Developing an Integration Infrastructure for Distributed Engine Control Technologies

Dennis Culley[1]
NASA Glenn Research Center, Cleveland, Ohio, 44135, USA

Alicia Zinnecker[2]
N&R Engineering, Parma Hts., Ohio, 44130, USA

Eliot Aretskin-Hariton[3] and Jonathan Kratz[4]
NASA Glenn Research Center, Cleveland, Ohio, 44135, USA

Turbine engine control technology is poised to make the first revolutionary leap forward since the advent of full authority digital engine control in the mid-1980s. This change aims squarely at overcoming the physical constraints that have historically limited control system hardware on aero-engines to a federated architecture. Distributed control architecture allows complex analog interfaces existing between system elements and the control unit to be replaced by standardized digital interfaces. Embedded processing, enabled by high temperature electronics, provides for digitization of signals at the source and network communications resulting in a modular system at the hardware level. While this scheme simplifies the physical integration of the system, its complexity appears in other ways. In fact, integration now becomes a shared responsibility among suppliers and system integrators. While these are the most obvious changes, there are additional concerns about performance, reliability, and failure modes due to distributed architecture that warrant detailed study. This paper describes the development of a new facility intended to address the many challenges of the underlying technologies of distributed control. The facility is capable of performing both simulation and hardware studies ranging from component to system level complexity. Its modular and hierarchical structure allows the user to focus their interaction on specific areas of interest.

Nomenclature

C-MAPSS40k	Commercial Modular Aero-Propulsion System Simulation 40k
CM	Control Model
CP	Control Platform
DC	Data Concentrator
DEC	Distributed Engine Control
DECWG™	Distributed Engine Control Working Group
ECU	Engine Control Unit (Fig. 7)
EM	Engine Model
EP	Engine Platform
HIL	Hardware-in-the-Loop
LAN	Local Area Network
OA	Operator Applications
OP	Operator Platform
NCC	Networked Component Configuration
NCM	Networked Component Model

[1] Research Engineer, Intelligent Control and Autonomy Branch, AIAA Senior Member.
[2] Control Systems Engineer, Intelligent Control and Autonomy Branch, AIAA Member.
[3] Research Assistant, Intelligent Control and Autonomy Branch.
[4] Pathways Intern, Intelligent Control and Autonomy Branch.

NPSS	Numerical Propulsion System Simulation
RTOS	Real Time Operating System
UDP	User Datagram Protocol
VCM	Virtual Component Model
VTC	Virtual Test Cell

I. Introduction

IN the past, most control systems existed as a single hardware element containing all the computational power and active electronics necessary to operate the system under control. The only items external to this federated hardware architecture were the sensing and actuation transducers, which are passive devices that provide information for use in the control algorithm and respond to control commands, respectively.[1] A more modern architecture is distributed control, where functionality is partitioned across intelligent hardware embedded throughout the system.[2] Each individual control element can freely communicate with the rest; thereby the system operates as a whole. The choice of control architecture involves a complicated trade-off between many variables. Both federated and distributed architectures are valid, representing the ends of a continuum of design choices for implementing a control system.

Less appreciated is how these design choices depend on the underlying electronics. Distributed control is only viable, in most instances, due to the wide selection, low cost, and availability of integrated electronics and microprocessor devices that are mainly targeted toward consumer applications, as well as the many design tools and software applications supporting system development. This array of choices has not always been available; hence, distributed control has been a relatively recent advancement that has become more pervasive over time. In fact, it is now typically the preferred approach to building systems.

The great advantage of a distributed system is modularity: a process of constructing systems from pre-existing functional elements. This elevates the design task, replacing many of the tedious details of constantly developing new systems from the ground up, to integrating interconnecting blocks consisting of known functions and interfaces. The effect is a more rapid and scalable design process with a more predictable performance outcome based on the *a priori* knowledge of these reusable blocks. Since the electronics in each element provide local intelligence, new opportunities are available for implementing additional features to improve performance, enable fault isolation, etc.

It is appropriate to think of electronics as the primary "material" of a control system. Like any other material, the constraints of these electronics must be properly accommodated in order for them to function reliably. Unfortunately, in some applications, the penalties involved in engineering these accommodations can outweigh any net system performance benefit. In these cases, the advantages afforded by the new technology cannot be realized. One application in which such barriers are present has been the turbine engine propulsion system. In turbine engines, the overriding need for extreme reliability (safety) and low weight (performance) in a very hostile environment defines a severely constraining set of conditions that cannot be accommodated by presently-available electronics.

Recent steps taken by the propulsion industry, through the formation of the Distributed Engine Control Working Group (DECWG™), aim to overcome many of the barriers that make distributed engine control (DEC) infeasible at the present time, especially the need for high temperature electronics. By working together, DECWG seeks to develop technologies that are presently unavailable from commercial sector electronics suppliers, and to present a unified market in which to sustain them. However, since DEC defines an architecture that affects the hardware and software technologies throughout the control system, electronics are not the only barrier. The nature of system integration, specifically how elements of control work together and assimilate with a larger system, can be drastically altered, especially when elements are acquired through a supply chain. The issue of intellectual property and how organizations interact needs to be addressed.

The propulsion industry is striking out on its own to develop a high temperature electronics capability. It is reasonable to also expect a need for new tools to support the design, analysis, and integration of high temperature-capable distributed controls. The intellectual property issues can be complicated when one considers that control element suppliers must participate in this development process. In this respect, the technology of DEC requires the development of an entire infrastructure, not only to develop specific underlying technologies, like electronics, but also to help disseminate their use throughout industry. This development is a formidable task, given the competitive environment and the need to protect intellectual property.

One solution to this problem is to develop a generic simulation from which to perform comparative studies on new technologies. This circumvents the need for proprietary information in simulation and can demonstrate relative results for design variations. NASA has developed simulation tools for this purpose in the past, which have been widely used and accepted by industry and academia.[3,4] The Commercial Modular Aero Propulsion System

Simulation 40k (C-MAPSS40k), which was developed in MATLAB/Simulink® (The MathWorks, Inc), is one such tool.[5,6] Unfortunately, these software simulations contain very little information with respect to control hardware or control architecture. Extending C-MAPSS40k to incorporate these details is a natural growth path and an express outcome of the activity presented here. It is highly desirable to simplify how specific control concepts and ideas (considered intellectual property) can be integrated into the larger control system. The modularity of distributed control architecture enables the protection of proprietary issues while leveraging standards in communications and networking to facilitate their integration. NASA is interested in this integration process relative to distributed technology precisely because it is vital for the development of stable and robust control systems and for ensuring the long-term growth of the technology.

The NASA facility described in this paper was developed as a virtual test cell with the flexibility to focus on one or more aspects of control technology. The system is modular and hierarchical, so as not to constrain future growth, and extensible and adaptable, so as not to be limiting in its capability. A top-down description of the facility's development will be used to explain the range of system capabilities, focusing on the interfaces of most interest to the typical user.

Section II will discuss motivation for the system design decisions, focusing on its evolution and the initial goals around which the system was developed. It will also explore what other capabilities are possible and how they can be achieved. Sections III to V will describe the three main operational modes of the virtual test cell. The top-level (Level 0), describing the overall structure of the simulation that can be used for testing engine system performance, is described in Section III. Section IV describes the interaction between the components of the virtual test cell (Level-1), focusing on the integration of an engine model, while Section V addresses the internal modular structure of the control system model (Level-2) for investigating control architecture. A summary is provided in Section VI followed by acknowledgements.

II. Motivation

The concept of a facility for control system simulation with hardware-in-the-loop (HIL) capability grew from a desire to focus on collaboration as a primary means of ensuring a growth path for the continuous maturation of distributed control technology. The scope of the technologies can range from embedded electronics, to control elements, to networks, to entire control systems; therefore, the facility needs to be able to adjust to that focus as well. Additionally, new control technology is often proprietary; therefore, concerns regarding intellectual property need to be addressed in order to advance the development of new ideas.

Figure 1 depicts the basic system structure: an open-loop engine model (EM) in closed-loop with the control model (CM). To complete the scenario, an operator application (OA) provides additional inputs normally provided through the airframe or environment. In the interest of flexibility, the CM is distinct from the EM to the extent that the models can be hosted on separate computational hardware during simulation. There are multiple advantages to this approach. First, it requires the establishment of well-defined interfaces between the major systems. Second, it forces the simulations to become asynchronous to each other, reflecting the actual open-loop configuration of the engine and control system. Third, both the engine and control models follow independent development paths that allow them to take advantage of advances in modeling techniques and the computational hardware they run on. Fourth, decoupling the CM and EM in this way opens up the possibility for incorporating other engine simulations as a "black box," thus keeping the implementation details hidden.

The dynamic engine model in C-MAPSS40k is likely to be familiar to potential users of the facility and is useful for being able to validate the new system's operation. Therefore, C-MAPSS40k was a natural and convenient basis for the new system's development. However, the ability to accurately simulate the impact of control system hardware and architecture on engine system performance requires a control model with higher fidelity than that available in C-MAPSS40k.

To begin these improvements, it is understood that all interaction between the engine and the controller occurs through the sensors and actuators, which are elements of the control system. This interaction is represented by a flow of digital data, occurring at a specified control

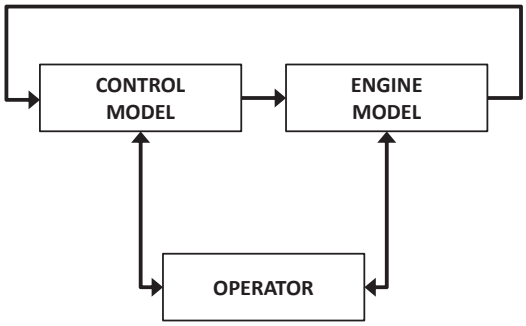

Figure 1. The basic structure of the virtual test cell is an engine model in closed-loop with a control model, where the interfaces are well-defined. The operator provides additional input/output to complete the engine system.

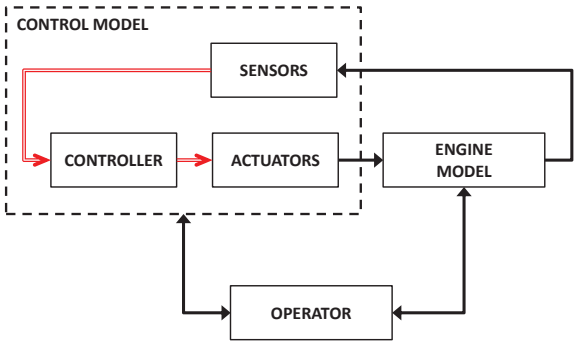

Figure 2. The control system is modeled as a collection of separate elements. The interfaces between the elements define the data flow within the system. E.g., data from the engine model first enters the control system model (dotted line) and is then distributed to the appropriate element model (sensor). Control elements communicate via the network models, depicted as red lines.

interval, between the engine and control models. Within the CM, data must be collected from/distributed to the sensor/actuator elements in a unidirectional pattern. These concepts are depicted in Figure 2, suggesting a functional decomposition of the top-level control model based on hardware elements.

Further complexity exists as a reflection of the importance of the network-connected control elements in the distributed control architecture. From a modeling perspective, the network is one of the greatest unknowns because any number of bus structures and protocols may be used within a given system. These system control elements would normally exist in hardware as asynchronous devices running at various rates. Therefore, one of the major functions of the control networks is to synchronize the control system through dataflow on the networks. The specific nature of this interaction depends on the characteristics of the particular network protocol.

Incorporating the increased fidelity of control models with networked control elements is a significant leap. Validating these models requires an ability to work with hardware and real-time determinism. If any of these control elements were to exist as actual hardware, as in HIL testing, each element would require access to both the engine model data (as an analog signal interface) and the control communication network interface. This is a significant complication over previous HIL facilities that only interface with an engine control unit.

The following sections of the paper will focus on the structure of the facility known as the Virtual Test Cell (VTC). The key to understanding user interaction is to follow the interfaces, which allow the system to be configured and operated as required.

III. Level-0: The Virtual Test Cell

The C-MAPSS40k software simulation was developed to aid in the design of new engine control strategies, providing realistic, closed-loop transient responses that do not violate engine safety and operability limits. In contrast, the purpose of the VTC is to aid in the development of new engine control hardware architectures. Thus, the VTC provides an infrastructure for developing detailed simulation of smart hardware elements, interconnected over digital communication networks, while interfacing with the state variables of an engine simulation. In effect, the VTC uses C-MAPSS40k as a vehicle to develop this infrastructure. However, there is no technical issue that should prevent the VTC from implementing any engine system simulation by following a similar model structure.

Level-0 defines some of the possible system configurations for the VTC to use while performing investigations of control technologies. These configurations may vary based on the needs for performing: conceptual studies in simulation, to more detailed studies connected to an external engine simulation, to high fidelity investigations with real-time hardware. Figure 3 is used to establish a frame of reference by describing some of the methods used to create these variations. Each configuration represents an extension of its predecessor.

Figure 3 also introduces some new terminology to help track these developments. So far, the three components of the facility have been discussed as the models or applications that perform the functions of the engine, the control system, and the operator. These are known as the EM, CM, and OA, respectively. As the facility is extended with additional capabilities, it becomes useful to make additional distinctions. A model or application running on computational hardware is known as a platform. Therefore, an Engine Platform (EP) consists of an engine model (EM) running on computational hardware. Similarly defined are the Control Platform (CP) and the Operator Platform (OP).

The following subsections describe representative configurations of the VTC and why they were developed. These are identified as the Virtual Component Model (VCM), the Networked Component Model (NCM), and the Real-Time VTC.

A. The Virtual Component Model

The VCM modifies the original C-MAPSS40k software by completely segregating the engine plant model, its native controller, and an operator interface into three model reference blocks within Simulink™. This basic improvement to the C-MAPSS40k configuration precisely defines the interface definition for the three main structures in the VTC and is shown at the top of Figure 3. The model reference blocks also simplify the further decomposition of each model allowing referenced models within them to be developed independently, an important consideration for future collaboration. Note that C-MAPSS40k uses the term "controller" to represent the entire control system because it does not discern any of the effects of a hardware control architecture in its lumped model. It is only concerned with function.

It is most intuitive to visualize the interfaces of the three major components by examining the dataflow from the perspective of each block. Figure 4 is derived from Figure 1 and knowledge of C-MAPSS40k. The nature of the data is categorized based on its purpose. This is further clarified in Table 1, which shows the actual variables used in C-MAPSS40k and describes their source and destination. The engine block produces data describing the state of the engine, as well as its health and information not available to the control system. The control block produces data describing the manipulation of engine control surfaces, as well as information internal to the control system, such as sensor outputs and estimates of engine system state that cannot be measured. The operator block produces data that would normally originate from the airframe, as well as variables to modify engine or control system operation.

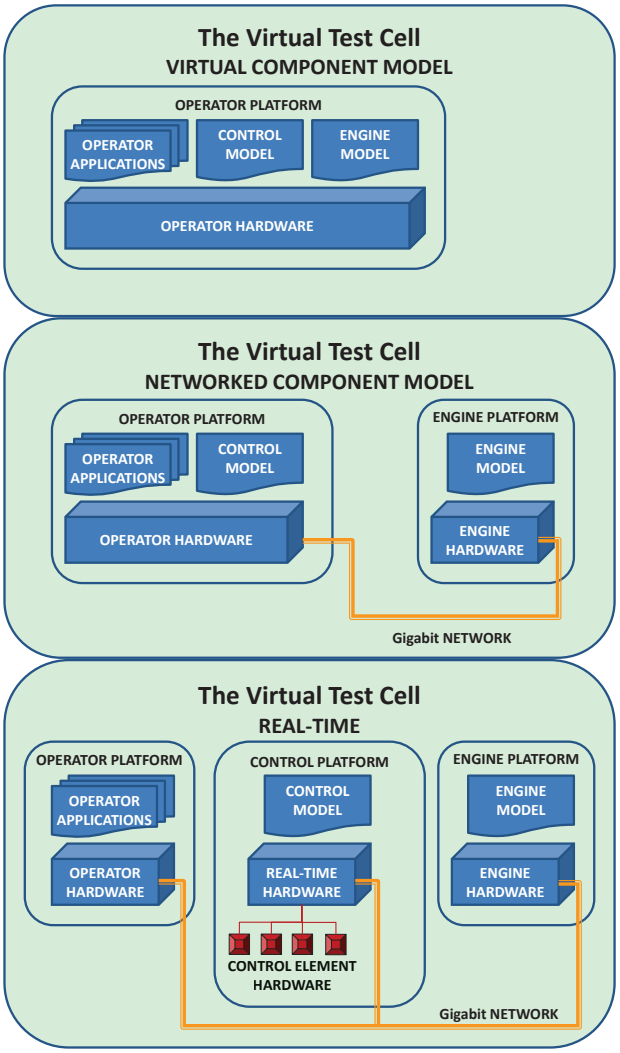

Figure 3. The architecture of the Virtual Test Cell is based on strong interface definitions that delineate engine system models along natural system interfaces. This allows the system configuration to change while maintaining model compatibility, thus aiding collaboration. The same models will run in any configuration as long as they can be compiled, which is required for real-time operation.

The VCM is a basic modification of C-MAPSS40k, but it is critical in extending the concept of modularity to enable the development of the more detailed control system configurations necessary for distributed engine control. The interfaces and modeling techniques used in the VCM configuration can be implemented on a single computer to simplify collaboration. The concept can also be extended as described next.

B. The Networked Component Model

The NCM further separates the EM, CM, and OA by enabling them to run on separate computational platforms. There is utility in having an ability to operate with engine models other than the native C-MAPSS40k. To work with another EM, one must obtain an open source model or have a means to protect the intellectual property of a proprietary model. The latter can be time consuming and still presents substantial problems to ensuring confidentiality, which will likely preclude it from happening. However, if the engine plant model can be removed to a separate computational platform it is more apt to occur, especially if that machine can be physically removed for protection.

Engine and control system models are also very different. Engine models are typically derived from high fidelity steady state models like the Numerical Propulsion System Simulation (NPSS), or from empirical data obtained during engine testing. Engine models are always reduced order models that emphasize speed and transient performance over absolute accuracy. In fact, C-MAPSS40k is a 0-dimensional model that operates many times faster than real-time. A 0-dimensional model contains no volume dynamics and cannot accommodate ad hoc additions to the corresponding control system because it cannot provide or accept data not captured by the model.

The NCM routes the data interface defined in the VCM configuration through a fast, low cost communication medium based on gigabit Ethernet. This makes the model data generated from each machine available to the others rapidly, so that there is negligible impact on the overall computational time and accuracy. The middle diagram of Figure 3 depicts only two platforms being networked together, as this can be done selectively.

One artifact of this particular implementation is the presence of a fixed time delay of one control interval between the engine and control simulations. The control interval is defined as the time step (fixed or variable) used by the engine and control simulations to evaluate the models. Although data is transferred at every control interval, control commands sent to the engine are one control interval old, as they were calculated based on the engine state at the previous time step. This is demonstrated by comparing the data transfer sequence for a single sequential model to that of the networked version, as shown in Figure 5.

Table 1. C-MAPSS40k variables and their source and destination assignments within the VTC structure.

		SOURCE		
		Control	Engine	Operator
DESTINATION	Control		P_actual (9) T_actual (7) W_actual (15) N_actual (2)	alt Mach EPR_dmd Nf_dmd control mode disable_limits WoW
	Engine	Wf_act VSV_act VBV_act		alt Mach dTamb health params (13) time
	Operator	P2 P25 Ps3 P50 T2 T25 T30 T50 Nf Nc EPR_sens Wf_dmd WfPs3 control_states (8) Wf VSV VBV	Fdrag Fnet Fgross Nf_dot Nc_dot SM_LPC SM_LPC_avail SM_HPC SM_HPC_avail Smdebit (3) Tmetal (5) dTMetal (5) Tau (3) x (5) sim_iter sim_error	

The networked technology has been demonstrated on a three-machine implementation of the NCM connecting the EM, the CM, and the OA. The system has shown no loss of data and no performance impact relative to the baseline C-MAPSS40k implementation and almost no degradation in speed when running faster than real-time.

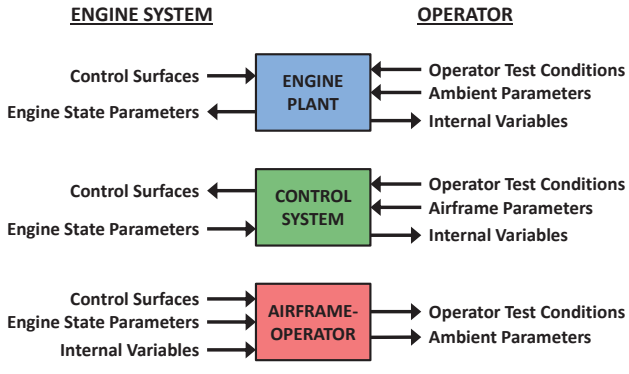

Figure 4. Major system level interfaces from the model perspective.

The Networked Component Model is the second step in the development of the VTC. Its value is in the added flexibility it brings to configuring the system as well as the increase in processing power to accommodate more complex models. An important feature of this configuration is an ability to operate the multi-machine system from one location using compiled models for engine and control simulation. The impact is to reduce the overall complexity and the need for licensing and installing multiple copies of application software. One potential disadvantage is the inability to compile certain simulation blocks from MATLAB/Simulink toolboxes.

C. Real-Time Virtual Test Cell

The Real-Time Virtual Test Cell further extends capability by operating the engine simulation in a

deterministic environment. In this configuration, models are compiled for a target computer running a real-time operating system (RTOS). This configuration is required for HIL and is shown at the bottom of Figure 3.

In an RTOS, the kernel is small, efficient, and guaranteed to complete its tasks in a deterministic fashion. The user then has the capability to configure the priorities of the remaining processes to insure the determinism of the overall application. By operating the control simulation in real-time, the user can establish the exact order, dependencies, and timing of the element models during runtime. This will be more fully explained with an example in Section V, looking at the internal operation of the VTC component models.

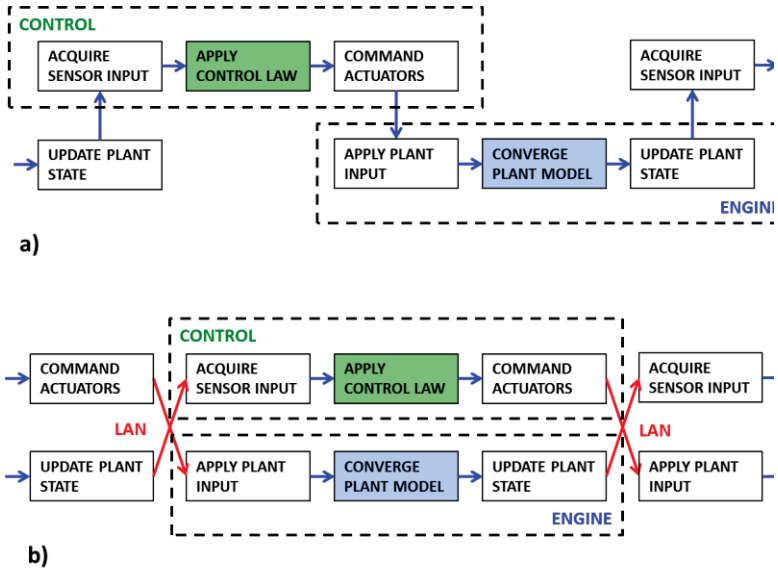

Figure 5. a) Baseline C-MAPSS40k sequential model. b) Operating the control system model and engine plant model on separate machines over a local area network (LAN). The models operate in parallel but are skewed by one control interval.

While the Real-Time VTC requires additional steps to operate a simulation, it uses the same source models developed and run in previous configurations. This continuity of environment is an intentional means of enhancing collaboration, by extending a capability for control development from concept to hardware validation.

IV. Level-1: Component Interaction

In the previous section, it was observed that creating strong interface definitions between components allowed a great deal of flexibility in the configuration of the VTC. A model of an engine system can be segregated into three components - the engine, the controller, and the operator – and the simulation run from one or more machines by transferring data over these interfaces. In this section, additional detail is provided about the mechanism used to simultaneously operate parts of the engine system simulation on more than one machine.

The basis of this capability is derived from Ethernet, which is universally available on every general purpose computer. Ethernet is actually a family of networking standards that greatly simplifies the transfer of digital data at high bit rates. The objective is to transfer the data produced by each model to the corresponding model consuming the data. In the context of the system operation, only current data is relevant. The transfer occurs at the start of every control interval, thus it also serves to synchronize the models according to the time base of the initiator.

Two major details are necessary for this to occur successfully. The information must be transferred within a time span that is small with respect to the control interval. In C-MAPSS40k, the control interval is 15 milliseconds and the information transfer is completed in less than 1 millisecond. Generally, “small” is considered to be less than 10% of the interval. However, the implication is that whatever time is used to transfer information is time not available for the evaluation of the model that consumes the data. The overriding concern is that this transfer mechanism does not impact the performance of the simulation.

A second detail is that the network used to transfer this data must ensure the data is not lost in the process because the data is time critical. The Ethernet protocol used is User Datagram Protocol (UDP), which has very low latency but implements no check on ensuring transmitted data has been properly received. Since the Ethernet physical layer is widely accommodating and imposes only minimal constraints on users, protection of the data transfer must be a design consideration. This requirement is accomplished by implementing a private local area network to eliminate competition for bandwidth. Data protection is also enforced by the data transfer sequence built into the models.

Figure 6 shows the sequence of data transfers that occur between the three components of the system during simulation. The sequence originates from the control component model at the beginning of a new control interval, so that the CM determines the time base of the entire system. The accuracy of the time source is consequential when operating in real-time. In the Real-Time VTC configuration, the CP has the only real-time clock.

A datagram is a collection of data organized into a single packet for transfer over the network. The six datagrams depicted in Figure 6 can be related to Table 1, where the sequence occurs as follows:

1. Actuator outputs from the CM are transferred to the EM.
2. In response, the EM provides the engine state parameters for the sensors located in the CM.
3. The CM initiates a data transfer to the OA consisting of the sensor outputs, internal controller variables, and actuator outputs.
4. In response, the OA provides to the CM updates to the airframe/pilot parameters and the operational mode of the controller.
5. The OA also provides similar data to the EM regarding ambient conditions and engine health.
6. The EM responds to the OA with its state parameters and other internal parameters.

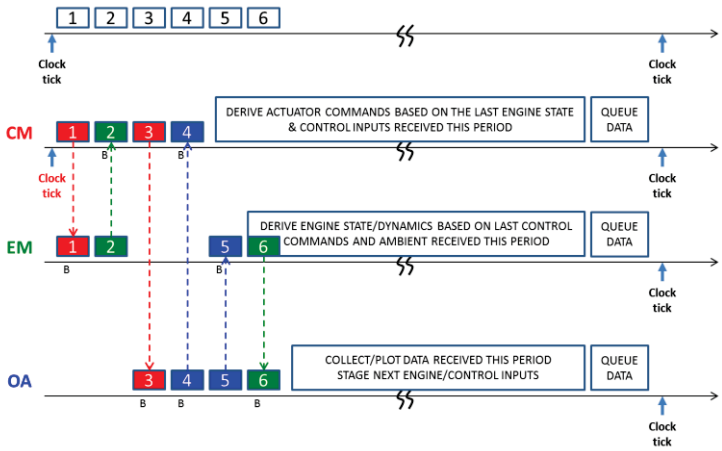

Figure 6. The information transferred between models follows the interface description shown in Table 1. Between the models, this occurs as a sequence of six datagrams where each model sends two and receives two. The sequence is initiated by the system clock of the computer running the control simulation. When the datagrams are sent over a network, the transfer serves to synchronize the entire system.

At the conclusion of this sequence, each model is evaluated simultaneously, queuing datagrams, and waiting for the next control interval. The function and implementation of this high speed data transfer process is described in Aretskin-Hariton.[7] The purpose of this data transfer process is not to change the function of the three models, but to provide additional flexibility in the configuration used to operate the models. This interaction has been demonstrated on a single machine configuration as well as a three-machine configuration running separate models.

V. Level-2: Subsystem Interaction

The interfaces described in Level-2 are concerned with the internal structure and configuration of the three component models and the approach used in developing them. Again, the control system design is the primary research focus so the discussion is restricted to the CM. Recall that in a distributed control system, individual hardware control elements operate independently and asynchronously by virtue of their embedded processing components. The hardware control system is effectively a system of these asynchronous systems that becomes synchronized through the communication networks. The control system model should be viewed in a similar manner to properly capture these hardware effects.

The methodology to accomplish this modeling framework is being developed and is partially described in Zinnecker.[8] Essentially, any control element can be decomposed into primitive functions and interfaces and the parts collected into libraries. These primitive functions may then be used to compose new intelligent control element models per the template described in IEEE-1451, the Smart Transducer Interface Standard.[9] Smart control element models so constructed provide the modular plug-in blocks to develop entire control system models when connected through models of the network.

A simple example is used to convey the importance of developing interface definitions while modeling distributed systems. Potentially, there are unlimited variations on how a distributed control architecture can be assembled. In the arbitrary example shown in Fig. 7, two network busses are constructed; on the left (purple blocks) designating the low–speed spool, and on the right (red blocks) designating the high-speed spool. The networks serve to connect the smart devices to the controller through a bridge, or data concentrator (DC).

Figure 8 describes this hardware architecture as a process flow. Each control element exists as a separate process in the model that connects to other control elements through control network models, Nci and Nxi. As the data flow progresses from sensors to the controller and back to the actuators it must pass through the network models. The network models require the element control models to conform to an interface standard. The network models also

introduce delays and enforce a particular ordering of the dataflow. Finally, the network models introduce new failure modes into the control system.

The importance of evaluating these DEC models on a real-time platform will become more apparent as the control architecture becomes more complex. Notice in Fig. 8 that a network is inserted between each control function, therefore the effect of the networks become cumulative over the course of a control interval. The networks order the data transfer process, as well as affect the precision of the data transferred; details that can easily be omitted from simulations. In the Real-Time VTC configuration, the user has precise control over the execution and interaction of individual control element processes that are unavailable in non-real-time systems. This control can more accurately represent the operation of a hardware system and is, in fact, required for HIL.[10]

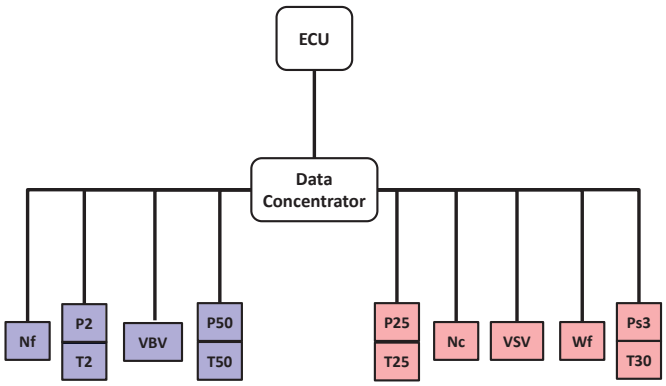

Figure 7. The C-MAPSS40k control system expressed in an arbitrary distributed control architecture. Two busses are created connecting smart elements. A data concentrator (bridge) then funnels the data along a single connection to the controller.

VI. Summary

The Virtual Test Cell (VTC) is a simulation facility with real-time hardware-in-the-loop capability. Its primary objective is to help improve the understanding of the effect of distributed control architecture on a turbofan engine system. The VTC is part of a development infrastructure designed to promote and enhance collaboration to help ensure distributed engine control technology is sustained in the future.

The development of the VTC is primarily based around the creation of and adherence to interface standards, which allow organizations to collaborate as various control technologies progress from concept to hardware. The interface standards begin with the approach to modeling control elements and how they interconnect as a system through the control networks. The interface standards encourage modularity of control functions, which allows the models to be quickly and easily reconfigured and individual elements to be upgraded.

The concept of modularity also extends to the development of the VTC as a hardware facility. The facility operates in a modular fashion using three distinct components: the engine, the control system, and the operator. By adhering to an interface definition between these components, the VTC can exist in a variety of system configurations. These configurations are intended to further cooperation in controls technology development.

Distributed engine control introduces very significant changes to engine control system technology, including how control systems are integrated and how failures may occur and affect the system. It is anticipated that the facility and methodologies described in this paper will aid in the development of distributed engine control technology, especially as it relates to the modeling of various control networks and the embedding of new value–added functionality into smart control elements.

Acknowledgments

The authors wish to thank the contributions of John McArthur and Oran (Bud) Watts of Rolls-Royce Corporation for their work in implementing the initial model reference version for C-MAPSS40k and the subsequent development of the preliminary models for real-time applications.

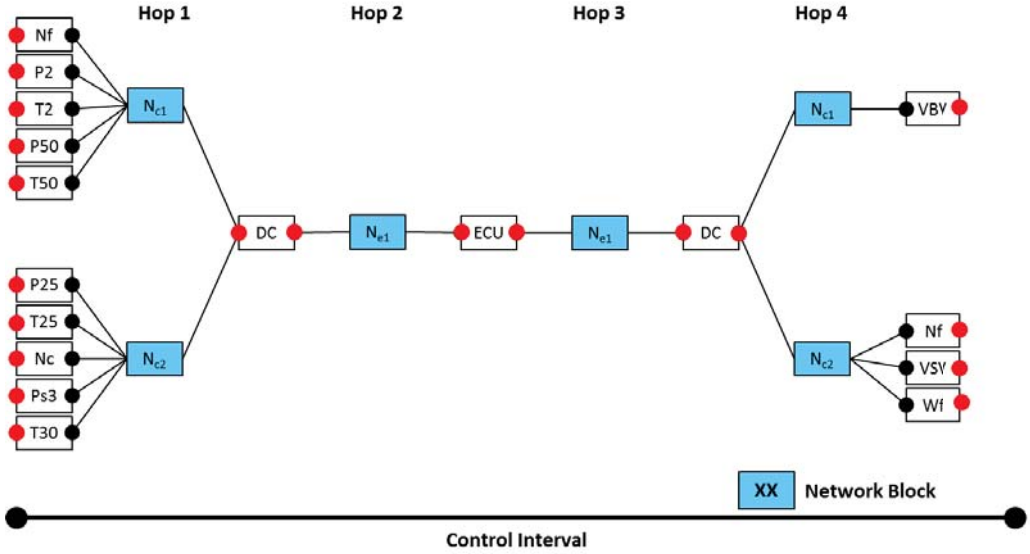

Figure 8. The arbitrary architecture described in Figure 7 is expressed as a process flow. The hardware architecture introduces effects to dataflow because of their connection and ordering on the control network. This is a very simple example to illustrate the potential impact network communication may have on the control system. Network blocks are designated in blue.

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.] Culley, D., "Transition in Gas Turbine Control System Architecture: Modular, Distributed, and Embedded," *ASME Turbo Expo2010: Power for Land, Sea, and Air*, Vol. 3, Glasgow, Scotland, United Kingdom, June 2010, pp. 287–297.

[3.] Parker, Khary I., Guo, Ten-Heui, "Development of a Turbofan Engine Simulation in a Graphical Simulation Environment," NASA TM-2003-212543, August 2003.

[4.] Frederick, Dean K., DeCastro, Jonathan A., Litt, Jonathan S. "C-MAPSS User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS),"NASA TM-2007-215026, October 2007.

[5.] 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

[6.] Csank, Jeffrey, Ryan D. May, Jonathan S. Litt, and Ten-Huei Guo. "Control design for a generic commercial aircraft engine." In Proceedings of the 46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Hartford, CT. 2010.

[7.] Aretskin-Hariton, E., Zinnecker, A.M., Culley, D., "Extending the Capabilities of Closed-Loop Engine Simulations using LAN Communication," AIAA Propulsion and Energy 2014, Cleveland, Ohio, (submitted for publication)

[8.] Zinnecker, A., Culley, D., Aretskin-Hariton, E.D., "A Modular Framework for Modeling Hardware Elements in Distributed Engine Control Systems," AIAA Propulsion and Energy 2014, Cleveland, Ohio, (submitted for publication)

[9.] Lee, K.C., Kim, M.H., Lee, S., Lee, H.H., "IEEE-1451-based Smart Module for In-vehicle Networking Systems of Intelligent Vehicles," IEEE Transactions on Industrial Electronics, Volume 51, Issue 6, Dec. 2004 Page(s):1150 – 1158.

[10.] McArthur, John, Travis Boehm, Bobbie Hegwood, and Oran Watts. "Decentralized Engine Control System Simulator," ASME Turbo Expo 2013: Turbine Technical Conference and Exposition, pp. V004T06A016-V004T06A016. American Society of Mechanical Engineers, 2013.