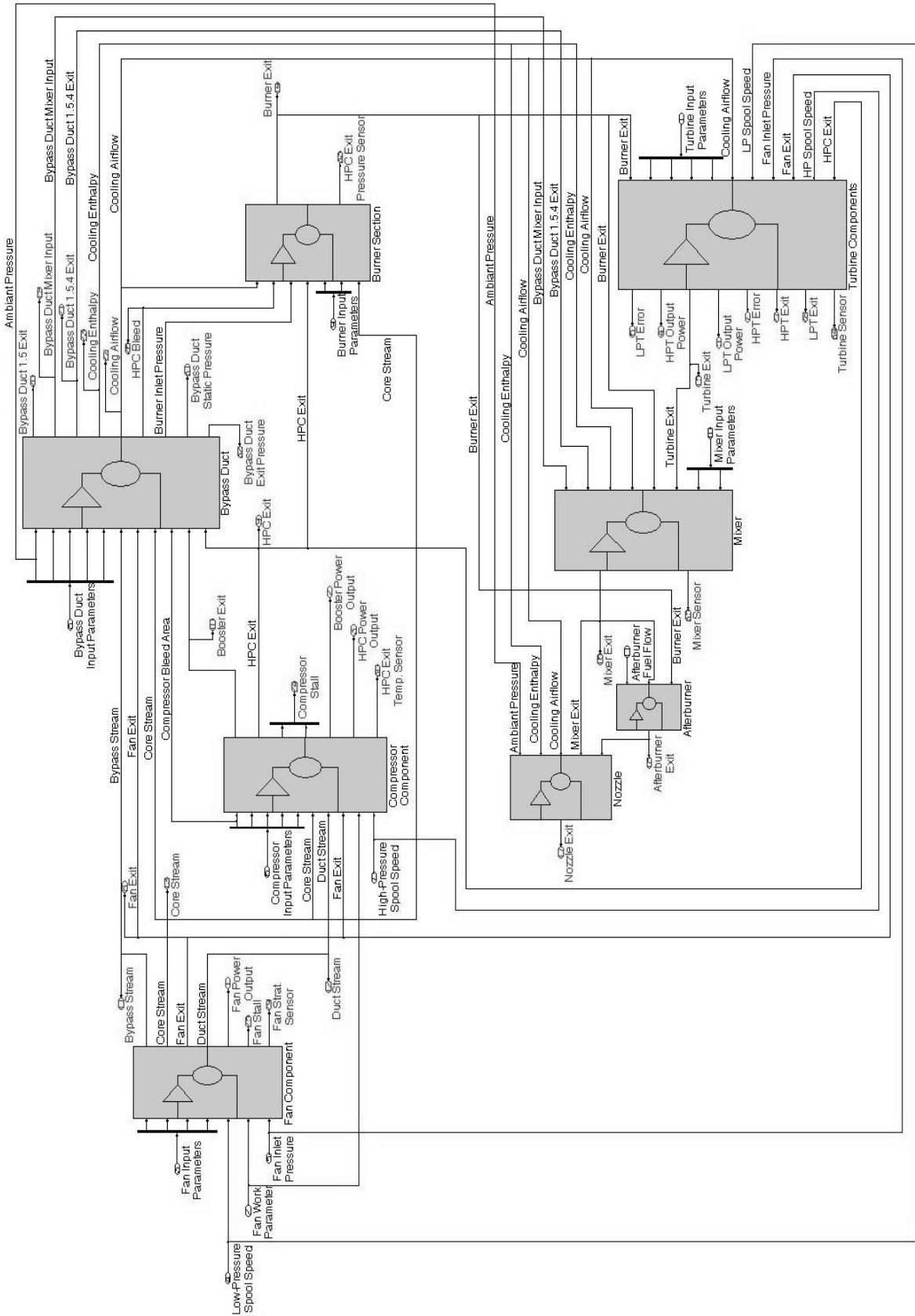


Figure 8: MAPSS Component Level Model.

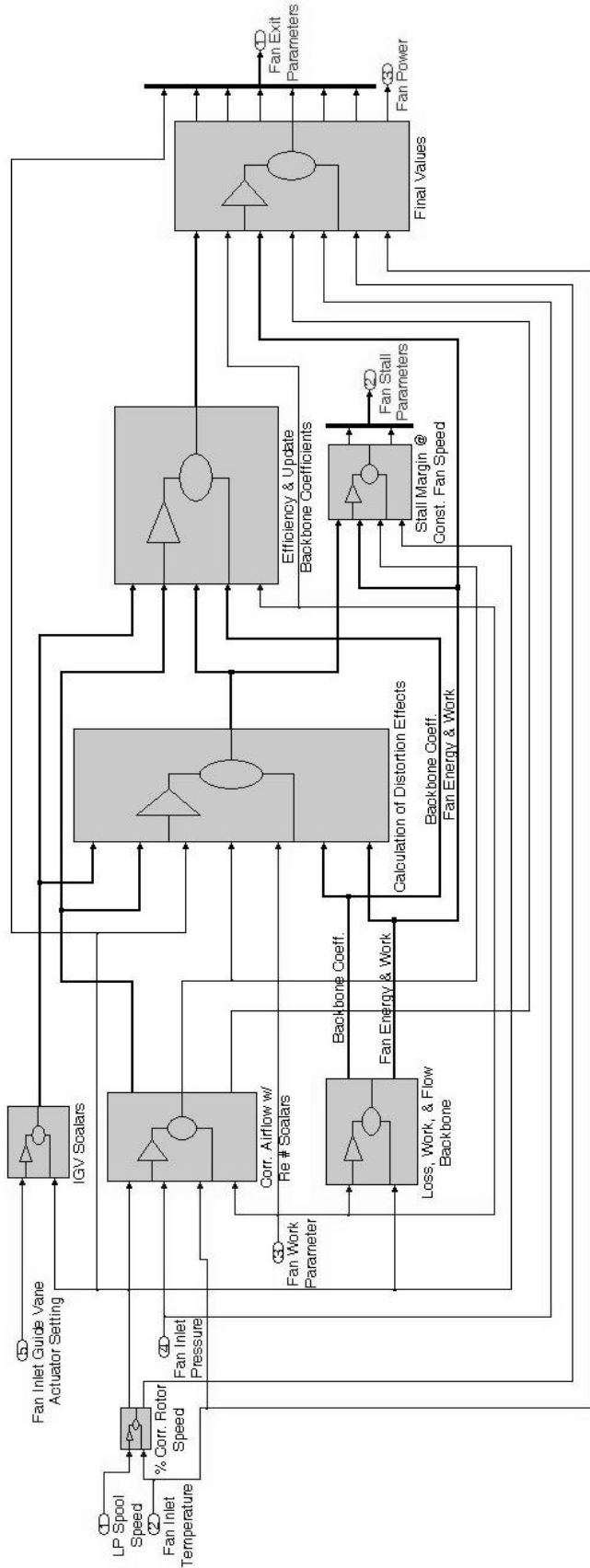
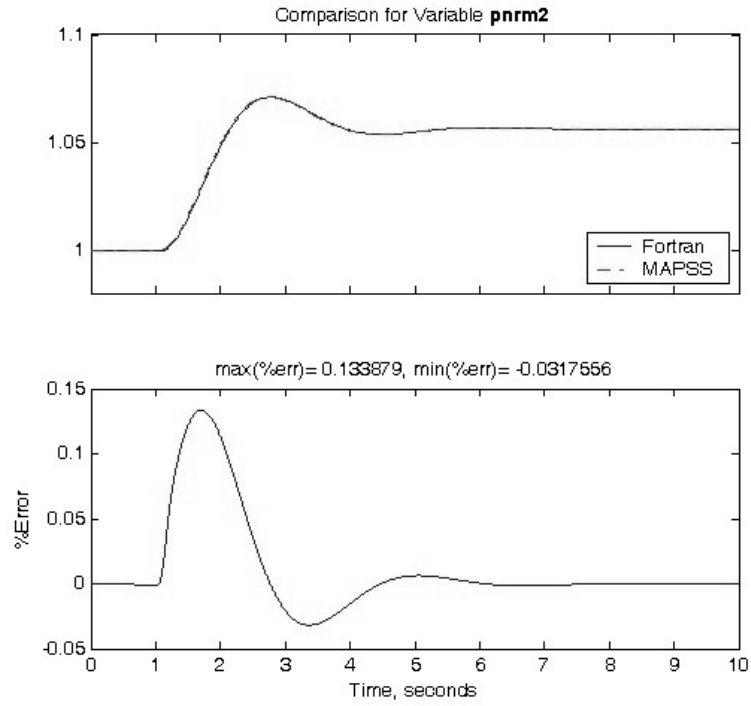
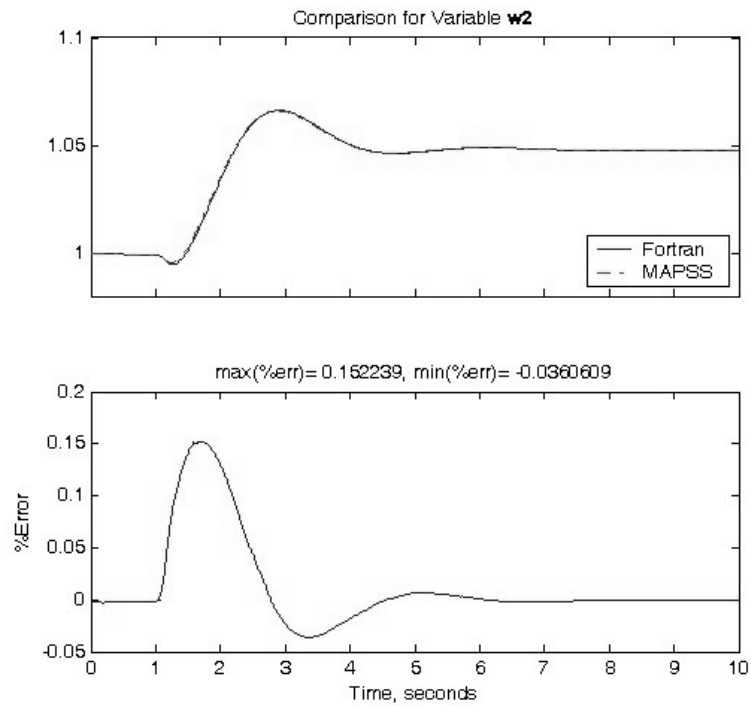


Figure 9: MAPSS Fan Component Model.

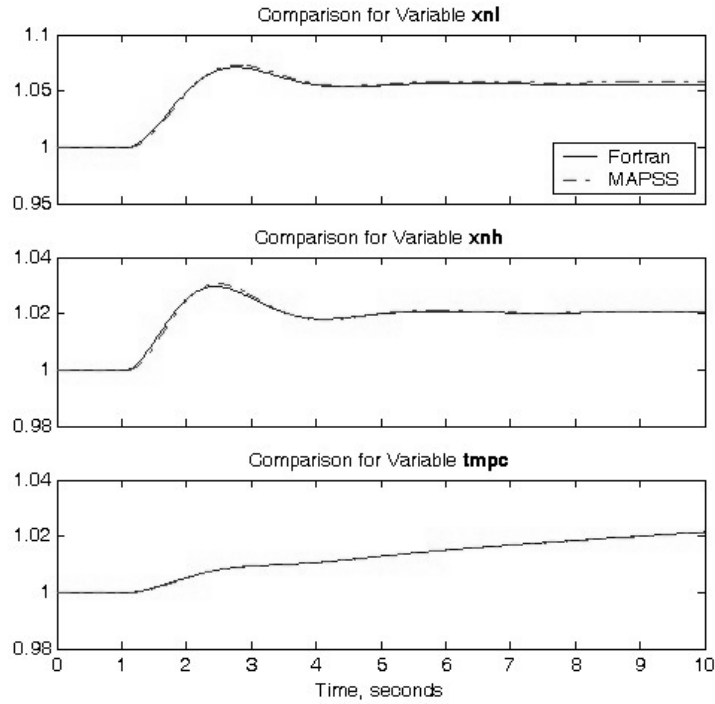


(a) Percent Corrected Fan Rotor Speed

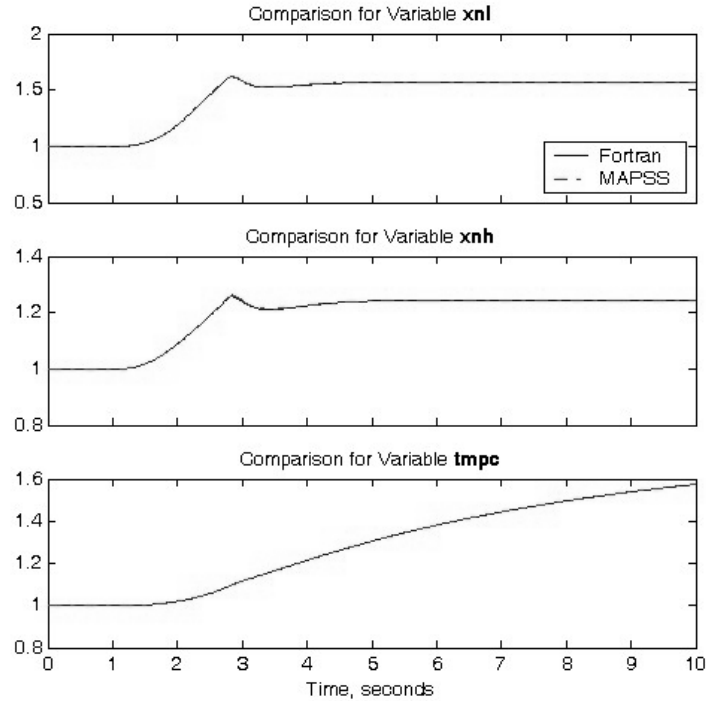


(b) Fan Exit Mass Flow Rate

Figure 10: Open-loop fan exit parameter comparison plots (normalized).



(a) PLA = 21 – 22 deg (Ground Idle Condition)



(b) PLA = 21 – 50 deg (Takeoff Condition)

Figure 11: Closed-loop comparison of model state variables (normalized).

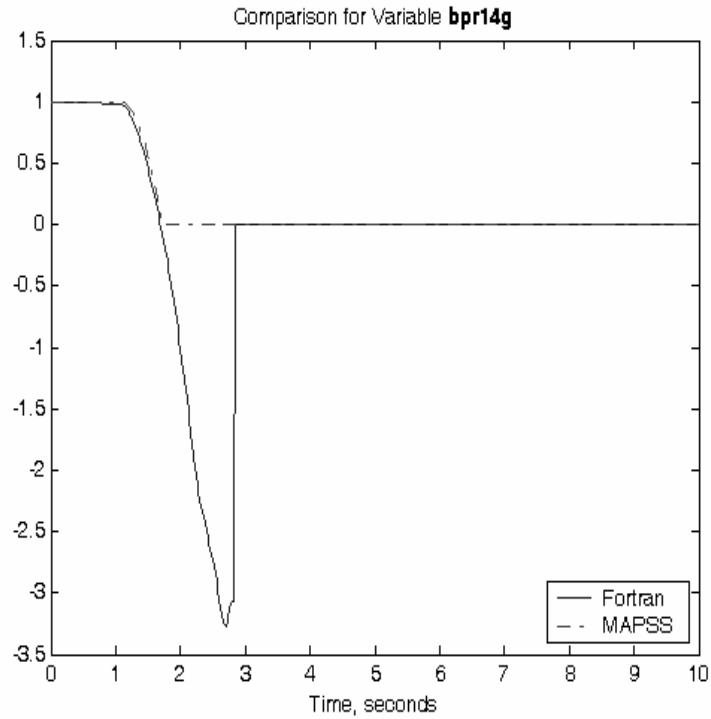


Figure 12: Bypass ratio measurements at forward bypass duct (normalized).

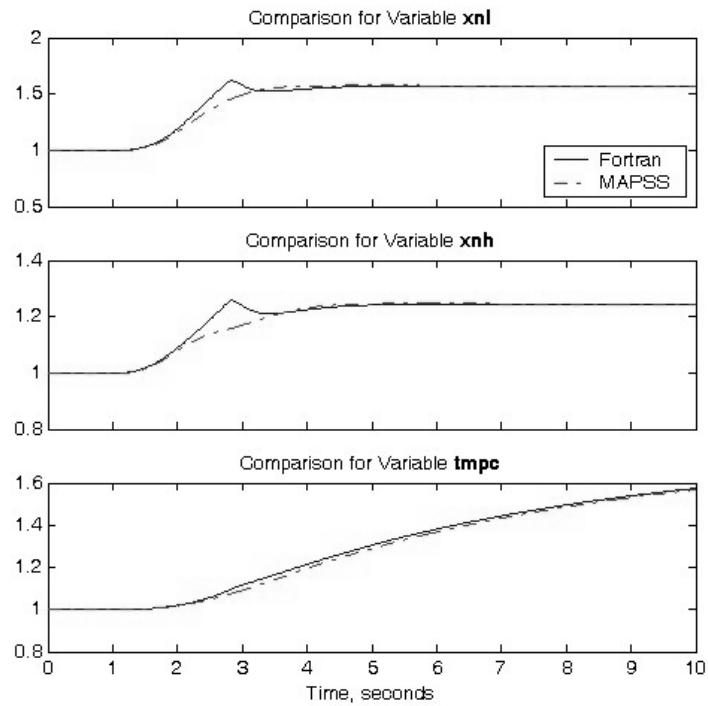


Figure 13: Closed-loop comparison of model state variables ($bpr_{14g} \geq 0$) (normalized).

REPORT DOCUMENTATION PAGEForm Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE August 2003	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE Development of a Turbofan Engine Simulation in a Graphical Simulation Environment			5. FUNDING NUMBERS WBS-22-704-30-01	
6. AUTHOR(S) Khary I. Parker and Ten-Heui Guo				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration John H. Glenn Research Center at Lewis Field Cleveland, Ohio 44135-3191			8. PERFORMING ORGANIZATION REPORT NUMBER E-14109	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-2003-212543	
11. SUPPLEMENTARY NOTES Responsible person, Khary I. Parker, organization code 5530, 216-433-8442.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Categories: 07 and 08 Available electronically at http://gltrs.grc.nasa.gov This publication is available from the NASA Center for AeroSpace Information, 301-621-0390.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This paper presents the development of a generic component level model of a turbofan engine simulation with a digital controller, in an advanced graphical simulation environment. The goal of this effort is to develop and demonstrate a flexible simulation platform for future research in propulsion system control and diagnostic technology. A previously validated FORTRAN-based model of a modern, high-performance, military-type turbofan engine is being used to validate the platform development. The implementation process required the development of various innovative procedures, which are discussed in the paper. Open-loop and closed-loop comparisons are made between the two simulations. Future enhancements that are to be made to the modular engine simulation are summarized.				
14. SUBJECT TERMS Digital simulation; Dynamic system; Models; Computer aided design; Control system design; Engine control			15. NUMBER OF PAGES 20	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	