NASA/TM—2013-217883

Integrated Tools for Future Distributed Engine Control Technologies

Dennis Culley, Randy Thomas, and Joseph Saus
Glenn Research Center, Cleveland, Ohio

July 2013

NASA/TM—2013-217883

Integrated Tools for Future Distributed Engine Control Technologies

Dennis Culley, Randy Thomas, and Joseph Saus
Glenn Research Center, Cleveland, Ohio

Prepared for the
Turbo Expo 2013
sponsored by the American Society of Mechanical Engineers (ASME)
San Antonio, Texas, June 3–7, 2013

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

July 2013

Acknowledgments

The authors wish to acknowledge the NASA Fundamental Aeronautics Program from which support has been provided through the Subsonic Fixed Wing and Aeronautical Sciences Projects.

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

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA Center for Aerospace Information
7115 Standard Drive
Hanover, MD 21076–1320

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Available electronically at http://www.sti.nasa.gov

Integrated Tools for Future Distributed Engine Control Technologies

Dennis Culley, Randy Thomas, and Joseph Saus
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Abstract

Turbine engines are highly complex mechanical systems that are becoming increasingly dependent on control technologies to achieve system performance and safety metrics. However, the contribution of controls to these measurable system objectives is difficult to quantify due to a lack of tools capable of informing the decision makers. This shortcoming hinders technology insertion in the engine design process. NASA Glenn Research Center is developing a Hardware-in-the-Loop (HIL) platform and analysis tool set that will serve as a focal point for new control technologies, especially those related to the hardware development and integration of distributed engine control. The HIL platform is intended to enable rapid and detailed evaluation of new engine control applications, from conceptual design through hardware development, in order to quantify their impact on engine systems. This paper discusses the complex interactions of the control system, within the context of the larger engine system, and how new control technologies are changing that paradigm. The conceptual design of the new HIL platform is then described as a primary tool to address those interactions and how it will help feed the insertion of new technologies into future engine systems.

Introduction

Turbine engines are correctly viewed as complex mechanical systems comprised of a set of specific and highly optimized components. From the fan to the nozzle, the explicit purpose of each component has remained unchanged over time. The design of these components is also highly specialized; dominated by the innovations of highly skilled individuals, sophisticated tools, and a multitude of technologies seeking the optimization of each device. The result is that the contribution from each component to the overall operation of the engine is well understood.

When completely assembled and viewed as an engine system, however, the limitations and weaknesses of each mechanical component become more apparent. Therein lies the purpose of the control system; to enable the sustained optimum performance, safety, and stability of the entire engine system across its full mission profile and lifespan.

The control system is also an engine component. Unlike its mechanical counterparts, however, the control system has drastically morphed over time, having transformed its implementation from the mechanical to present day computerized technologies (Ref. 1). In functional terms, its role has vastly expanded from simple speed control to a broad array of capabilities designed to maximize performance as well as alleviate pilot burden and extract maximum useful life from the engine's mechanical parts.

As an engine subsystem, controls are both highly specialized and ubiquitous, having to interact with virtually every engine component. The control system operates both locally and globally across the engine system, yet the contributions of control hardware and software are difficult to analyze in terms of their individual impact on the system performance. Consequently, these contributions are often excluded from the multidisciplinary design and optimization (MDAO) process, which has become the predominant methodology driving design decisions on turbine engine systems (Ref. 2). The lack of tools that can help quantify the contributions of controls also impedes new research and development and, ultimately, the adoption of new control technologies on production engines. Subsequently, control design in practice is often regarded as a mysterious, trailing-edge step in the turbine engine design process.

Future engine systems will require the presence of far more control capability to implement the suite of advanced technologies now in the pipeline (Refs. 3 to 5). Anticipating this need, better techniques and tools will be required in the development of next generation systems. The key will be to work more cohesively during the early development of individual turbine components rather than for controls to remain primarily in the role of system integrator after the major design decisions have been made.

The expanded use of modeling and simulation will be a key element in this process. One of the first steps in the design of every modern engine is to create detailed numerical models of engine components. These models necessarily emphasize absolute accuracy. However, this type of modeling is typically not suited for controls development because it fails to adequately replicate system dynamics, or at least requires enormous resources to do so. While a mechanical engine system can be relied upon to perform at any valid operating point, it is an explicit purpose of the control system to safely bridge the transition between those conditions. So the requirement for modeling in the controls domain is an accurate replication of the dynamics of the system with possible compensation for the steady state. It is also highly desirable that such models be capable of running *at least* in real-time, whether the modeling is for an on-board application, ground application, or development purposes. In fact, when hardware is involved, real-time is a requirement. This is typically accomplished using piece-wise linear representations obtained from higher order models (Ref. 6).

The development of Hardware-in-the-Loop (HIL) systems is a natural follow on to these modeling activities as they afford the opportunity for low cost testing of control hardware against a realistic simulation of the expected environment. The automotive industry is far advanced in this area, having demonstrated HIL to not only develop control hardware but to evaluate the efficacy of new technologies (Refs. 7 and 8). There are a number of other HIL examples in industrial control networks (Ref. 9), flight control (Ref. 10), and even targeted applications on turbine engines (Ref. 11).

It is clear that new technologies are rapidly changing products in many industries, not just engine control. This is especially true as electronics and controls become more distributed and embedded in the components of complex systems. Less apparent is how these technologies are affecting the product development processes, the supply chain, and even the capabilities of manufacturing organizations themselves (Ref. 12).

In 2009, the Distributed Engine Control Working Group (DECWG) put together a cooperative demonstration of control system integration at the American Institute of Aeronautics and Astronautics Joint Propulsion Conference in Denver, Colorado (Fig. 1). Network enabled control components representative of distributed engine control sensors, actuators, and subsystems were quickly developed by 12 separate companies in a timeframe of about two months. Most of the integration occurred remotely by teleconference simply because it only involved establishing a well-defined interface consisting of power and communications protocol. The full physical integration actually occurred at the conference exhibit. Its success was in demonstrating that separate organizations, in fact, competitors, could collaborate in the development of a common controls technology.

This event actually spurred a renewed interest in the cooperative precompetitive development of distributed control technologies among government, industry and academia that exists to this day. In this community, it is widely recognized that the development of distributed engine control is beyond the reach of any individual organization. Without cooperation and the leveraging of technology development investments, distributed engine control components would be novelties, i.e., they could not exist on the scale necessary to incorporate in production turbine engines.

NASA Glenn Research Center views tool development as one of the main contributions they can make to distributed engine control technology. A new Hardware-in-the-Loop platform is therefore being implemented with the intention of using it to develop and communicate the benefits of new technologies as they apply to turbine engine control; a capability needed across industry.

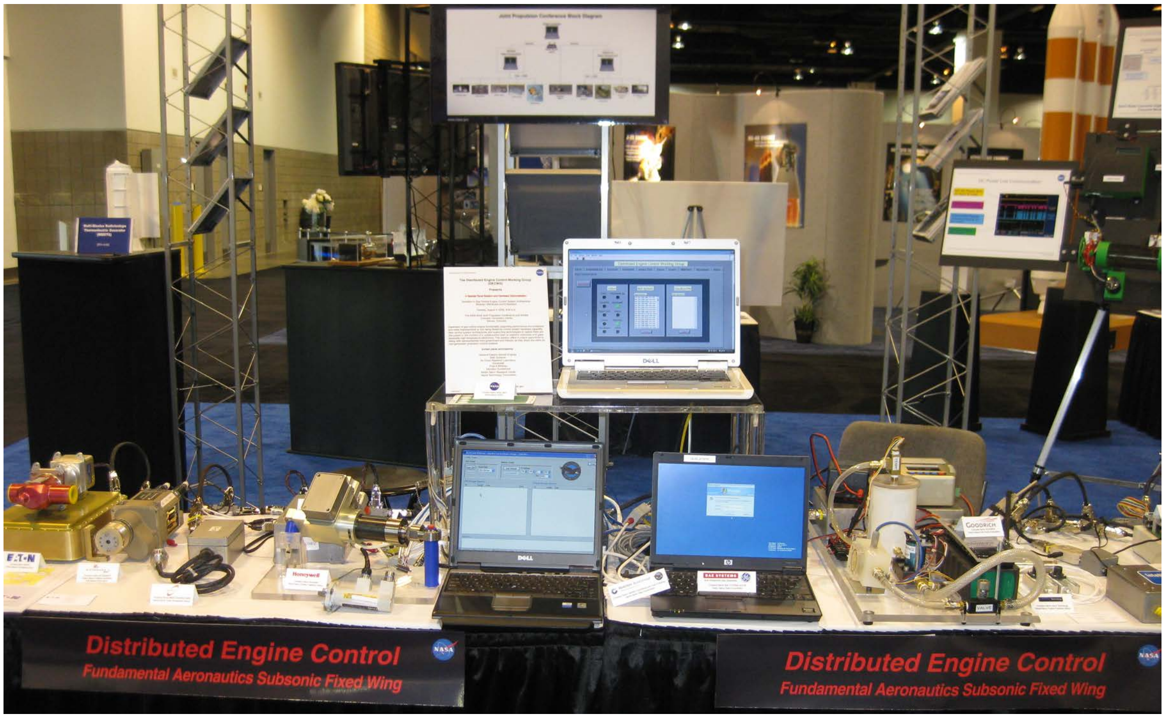
Distributed Engine Control
Fundamental Aeronautics Subsonic Fixed Wing
Distributed Engine Control
Fundamental Aeronautics Subsonic Fixed Wing

Figure 1.—Working demonstration of the integration of distributed engine control components at the 2009 AIAA Joint Propulsion Conference. Twelve companies participated in providing "smart" networked sensors and actuators in a conceptual architecture.

This paper will discuss the changing environment for engine control architecture, a topic relevant to current propulsion research at NASA, and explain how HIL can become a focal point and pathway for the integration of engine control technologies. The paper begins with a discussion of the interaction between the engine system and controls and how this interaction occurs on multiple levels that are not always readily apparent. Next, a description will be offered as to how new distributed control technologies are altering this paradigm and the effect it will have on how organizations interact with each other. Finally, a discussion will be provided that outlines the development of NASA's new Hardware-in-the-Loop platform and its potential to extend collaboration between organizations in the development of new turbine engine capabilities.

Engine System—Control System Interactions

When a complex system is designed, it is natural to try to establish an organizational framework around the system. The organization provides a methodical approach that allows several less complex problems to be worked in parallel. This structure extends not only to the physical design of the system, but often to the way organizational resources are allocated. Although the intent of the structure is to solve the design problem efficiently, it can also create walls that prevent the insertion of new ideas. This is especially true when technical knowledge is imperfectly shared between groups.

The turbine engine is comprised of a fan, compressor, comcombustor, turbine, etc. These subsystems, or components, provide the natural delineations that define the traditional organizational framework for the turbine engine system. The interface between these adjacent components defines all interaction and can be identified by the set of inputs and outputs for each subsystem. Adequately defining and understanding these interfaces is critical to the overall engine design. How well this process is accomplished becomes apparent when the subsystems are integrated.

The control system is an unusual engine component due to the fact that there are interfaces, and therefore interactions, between it and every other component. Some of these interfaces are through the direct physical contact of the sensors and actuators. Some interfaces are virtual; that is, the operation of a component can be indirectly altered when the control logic attempts to optimize the entire engine system. Still other interfaces are even less apparent to control and engine function, such as the structural mounting of hardware and the routing of wiring harnesses. The implication is that control system interfaces are very complex; they occur on multiple levels of abstraction, and are therefore difficult to clearly and completely allocate to individual engine components.

Nevertheless, it is always true that components on either side of an interface affect one another because the interface requires both a definition of requirements and an imposition of constraints (Ref. 13).

Today, turbine engine control is generally implemented through a Full Authority Digital Engine Control (FADEC) with an integral electrical power system, and a suite of sensors and actuators specific to that engine. The control system hardware is fully integrated with the mechanical engine system, making the engine functionally independent of the airframe. Operating the engine requires interaction over a relatively simple interface that primarily includes a throttle command and a fuel source. This engine-airframe interface tacitly acknowledges an organizational boundary between different manufacturers.

Traditionally, the FADEC, or engine controller, has many specific interfaces that exist as dedicated physical connections to hardware control elements scattered throughout the engine system. Some of these are shown in Figure 2. Internally, the FADEC is essentially a digital computer with a substantial number of I/O channels ported off a common internal data bus. The software application code is often accompanied by hardware logic for redundancy of critical control elements. The design of the FADEC hardware and software, a subsystem of the control system, is therefore dependent on the design of the entire control system. It also follows that the larger control system design, a subsystem of the engine system, is dependent on the design of the entire engine system. Control engineers are often heavily invested with closing out the engine system design process under severe cost and schedule constraints because they are at the tail end of the critical path. This approach leaves little room for new technology insertion in the control system.

Fortunately, the control system can be visualized from a different perspective. In a virtual sense, the control system is simply a set of algorithms that operate on input data to provide output commands that control the operation of the engine. In this abstraction, control occurs strictly through the processing of algorithms that define the engine operation. There is no hardware definition or location dependence for processing in this view. The real requirement is the timeliness and accuracy of the data flow connecting functional elements of the control system.

Designing the control system from a functional perspective rather than beginning with a preconceived hardware configuration, like the current FADEC, leads to a more flexible conceptualization of the control architecture. The decomposition of control functions into elemental constructs is fundamental to the systems engineering process. These primary elements can then be arbitrarily reconstituted into flexible and modular blocks representing various system architectures. The design decisions regarding how to assemble these blocks can, and should, take into account all aspects of the system design, especially how constraints are allocated among the various control blocks. In this way, constraints can become a system variable instead of a fixed parameter.

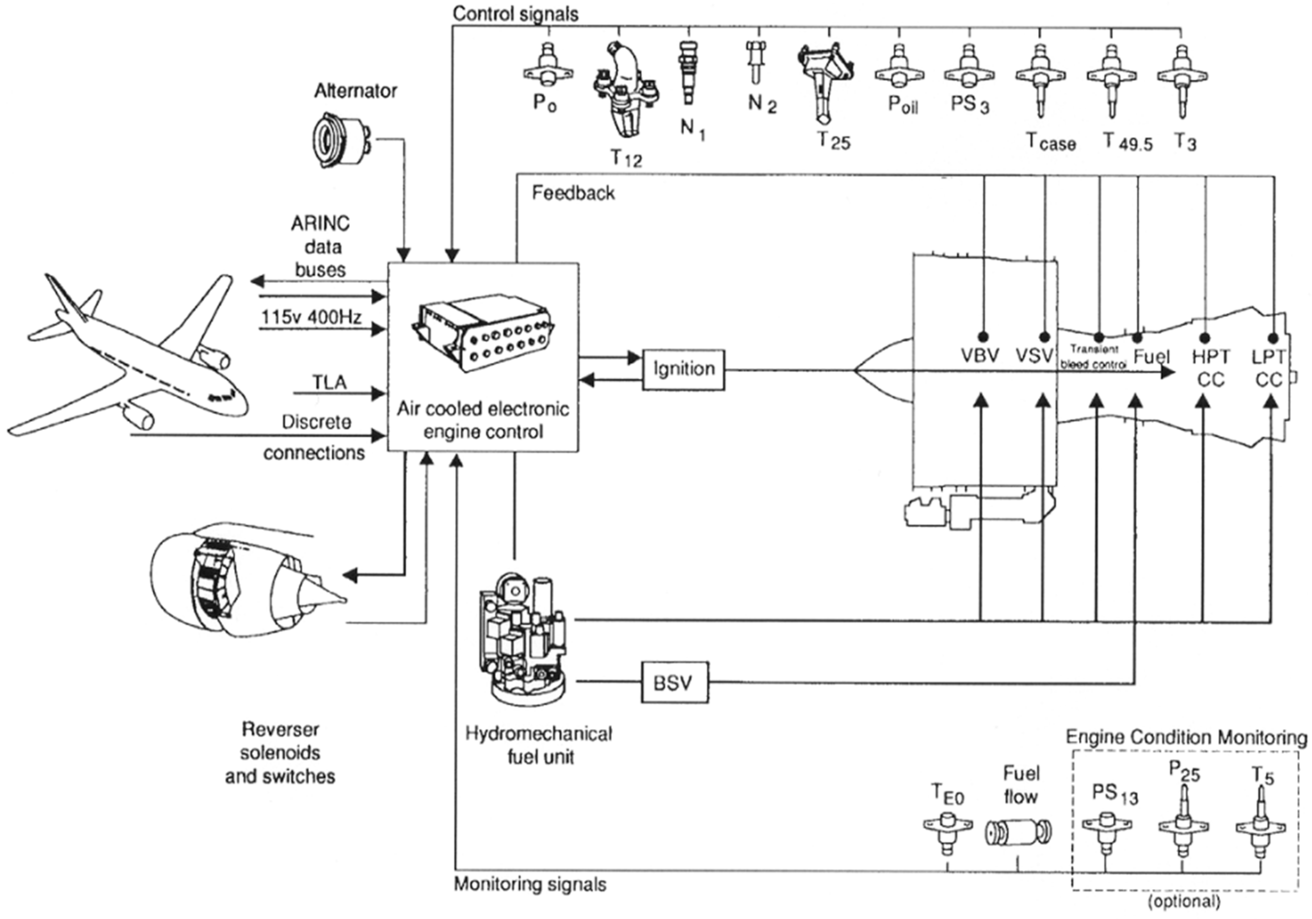

Figure 2.—A typical engine control system is shown. The centralized engine control unit, or FADEC, has many discrete analog interfaces that are specific for each control element. The controller also has interfaces that integrate the engine control into the airframe.

Control System—Impact of Control Architecture

As previously mentioned, organizational and technological issues strongly affect how complex problems are solved. Control system design is no exception. However, the barriers presented by organizational concerns seem to persist for longer periods, even after technical solutions offer new opportunities for progress. These barriers are likely due to imperfect knowledge in an organization and a natural aversion to risk. The limited progress in advancing engine control technologies is often an example of an organizational barrier at work.

All control systems have basic elements consisting of sensors, controller, and actuators. For obvious reasons, the functions of sensing and actuation are restricted to specific physical locations on an engine; you can’t sense something without a sensor or move a control surface without an actuator. This limits the degree of design freedom in placing controls.

In addition, most control hardware is also strongly affected by environmental factors that result from the normal operation of the engine. This is a consequence of an underlying dependence on electronics, which are especially affected by temperature. Minimizing the number of heavy enclosures to protect electronics usually results in minimizing the overall cost function of the engine design. Preferably, these enclosures are situated in a relatively cold location on the engine, typically the fan casing, as shown in Figure 3. The aggregation of electronics in centralized enclosures leads to a centralized control architecture, resulting in a FADEC with complex interfaces that are typically expressed externally as an equally complex web of wiring harnesses. Adjusting the location of any control element, especially the FADEC, is not trivial. Still, it is relatively easy to visualize an item being moved around the engine envelope while dragging its interface cables around with it.

Moving a centralized FADEC to a new location on the engine typically does little to alter its functionality. Nevertheless, the move can have a substantial impact on engine system weight (changes in harness length), maintenance (accessibility of control elements), and ancillary support systems (heat removal), etc.

One of the major technical barriers underlying distributed control is the need to overcome the environmental constraints

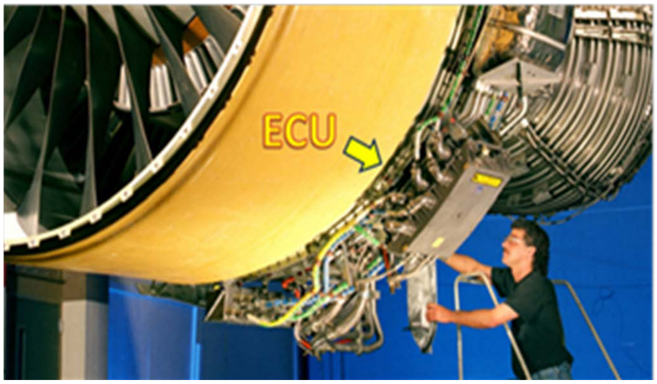

Figure 3.—The engine control unit (ECU), or FADEC, is shown mounted on the casing of a high bypass turbine engine, illustrating the size and complexity of the control system hardware and interfaces. Public release photo courtesy of Pratt & Whitney.

on control electronics. This is not a constraint on control itself but on the fundamental technology underlying control components. Consequently, these constraints exist regardless of the control implementation approach used on the engine. However, sufficiently raising the operational temperature for control electronics allows the physical implementation of control functions to more easily replicate any of the arbitrary control schemes that can be envisioned through the system engineering process previously described. This includes local control intelligence and digital communication capability embedded in each control element. High temperature electronics clearly enables *more distributed* engine control, assuming the absence of other barriers, like cost.

Once the primary barriers to distributed control are removed, the physical elements of the distributed control system can be included and evaluated at the engine component level during component design. Interface complexities and design dependencies between control elements largely disappear because they communicate over a digital network. Control system integration becomes a much simplified task involving data flow analysis between control elements, which can be accomplished very early in the design process. The fundamental control problem at the time of engine system integration then becomes "*how to integrate the control system*" instead of the original problem, which was "*how does control integrate with the engine system*." The "cost" of control, and by default the benefit, becomes a quantifiable part of the design process instead of a constraint on the end product. This shift in approach opens up the possibility of simplifying the evaluation of new technology insertion at an earlier stage of development.

Just as high temperature electronics fundamentally alters control integration, distributed control fundamentally alters engine system integration. Consider the case of an engine sensor such as Ps3, the static pressure sensor at the exit of the compressor. The Ps3 sensor signal is used in the engine controller for a variety of reasons that include calculating overall pressure ratio, specific fuel consumption and thrust estimation, and as a limiting value for compressor operation. For the purpose of control, the signal from the sensor simply becomes an input value used in processing the control laws.

In a typical physical implementation of centralized engine control, the Ps3 sensor (static pressure at the exit of the high pressure compressor) is housed in the environmentally controlled FADEC enclosure and connected by a long pneumatic tube to the casing wall at the high compressor exit. The sensor is supplied by the FADEC with whatever electrical power it requires. Meanwhile, the sensor output signal is filtered, amplified and digitized as one of many I/O channels. From there, the signal is made available to the FADEC for any additional signal processing prior to use in the control logic.

The distributed control implementation, in contrast, would package the Ps3 sensor, power conditioning, signal conditioning and digitization together at the high pressure compressor casing in a "smart" sensor module. Included in the package would be the electronics for communicating the Ps3 data over a digital network. These differences in implementation are shown in Figure 4.

The system impact of this change is profound, if not readily apparent. At the compressor, the simple pneumatic penetration is replaced with the sensor module containing the volume and mass of the transducer and associated electronics. The physical complexity is locally increased at the compressor, but the packaging constraints are not as rigid as when Ps3 was one of

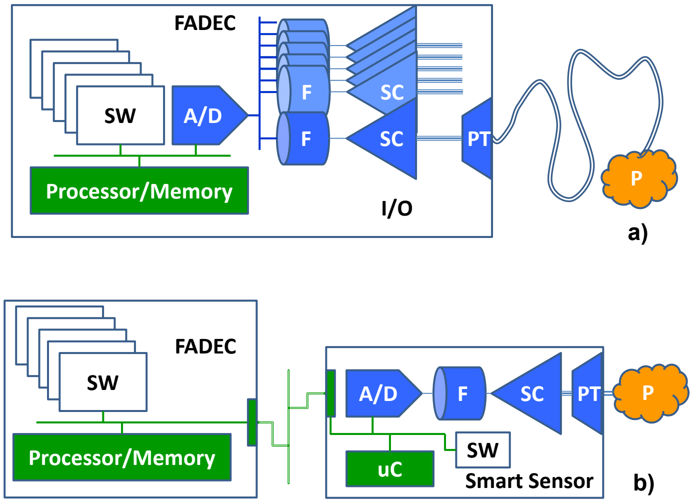

Figure 4.—Implementation examples for obtaining Ps3 data; (a) Centralized architecture with multiple channels of I/O in the FADEC; (b) Distributed architecture with an all-digital FADEC and external "smart" devices connected by a digital network. SW = software control applications, A/D = analog-to-digital conversion, F = filter, SC = signal conditioning, PT = pressure transducer, μC = microcontroller, P = Physical Process being sensed.

many collocated I/O inside one FADEC enclosure. Instead of using a rigid pneumatic tube, the pressure signal and electric power are transmitted over a standard interface using a thin bundle of wires. In the case of the distributed FADEC, all the dedicated, analog I/O electronics have been removed and replaced with an interface for multi-drop communications. The volume, mass, and complexity of the new FADEC is therefore substantially reduced.

In the new paradigm, the control "function" of producing Ps3 data is wholly contained in one component located on the engine casing. Meanwhile, the FADEC is now entirely digital and is no longer constrained by a multitude of physical interfaces and the weight of associated cables. Since the FADEC is no longer dependent on the I/O configuration, its physical design is largely independent of changes in the engine configuration, including changes to the number, configuration, and function of sensors and actuators. Perhaps most dramatically, the FADEC box could even be located off-engine for additional weight reduction.

From a systems perspective, the Ps3 smart sensor module can be "owned" by the compressor component designers who would specify its entire function. The specification can be released to a pool of suppliers that would manufacture it as a device to be plugged into the control system. The functionality of the module could even be expanded by design, or as a value-added feature, without significantly impacting the engine controller hardware design. For example, temperature measurement or compressor stability detection could be incorporated into the same smart device.

Distributed control has the capability to change the design and development processes of the entire turbine engine. This one example served to illustrate the complex interactions between the engine manufacturer, system integrator, supplier and air framer, which were all affected. Due to the control system's prominent role in the integration of the engine system, it is very likely that a change in control architecture to distributed control will create changes in each of these organizations as well.

Hardware in the Loop System

The NASA Hardware-in-the-Loop system is a research tool primarily intended to develop and demonstrate distributed control technologies. It is also being developed as a non-proprietary system tool to promote collaboration across the controls community. There is potential to eventually create a seamless platform that can bridge the processes from control system design through hardware development, integration activities, and even into system verification. The initial goal is to create a scalable system that can be made available to users who will implement as much or as little of the real-time hardware capability as they require. As an open-ended development tool, HIL can proceed in stages. The approach is to integrate a variety of design and analysis tools into HIL to assist the unrestricted exploration of turbine engine control technologies.

The HIL system, as currently envisioned, has three distinct parts: an Engine Plant Model (EPM), a real-time Control System Platform (CSP), and the User Interface (UI), as shown in Figure 5. The purpose of the EPM is to provide accurate stimulation to the control system sensors while realistically responding to control system commands. The EPM will also be responsive to signals representative of ambient condition inputs as it processes mission profiles in the simulated environment. The purpose of the CSP is to provide a platform that allows users to easily design, develop, and execute models of the control system, including individual distributed control components. If desired, these control component models can be transitioned to hardware, eventually replacing the simulated model in the CSP. In addition to the sensor and actuator interfaces with the EPM, the CSP will interface to signals representative of the airframe and cockpit controls as was previously shown in Figure 2. Lastly, the UI will interact with both the EPM and the CSP. Operationally, the UI will provide the capability to control the simulated engine, through the control system, in a manner representative of the engine-airframe interface. As a research tool, the UI will also allow deeper interrogation and manipulation of EPM and CSP variables to gain fundamental understanding of performance characteristics while also having the ability to introduce anomalies.

Like distributed control, the HIL is organized as a modular system and is dependent on the interfaces between its components. These are described in some detail below.

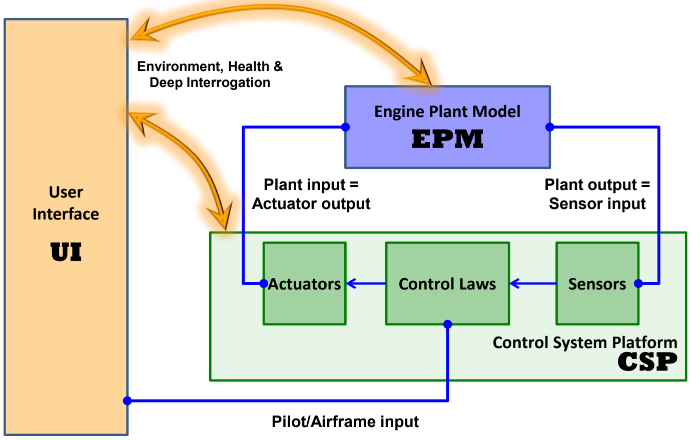

Figure 5.—The functional interfaces of the control system are all rigidly defined to strictly replicate the type of information that flows between the engine and control system. However, the User Interface (UI) is also able to perform deep interrogation of both the engine and control simulations.

Engine Plant Model

HIL system development begins by leveraging the NASA Glenn software known as C-MAPSS40k, which is representatively shown in Figure 6. C-MAPSS40k is a realistic 0-D simulation (a non-linear model lacking radial and axial detail), representative of a 40-thousand pound thrust class turbine engine. It includes the accurate engine dynamics that are necessary for controls development. It runs under MATLAB/Simulink (MathWorks, Inc.) and is capable of much faster than real-time execution in a common PC platform. This software is available to U.S. citizens only.

The existing C-MAPSS40k software is very modular in design in order to accommodate user modifications to individual engine components. The code also incorporates basic controller algorithms that allow engineers to evaluate engine performance under various conditions. As a first step, the EPM in C-MAPSS40k will be isolated from the control system, the ambient environment, and the airframe. Interfaces between these components are defined in such a way that they replicate the flow of data across the physical boundaries of an engine and its control system and are therefore accessible to the user as they would be in the physical system. The goal is to isolate the EPM and create a common interface definition and methodology that defines how users interact with the engine. Eventually, this format is expected to enable engine plant simulations from other sources to be inserted into the HIL system in place of the non-proprietary C-MAPSS40k. This feature will be useful to users of the HIL tool for their proprietary models.

All of the calculated variables of the engine plant are classified as engine plant outputs and will be accessible as digital values in engineering units, if applicable. All of this information will be available to the UI. A subset of these EPM output values represent sensed parameters and these will be made available to the control system as sensor input values.

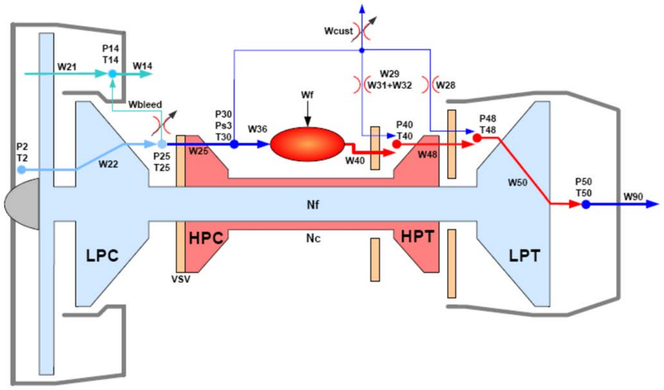

Figure 6.—Diagram of the C-MAPSS40k engine simulation which will form the basis of the Engine Plant Model to be used in the development of the HIL. The model provides the realistic engine dynamics necessary for controls development and it can run in real-time or faster.

Input variables to the EPM will likewise be in engineering units. Those values that represent actuator outputs will originate from the control system. The remaining EPM inputs will be provided by the UI and will represent quantities that originate from one of three sources: simulated ambient operating conditions, the simulated airframe, or user-determined engine health parameters. These health parameters are intended to provide the user with the capability to test the operability of a degraded engine.

The interface definition surrounding the EPM is of special significance, assuming HIL has the capability to “drop in” other EPMs. One set of issues regards intellectual property while a second regards the complexity of the interface. Accommodating these collaborative user concerns in the HIL design is explained below.

EPMs represent a very significant investment in an engine company’s intellectual property that must be adequately protected. It is likely that these models will at least need to be provided as an executable or use some other method of obscuring the source code. In addition, legacy engine models may exist that have been developed outside of the MATLAB/Simulink environment. Instead of disclosing or recoding these models, a process for integrating them into the HIL environment will need to be developed. In either case, it would be unusual for an organization to allow such valuable information to be held in the control of another party.

The issue of complexity arises in the form of how the engine model interfaces with the rest of the HIL system, especially when control hardware is involved. Keeping the complexity of physical control hardware isolated from the EPM simplifies this interface. The EPM must produce the data that a hardware control sensor eventually senses and it must accommodate the actual response of a hardware control actuator. However, it does not have to be involved in creating the physical analog of these interfaces. This complexity will be discussed in the section on the CSP.

There are advantages to running engine models and control models on the same hardware platform. However, maximum flexibility and independence is likely to be achieved by running them on separate machines connected by a high speed data interface. Both options may likely need to be accommodated.

Control System Platform

The Hardware-in-the-Loop test bed is primarily intended for the development of control technology as it relates to distributed control. This is a fundamentally different problem than what is encountered in test beds for centralized control architectures due to the nature of the control interfaces.

The controller stripped out of C-MAPSS40k does not reflect any of the detail of control hardware since it operates at a high level of abstraction. However, research in distributed control is intended to further the understanding of how the hardware configuration affects the performance of both the control system and the engine system. This includes a better understanding of system constraints, which can play a huge role in determining whether technology is adopted. While performance can include such things as transient response and stability, constraints may involve a wide array of parameters like weight, reliability, fault isolation, and cost. A detailed understanding of the control system components and interfaces is required to accomplish this task.

The first step in development of the CSP will be to create a library of elemental control functions in simulation. These can be used as modular blocks to build any type of control system component including sensors, actuators, data concentrators, loop closure elements, complex subsystems, etc. Over time, it is expected that control system element models will be provided by many sources, including actual production suppliers of control components.

Control component models, once assembled, can then be connected in simulation using various system configurations representative of a control architecture. In a distributed control system, the interconnection of control elements occurs through a digital network. However, the network is another example of a control system component. It too must be modular in nature so that it can be upgraded or replaced as technology improves.

As in the EPM, control system interfaces will be rigorously enforced. All communication between the EPM and the CSP will be through sensors and actuators. Any data that would normally pass between the control system and the airframe, such as throttle position, will originate at the UI; however, those signals should use the appropriate control system interfaces designed for that purpose. Of course, the CSP will produce and use other data in its internal calculations, which will also be visible at the UI, when operating in simulation mode. Such data include parameters like system health, which govern faults or degradation. Once control hardware is inserted in the system, however, the component interface may preclude such information from being accessible to the UI. In other words, it is possible to deeply interrogate a computer model of a component, but it is not so easily accomplished when that component exists in hardware form. These interfaces are shown in Figure 7.

Network communication is a component of distributed control that serves the critical function of gluing the control modules together into a cohesive system (Refs. 14 and 15). One or more control networks, of different types, can potentially exist in any given control architecture. These control networks are expected to impact system stability and reliability as they introduce sources of delay, noise, and data loss. Engine network technology is intimately linked to the capability of high temperature electronics and may change rapidly.

In these distributed systems, it is the network that primarily defines the interface between components. Using interface-based design (Refs. 16 and 17), it is possible to allocate system functionality in a variety of network connected architectures. Analyzing the data flow between these elements can reveal the overall system performance as well as define the individual functional performance of each connected control element. By strictly conforming to these interface definitions, any simulated component can be replaced by its hardware equivalent in a manner that is transparent to system operation.

It must be pointed out that the time base in which all EPMs, control system models, and control hardware operate under must be consistent. They do not necessarily have to be synchronized, but the unit of time must be consistent in all components in order for their interactions to be properly evaluated. When control hardware is involved, real time is the only possible time base; otherwise simulations can occur faster than real time.

Other complexities also arise with control hardware. A hardware control sensor is designed to convert a physical parameter, like pressure, speed, or temperature originating from the engine, into an electronic signal. Likewise, a hardware control actuator is designed to convert an electronic command into a physical action. In simulation, this is not a problem because there are only digital data originating or terminating at the EPM. When hardware is involved, the physical nature of the control devices must be confronted.

One method of dealing with hardware control actuators is to run the hardware open-loop, but in parallel with an instance of the actuator model that feeds digital data into the EPM input. An alternate method is to construct a specialized apparatus to properly load and measure the action of the hardware actuator, converting the physical parameter into data for the EPM input. This apparatus is depicted in Figure 7 as the analog subsystem. This latter method runs the risk of introducing dynamics into the actuator response that may not exist in an actual application.

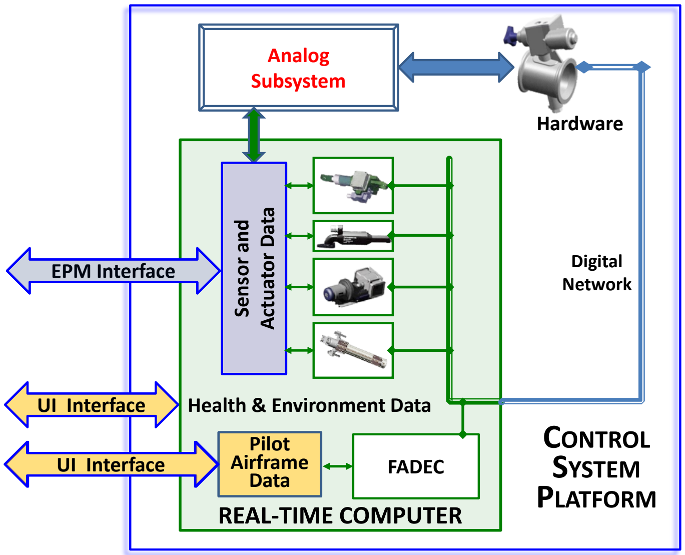

Figure 7.—The Control System Platform (CSP) consists of a real-time computer running control element models and an analog subsystem to accommodate the physical interfaces of control hardware. All interfaces attempt to mimic those of a real engine system.

For hardware sensors there is typically no convenient way to bypass the need for a specialized analog interface in which to stimulate the sensor. However, in some instances, it may be acceptable to use an electrical analog of the signal instead of a physical analog. This can be accomplished by inserting the electrical analog signal downstream of the transducer inside of the smart sensor. Of course, this capability would need to be accounted for in the design of the sensor. As previously stated, attempting to generate the physical analog of an engine parameter is prone to spurious dynamics and safety issues unless carefully constructed.

User Interface

The UI will provide a means for the user to configure and operate a control system while examining those metrics that are relevant to controls and to engine system performance. The idea is to have an ability to quickly create, test, and evaluate relevant situations that will help optimize control system design. This is best accomplished using a unified platform instead of a collection of separate component interfaces.

The UI will interact directly with the engine (EPM) and control system (CSP). This basic interaction can be summarized in the following list:

- Create a library of modular control functions
- Use library functions to design control elements
- Use control elements to design control architectures
- Assign engine-input/control-output variables and engine-output/control-input variables
- Operate control architectures through engine-airframe interfaces
- Provide mission profile scripts for control system operation and the ambient environment for both engine and controls
- Provide scripts to introduce engine or control system faults or degradation
- Collect and archive data from the control system
- Collect and archive data from the engine system
- Analyze data sets

The interaction between the engine and control system is limited and defined by specific interfaces that replicate the same interactions that occur in the physical engine system. However, no such restriction is necessary with the UI except when the UI is simulating the pilot and airframe interaction. In fact, the ability to visualize and manipulate all system information is the UI's greatest asset. This capability can allow efficient data collecting, archiving, and post processing operations.

The exact implementation of the UI may depend on what a particular user intends to accomplish with the HIL system. If there is no intention to use external EPMs beyond C-MAPSS40k, then the EPM could be integrated into the CSP. If there is no intention to introduce hardware into the HIL, then there is no need for a real-time computer in the CSP. The implementation of the entire HIL system, consisting of UI, EPM and CSP, could then feasibly exist on a single personal computer. However, the maximum capability of the HIL is achievable only by adhering to the interface definitions that allow the system to be easily scaled up or down. The interface definitions between the three components are functional interfaces, if not actually physical interfaces.

The UI will also host an additional range of tools that are required to thoroughly understand system constraints.

Tool Development

A wide variety of tools can be developed for use in the HIL that can evaluate more than just performance. Many of these tools can be associated with a stage of development, e.g., design, operation, or analysis, or whether the application involves simulation or hardware. As a non-proprietary system, however, these tools should be designed to be compatible with each other, at least to the extent that data can be shared. Ultimately, one of the end goals is to produce information that will allow control systems to be an integral part of the MDAO effort in engine design.

The need for many tools can be drawn from the previously presented discussion. For example, as control elements are constructed in simulation it would be useful to track information, including estimates for reliability, weight, and cost. When an architecture is assembled from these elements, tabulating knowledge about these parameters would be useful in constructing trade studies.

Likewise, as simulated control elements are located on the "physical envelope" of the simulated engine, environmental information extracted from the engine simulation could be used to develop specifications for a hardware design. Later, this environmental information could be used to provide a forcing function to test actual hardware for temperature and vibration.

Locating control elements on an engine structure is also complicated by the routing of wiring harnesses. Having a capability to place these components, then automatically route cabling while tabulating weight and pin contacts, could be extremely valuable. A notional depiction of how this might interact with computer aided design software is shown in Figure 8.

Automated monitoring of data flow through the control system is necessary in order to understand stability, transient response and performance. This capability will allow the requirements for individual components and networks to be quantified. It will highlight system bottlenecks, allowing the optimization of control architecture as well as the analysis of system faults and anomalies.

There is also a need to consider the future direction of hardware and software technologies. For instance, commercial computing platforms are moving rapidly in the direction of powerful multi-core processors that provide the capability for concurrent as well as parallel execution of software. This capability inherently exists in a physically distributed control system; however, tools to evaluate and optimize the partitioning of these functions across a control system are needed. Even within the HIL simulation platform, this capability enables complex models to be subdivided and executed in parallel. In fact, each control element model in the HIL could eventually be assigned to its own processor. In this way, the simulation process could be exactly analogous to the physical implementation of a distributed system. There are active research projects, such as the Ptolemy Project (Ref. 18), that are investigating ways to take advantage of these powerful computing platforms, especially as to how they relate to real-time and embedded processors.

Another aspect of distributed systems is their ability to accommodate new types of control hardware. The hardware technology of high temperature electronics occupies a relatively small but important niche. Since one of the primary purposes of HIL is to develop high-temperature capable smart control effectors, the limitations of this type of electronics must be well understood. HIL could play a significant role in the development of standardized programming tools for embedded, high-temperature electronics software and firmware.

Distributed control components will likely follow similar development paths whether they are physical hardware or simulation because they are likely to take advantage of reusable design elements (Ref. 19). Portions of control components can be extracted from a standard library just as circuits are constructed from a catalog of electronic components. As individual simulated elements are constructed, they can be tested through their defined interfaces prior to insertion into the control system. This naturally leads to development of verification and validation technologies.

By designing the HIL system as a flexible, scalable platform, it has the potential to bridge the control development process from low level research through hardware development, including the potential to extend into the realm of control system verification and validation. Without such a capability driving the controls community toward cooperative interface standards development, it is likely that the integration of control systems will continue to be a costly and time-consuming process.

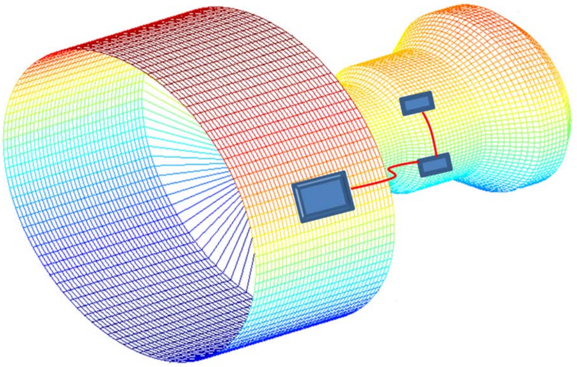

Figure 8.—A notional depiction of how tools accompanying the HIL platform could interact with other engine design processes. Here, physical placement of control system components on the engine envelope could be used to assess the weight of control architecture configurations or associate engine environmental characteristics to control elements.

Summary

A Hardware-in-the-Loop (HIL) system is being developed at the NASA Glenn Research Center with the intention of developing distributed control technologies and understanding control system constraints. The HIL is being based on the NASA C-MAPSS40k software, which simulates the dynamics of a realistic 40-thousand pound thrust class turbine engine. The HIL will be modular, consisting of three major components: the Engine Plant Model (EPM), the Control System Platform (CSP), and the User Interface (UI). Interfaces between the three subsystems are defined to closely correlate with the interfaces found in actual aero-engine applications, specifically engine-to-control-system and control-system-to-airframe. The CSP will host the simulated components of a distributed control system including simulated communication network structures. All of these components can be assembled from a resident library of basic control functions and subsequently connected into arbitrary control architectures that can be tested and evaluated in the HIL. Hardware elements, potentially derived from initial HIL simulations, can be integrated into the HIL and the entire system run in real-time. Identified for inclusion in the UI will be a host of tools to aid in the design, optimization, and integration of control systems, including the ability to quantify system metrics and benefits that will lead to enhanced participation in the design of future engine systems.

References

1. Jaw, L., Mattingly, J.D., 2009, *Aircraft Engine Controls: Design, System Analysis, and Health Monitoring*, American Institute of Aeronautics and Astronautics, Inc., pp. 1-35.
2. Gray, J.S., Moore, K.T., Hearn, T.A., Naylor, B.A., “A Standard Platform for Testing and Comparison of MDAO Architectures,” 53rd AIAA Structural Dynamics Conference, 2012, Honolulu, HI, United States, NASA/TM-2012-217652, E-18330, AIAA Paper-2012-1586.
3. Lord, W.K., MacMartin, D.G., Tillman, T.G., “Flow Control Opportunities in Gas Turbine Engines,” AIAA Fluids 2000, Denver, CO, AIAA-2000-2234.
4. Strazisar, A.J., et al., “Compressor Stall Control Through Endwall Recirculation,” Proceedings of the ASME Turbo Expo 2004, Vienna, Austria, GT2004-54295.
5. Litt, J.S., Simon, D.L., Garg, S., Guo, T.H., Mercer, C., Miller, R., Jensen, D.T., 2005, “A Survey of Intelligent Control and Health Management Technologies for Aircraft Propulsion Systems,” NASA-TM-2005-213622.

6. May, R., Csank, J., et al., “A High-Fidelity Simulation of a Generic Commercial Aircraft Engine and Controller,” 46th Joint Propulsion Conference and Exhibit, Nashville, TN, 2010, AIAA-2010-6630.
7. Short, M., Pont, M.J., “Assessment of High-Integrity Embedded Automotive Control Systems Using Hardware in the Loop Simulation,” Journal of Systems and Software, Volume 81, Issue 7, July 2008, pp. 1163-1183, ISSN 0164-1212, 10.1016/j.jss.2007.08.026.
8. Hillenbrand, M., Heinz, M., Müller-Glaser, K.D., “Rapid Specification of Hardware-in-the-Loop Test Systems in the Automotive Domain Based on the Electric / Electronic Architecture Description of Vehicles,” 21st IEEE International Symposium on Rapid System Prototyping (RSP), 2010, pp. 1-6, June 8-11, 2010.
9. Mahalik, N.P., Kiseon Kim, “A Prototype for Hardware-in-the-Loop Simulation of a Distributed Control Architecture,” IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 38, no. 2, pp. 189-200, March 2008.
10. Karpenko, M., Sepehri, N., “Hardware-in-the-Loop Simulator for Research on Fault Tolerant Control of Electrohydraulic Actuators in a Flight Control Application,” Mechatronics, Volume 19, Issue 7, October 2009, pp. 1067-1077, ISSN 0957-4158.
11. Ramalingam, S., Green, A., Lamm, P., Barnard, H. et al., “Integrated Hardware-in-the-Loop Simulation of a Complex Turbine Engine and Power System,” SAE Technical Paper 2006-01-3035, 2006.
12. Prencipe, A., “Breadth and Depth of Technological Capabilities in CoPS: the Case of the Aircraft Engine Control System,” Research Policy, Volume 29, Issues 7–8, August 2000, pp. 895-911, ISSN 0048-7333, 10.1016/S0048-7333(00)00111-6.
13. Culley, D., “Transition in Gas Turbine Control System Architecture: Modular, Distributed, and Embedded,” ASME Turbo Expo 2010, Glasgow, Scotland, United Kingdom, 2010, GT2010-23226, NASA/TM-2010-216806.
14. Naghshtabrizi, P., Hespanha, J.P., “Analysis of Distributed Control Systems with Shared Communication and Computation Resources,” American Control Conference 2009, pp. 3384-3389, June 10-12, 2009.
15. Panov, V., “Distributed Network System for Real-Time Model Based Control of Industrial Gas Turbines,” ASME Turbo Expo 2011, Vancouver, British Columbia, Canada, GT2011-45298.
16. Wandeler, E., Thiele, L., 2005, “Real-time Interfaces for Interface-Based Design of Real-Time Systems with Fixed Priority Scheduling,” Proceedings of the 5th ACM international conference on Embedded software, pp. 80-89, ACM.
17. Wandeler, E., Thiele, L., 2006, “Interface-Based Design of Real-Time Systems with Hierarchical Scheduling,” Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 243-252.
18. Lee, E.A., 2008, “Threaded Composite: A Mechanism for Building Concurrent and Parallel Ptolemy II Models,” UCB/EECS-2008-151, California University Berkeley Dept. of Electrical Engineering and Computer Science.
19. Heck, B.S., Wills, L.M., Vachtsevanos, G.J., 2009, CH. 11: Software technology for implementing reusable, distributed control systems. *Applications of Intelligent Control to Engineering Systems*, pp. 267-293.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
01-07-2013

2. REPORT TYPE
Technical Memorandum

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE
Integrated Tools for Future Distributed Engine Control Technologies

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Culley, Dennis; Thomas, Randy; Saus, Joseph

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
WBS 794072.02.03.06.02

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-18694

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITOR'S ACRONYM(S)
NASA

11. SPONSORING/MONITORING REPORT NUMBER
NASA/TM-2013-217883

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Categories: 01 and 07
Available electronically at http://www.sti.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 443-757-5802

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Turbine engines are highly complex mechanical systems that are becoming increasingly dependent on control technologies to achieve system performance and safety metrics. However, the contribution of controls to these measurable system objectives is difficult to quantify due to a lack of tools capable of informing the decision makers. This shortcoming hinders technology insertion in the engine design process. NASA Glenn Research Center is developing a Hardware-in-the-Loop (HIL) platform and analysis tool set that will serve as a focal point for new control technologies, especially those related to the hardware development and integration of distributed engine control. The HIL platform is intended to enable rapid and detailed evaluation of new engine control applications, from conceptual design through hardware development, in order to quantify their impact on engine systems. This paper discusses the complex interactions of the control system, within the context of the larger engine system, and how new control technologies are changing that paradigm. The conceptual design of the new HIL platform is then described as a primary tool to address those interactions and how it will help feed the insertion of new technologies into future engine systems.

15. SUBJECT TERMS
Turbine engine; Engine control; Engine design

16. SECURITY CLASSIFICATION OF:

a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	**17. LIMITATION OF ABSTRACT** UU	**18. NUMBER OF PAGES** 18	**19a. NAME OF RESPONSIBLE PERSON** STI Help Desk (email:help@sti.nasa.gov) **19b. TELEPHONE NUMBER** *(include area code)* 443-757-5802