Extending the Capabilities of Closed-Loop Distributed Engine Control Simulations using LAN Communication

Eliot D. Aretskin-Hariton[*]
NASA Glenn Research Center, Cleveland, Ohio, 44135

Alicia M. Zinnecker[†]
N&R Engineering, Cleveland, Ohio, 44135

and

Dennis E. Culley[‡]
NASA Glenn Research Center, Cleveland, Ohio, 44135

Distributed Engine Control (DEC) is an enabling technology that has the potential to advance the state-of-the-art in gas turbine engine control. To analyze the capabilities that DEC offers, a Hardware-In-the-Loop (HIL) test bed is being developed at NASA Glenn Research Center. This test bed will support a systems-level analysis of control capabilities in closed-loop engine simulations. The structure of the HIL emulates a virtual test cell by implementing the operator functions, control system, and engine on three separate computers. This implementation increases the flexibility and extensibility of the HIL. Here, a method is discussed for implementing these interfaces by connecting the three platforms over a dedicated Local Area Network (LAN). This approach is verified using the Commercial Modular Aero-Propulsion System Simulation 40k (C-MAPSS40k), which is typically implemented on one computer. There are marginal differences between the results from simulation of the typical and the three-computer implementation. Additional analysis of the LAN network, including characterization of network load, packet drop, and latency, is presented. The three-computer setup supports the incorporation of complex control models and proprietary engine models into the HIL framework.

Nomenclature

C-MAPSS40k	Commercial Modular Aero-Propulsion System Simulation 40k
CP	Control Platform
DEC	Distributed Engine Control
ECU	Engine Control Unit
EP	Engine Platform
EPR	Engine Pressure Ratio
FADEC	Full Authority Digital Engine Control
Gbps	Gigabits per second
HIL	Hardware-In-the-Loop
IP	Internet Protocol
LAN	Local Area Network
lb_f	pounds force
Mbps	Megabits per second
ms	milliseconds
OP	Operator Platform

[*] Research Engineer, Control Systems, eliot.d.aretskin-hariton@nasa.gov
[†] Control Systems Engineer, alicia.m.zinnecker@nasa.gov, AIAA Member
[‡] Research Engineer, Control Systems, dennis.e.culley@nasa.gov, AIAA Senior Member

pps	pounds per second
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
W_f	Fuel flow rate
σ	Standard Deviation

I. Introduction

DISTRIBUTED Engine Control (DEC) has been used in motor vehicles for over a decade. However, the application of distributed control techniques to gas turbine control is still in an early phase of development. The primary goals of DEC are to enable new control technologies that decrease fuel burn, life cycle costs, and downtime for maintenance[1] while alleviating the constraints placed on control hardware by engine systems. The Engine Control Units (ECUs) that control modern aircraft engines communicate with sensors and actuators (network nodes) using point-to-point analog signals in a star topology.[2] This is a weight-expensive method of ensuring communication between nodes. The system containing the sensors, actuators, the ECU, and all the cabling is termed Full Authority Digital Engine Control (FADEC). The extensibility of the FADEC is limited due to rigorous certification requirements that may force the redesign and recertification of the FADEC if a single sensor is added to the system. Furthermore, each FADEC is custom-designed, for a particular engine type.[3] DEC seeks to advance the state-of-the-art in aircraft engine controls by using a digital communication network with a more robust network such as a mesh, ring, or bus topology. This will lead to the development of FADEC systems with greater extensibility and higher capacity for upgrades.

The Hardware-In-the-Loop (HIL) test bed, at NASA Glenn Research Center, emulates the operator- controller-engine interaction and provides a means for demonstrating DEC. The HIL test bed contains engine and controller models based on the Commercial Modular Aero-Propulsion System Simulation 40k (C-MAPSS40k) developed at NASA Glenn Research Center. C-MAPSS40k is a closed-loop, 0-D simulation of a twin-bypass 40,000lb thrust-class engine, with a set-point controller and limit regulators.[4,5] The typical C-MAPSS40k engine and controller are simulated on the same computer, along with a system that provides operator, airframe, environmental, and health inputs to the model (Figure 1). The user interface is on the same computer and provides a system whereby flight profiles and engine health parameters can be loaded into the simulation. This setup is referred to as “the typical C-MAPSS40k setup” throughout the paper.

The shared computational resources required for such a setup may pose problems in the future, if the computational complexity of the engine or control system models is significantly increased. To alleviate this, the HIL test bed contains three platforms, each dedicated to running one of the three main parts of the C-MAPSS40k system: the operator platform, the control platform, and the engine platform (Figure 2). These three platforms are connected over a dedicated Local Area Network (LAN). They communicate using User Datagram Protocol (UDP) over Gigabit-Ethernet. The typical C-MAPSS40k simulation runs the models sequentially. By splitting the C-MAPSS40k simulation across three computers, the models run in parallel. Additionally, the hardware for each platform can be upgraded independently, which supports future control and engine model expansions. Decoupling the models simplifies the process of including different engine models in the HIL test bed.

Other companies have similarly implemented UDP LAN systems in place of reflective memory systems for other applications.[6] Reflective memory is a proprietary technology developed in the 1980s by Encore Computer Co. It provides a shared memory set that contains up-to-date data from multiple sources, allowing multiple computer systems to synchronize information at high speeds.[7] This enables real-time communication between multiple connected systems to occur in a deterministic fashion. The main drawback of reflective memory is that it requires expensive proprietary hardware. In the HIL test bed, changes in the state of one machine are sent to the other two machines in a deterministic fashion using commonly available 1Gbps UDP Ethernet, achieving the same functionality as reflective memory without the expense of custom hardware (Figure 2).

In this paper, the software and hardware used for setting up the UDP LAN communication between the engine, control, and operator platforms are detailed in Section II, along with details on how the C-MAPSS40k model is adapted to the networked implementation. Results comparing the typical C-MAPSS40k setup and the new three-computer implementation, and an exploration of the LAN network robustness, are provided in Section III. Section IV summarizes the conclusions and presents suggestions for future work.

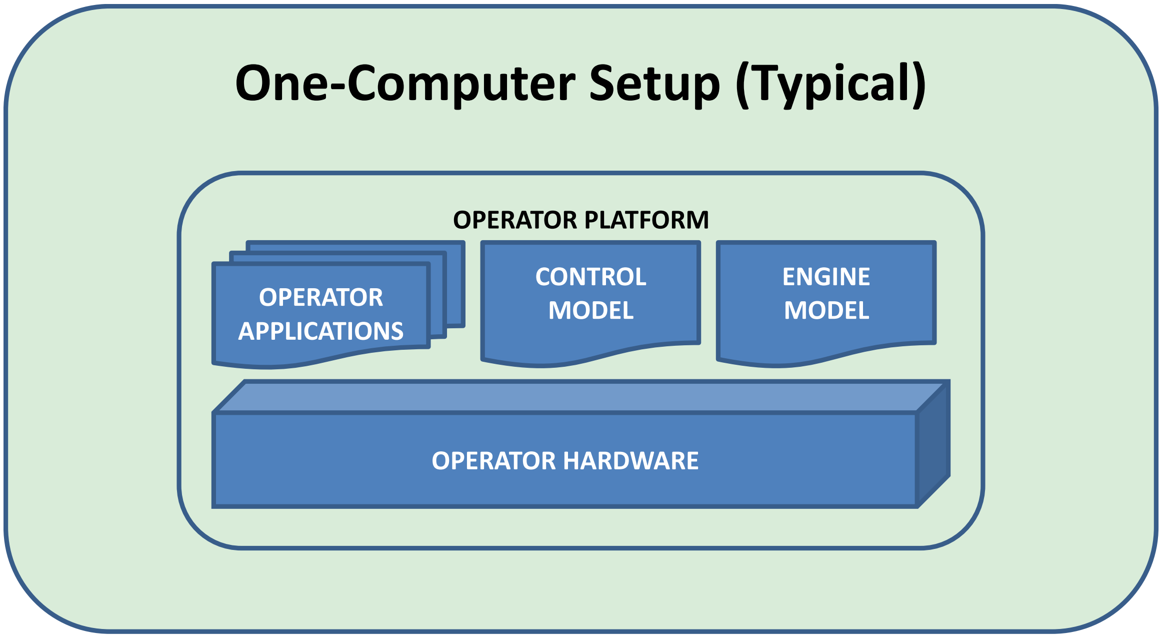

Figure 1. Typical C-MAPSS40k setup with all models implemented on a single computer

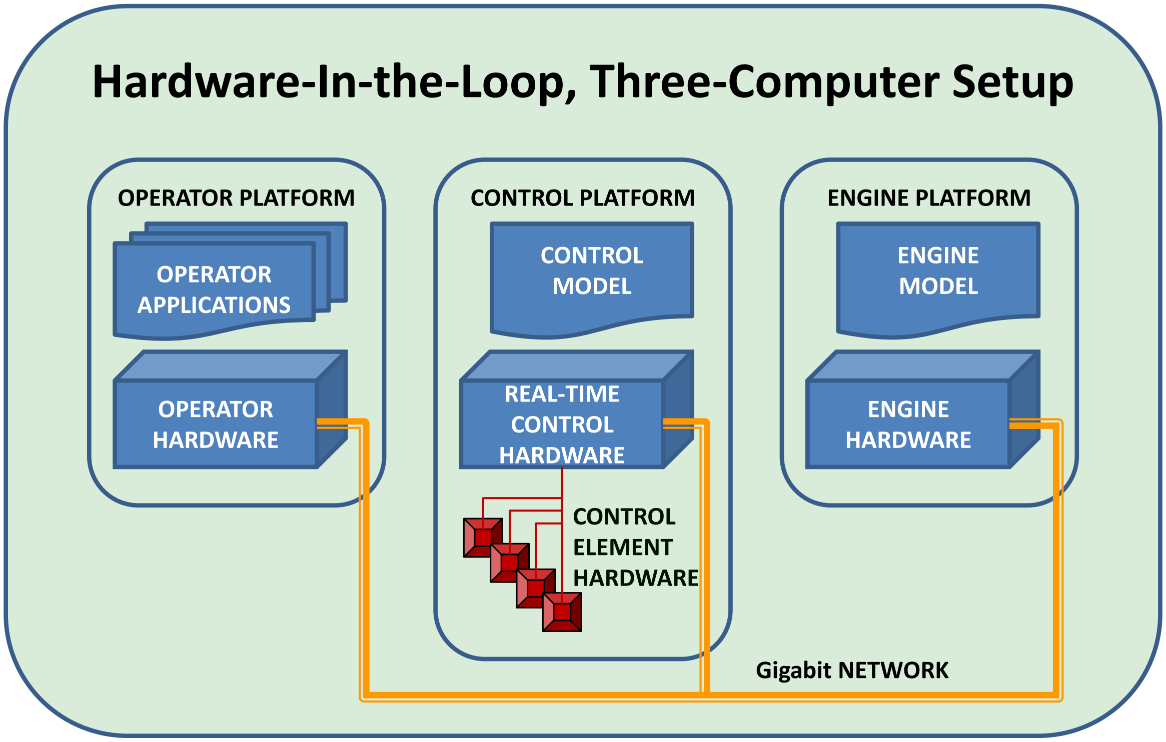

Figure 2. Three-Computer C-MAPSS40k setup with three networked computers

II. Implementation

The HIL system, as currently implemented, resides on three computer platforms running on a Windows® operating system (Microsoft Corporation). The term platform is used here to refer to the combination of computer hardware and model (or application software). The operator platform is the system containing the operator applications and the operator hardware, which is the physical computer system. The operator applications include programs to simulate pilot input, airframe characteristics, and ambient environment information. The control platform encompasses the control model and the control hardware, which may include physical control elements like sensors and actuators in addition to the computer on which the model resides. The engine platform is made up of the engine model and the engine hardware. All computer hardware is connected with Category 6 cable to a 1-Gigabit switch to form a dedicated LAN for the three platforms. This dedicated network is not connected to the internet, and there are no wireless connections anywhere in the test bed. The only data packets sent on the LAN are from the HIL simulation[§]. A full-duplex Gigabit switch is used to route communications between the three platforms. The use of a switch minimizes the possibility of packet collisions.

The operator applications and control model are implemented in MATLAB/Simulink® R2011b (32-bit) (The MathWorks, Inc.). The operator application communicates directly with the control model using the TCP/UDP/IP Toolbox for initial configuration and s-functions for transferring data during simulations. The TCP/UDP/IP Toolbox functions are available on the MATLAB file exchange.[**] The s-functions, which were written in-house, utilize Windows Sockets 2.[††] These tools enable UDP communication through specified ports. The engine model is an executable file compiled on the operator platform and runs on a low-cost laptop without an installation of MATLAB/Simulink. The engine model contains custom-built s-functions, which open local UDP ports to communicate engine state information to the control model and the operator applications.

It is expected that, as engine model fidelity improves, the required computing power to ensure real-time simulation capabilities will increase. This is one of the primary reasons that the HIL test bed is designed around three hardware elements. An important secondary result of this implementation is that it allows an engine model to be a self-contained executable. Any company with an engine model that can produce the requisite UDP communication streams can provide the engine platform, for integration with the test bed. This protects the intellectual property of the engine manufacturers while encouraging a collaboration with NASA on the development of next generation engine controls technology.

III. Results

After constructing the three-computer HIL system, it is necessary to verify that the presence of the LAN does not negatively affect the results for a sample simulation. Additionally, the capabilities of the LAN network to connect the three-computer C-MAPSS40k simulation, is investigated in order to understand its limitations.

A. Verifying the system setup through simulation

This section presents the results of a test case executed on both the typical and the three-computer C-MAPSS40k systems. To set up the test, a flight profile was loaded into the operator application, which commands power lever angle, altitude, Mach number, and temperature. For this test, the profile commanded a series of small steps in power lever angle at sea-level static conditions (0 ft altitude, 0 Mach, standard-day temperature). This is a standard profile that may be used to test the tracking ability of the controller. The Engine Pressure Ratio (EPR), net thrust, and fuel flow rate (W_f) profiles from the test are compared in Figure 3. The percent error shown in Figure 3 is calculated by taking the difference between output parameters of the typical and the three-computer simulation. Inspection of the results shows that, at less than 1 percent (%), net thrust exhibits the largest percent error between the simulations. The largest errors are present in the initial transient, and are related to a difference in initial conditions. The engine model is compiled with a default initial condition and is therefore allowed 20 seconds to trim to the simulation initial conditions before the flight profile begins (not shown in Figure 3). However, the typical C-MAPSS40k implementation uses a steady-state solver to determine the initial conditions based on the initial values set in the flight profile. Because the steady-state solver has a smaller error tolerance than the iterative solver used during simulation, there is a small discrepancy in initial conditions after the trim period resulting in a startup transient, which can be seen in the first 15 seconds of Figure 3. After 15 seconds, the net thrust error between the two simulations decreases to 4.7e-4 % and remains there for the rest of the simulation. Thus, with very small average error after 15 seconds of simulation, it appears that the presence of the LAN in the HIL test bed does not have a significant effect on the simulation of C-MAPSS40k.

[§] The LAN discussed here carries data involved in the closed-loop control between the engine and control system models and does not function as a ‘control network’ for the purposes of Hardware-In-the-Loop testing. In the HIL system, the control network is a separate network, which is a subcomponent of the control platform and connects sensors and actuators to the control model (Figure 2).

[**] TCP/UDP/IP Toolbox 2.0.6: http://www.mathworks.com/matlabcentral/fileexchange/345-tcpudpip-toolbox-2-0-6

[††] Windows Sockets 2 application programming interface:
http://msdn.microsoft.com/en-us/library/windows/desktop/ms740673(v=vs.85).aspx

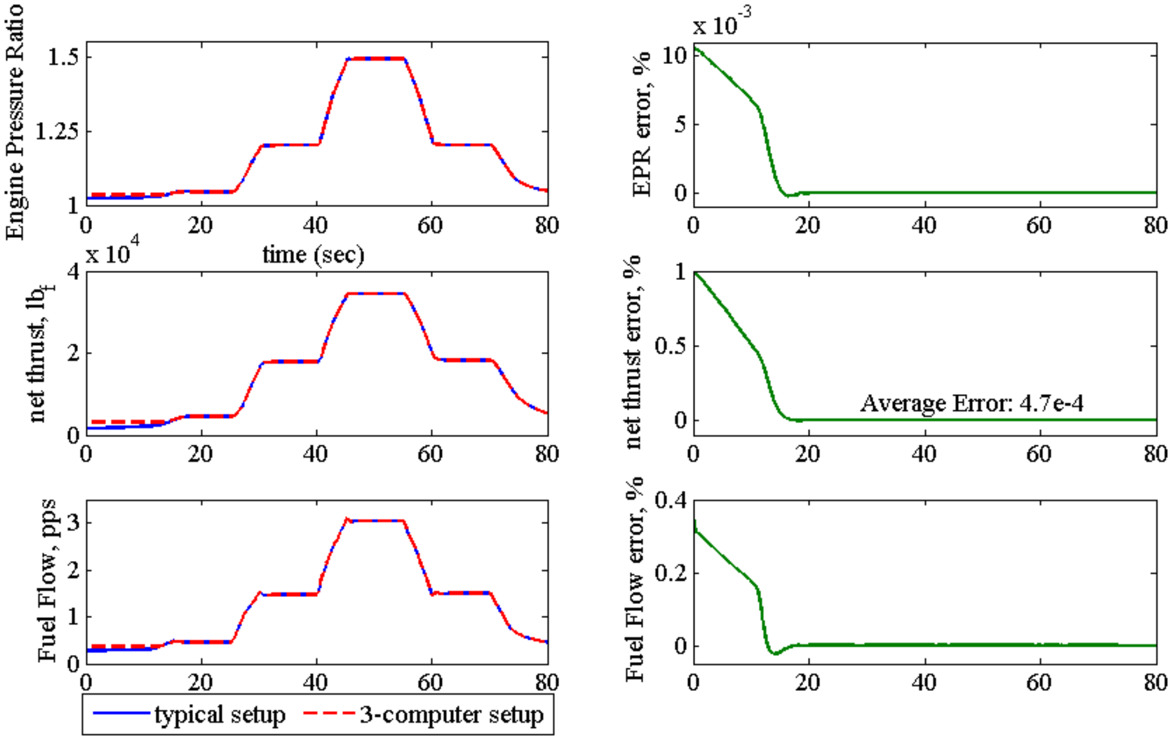

Figure 3. Comparison of simulation results for an 80 second flight profile. In the left column, blue solid lines indicated results from the typical C-MAPSS40k setup while red dashed lines are results from the three-computer setup. The differences between the two sets of results are shown in green in the right column.

B. Testing limitations of the network

To investigate the performance of the LAN, a series of network tests are performed using the open-source network analysis software Wireshark® (Wireshark Foundation).[8] Wireshark captures and provides information about each data packet sent and received by a computer. This information may be used to analyze network performance in terms of data throughput, packet drop, and average latency. Where data throughput is measured in megabits per second (Mbps), packet drop is measured as a percentage, and latency is measured in seconds. Complete information about data throughput and dropped packet count is obtained by simultaneously using Wireshark on all three platforms in the HIL test bed. Data throughput was measured by executing a flight profile without variation (no step input). The amount of traffic present on the network is independent of flight profile because the same amount of information is transferred between platforms every control interval, even if there are no new control commands. The simulation was run for 22 minutes with a control interval of 15 milliseconds (ms).

This test revealed that packet transfer rates for the operator platform and the engine platform averaged 3.2 Mbps. The control platform had an average packet transfer rate of 1.8 Mbps. Traffic trends from the control platform during an engine simulation are shown in Figure 4; trends from the operator and engine platforms were similar. Peak transfer rates for the different platforms ranged from 4.1 to 6.7 Mbps and are experienced during test setup (100 to 110 seconds in Figure 4). For a 1-Gigabit network, this communication scheme utilizes less than 1% of the network capacity at startup and less than half of that during the rest of normal operations. This indicates that the amount of data passed during a control interval is small compared to the overall available bandwidth. If the amount of data to be transferred exceeded the available network bandwidth during a control interval, network utilization would tend towards 100%. Furthermore, there was not a single occurrence of packet loss over the 22-minute test. This is not unexpected, as the network load is very low and there is no external traffic traversing the network. Additionally, full-duplex network hardware was selected to minimize collisions. If a collision does happen during a simulation, the data from the previous time step will be used in place of missing data from the current time step.

The 15 ms control interval in C-MAPSS40k corresponds to an average network loading of less than 1%. With such light network loading, the LAN can accommodate a shortened control interval. However, with previous results showing marginal differences between the typical and the 3-computer C-MAPSS40k setup, there is no need to shorten the control interval at this time. Additionally, there is plenty of bandwidth to increase the amount of data transferred at each control interval. This supports adding additional sensors into the model at a future date.

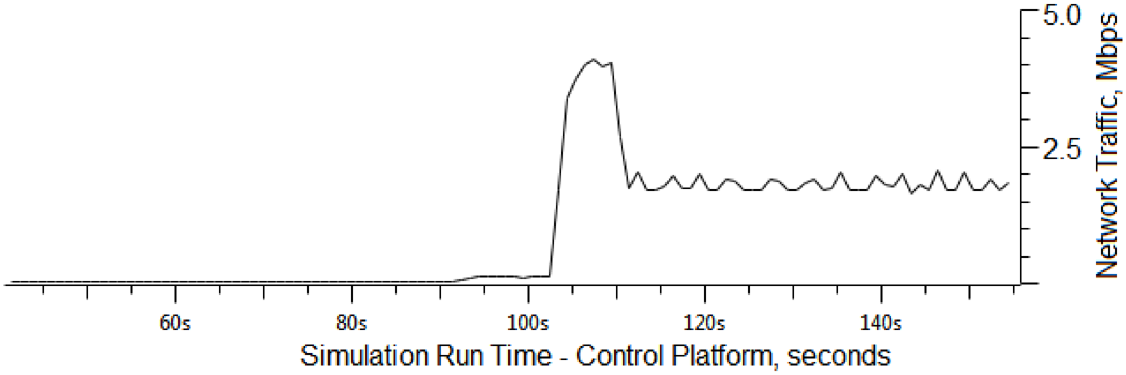

Figure 4. Network transfer speeds for the control platform. Data produced by Wireshark.

Latency of the network, also known as packet delay, is determined by measuring the response time to the ping command issued to each network link. From this, the round-trip delay could be calculated. No network loading was present during these tests, a condition that approximates the expected <1% loading of the network during a typical simulation. Since latency is a function of network loading, it is expected that a heavily used network would exhibit longer latency. However, due to the small amount of information transferred between platforms, doubling the number of sensors or decreasing the control interval to 1 ms will not cause high network loading. Thus, the experimental latency values presented in Table 1 should be typical of any simulation conducted on the system. Latency detected by Wireshark is reported as a round-trip time. The measured latency was halved to yield the one-way latency for each link.

Table 1. One-way packet latency between HIL platforms:
Control Platform (CP), Engine Platform (EP), and Operator Platform (OP)

	CP/EP	CP/OP	OP/EP
Mean (ms)	0.219	0.146	0.196
Standard Deviation 1σ (ms)	0.015	0.020	0.025

The communication scheme currently implemented in the HIL test bed involves a six-packet transfer sequence to allow all the information to be exchanged between platforms. Assuming a sequential packet transfer, we would expect the total mean transfer time to be 1.06 to 1.18 ms (3σ). A UDP broadcast communication scheme is being investigated which is expected to cut the total number of transfers to three and reduce the time needed to perform the data transfer to 0.53 to 0.59 ms (3σ). The current control interval for the HIL test bed is 15 ms. Using a broadcast scheme will use <4% of the control interval for communications over the LAN, as opposed to the 8% of the interval used by the six-packet scheme. It is desirable that data transfer takes as little time as possible, so that the maximum amount of time during the control interval can be allotted to control calculations. In either case, data transfer leaves ample time for calculations in real-time operation and allows for the possibility of reducing the control interval in the future, if desired.

IV. Conclusions & Future Work

The Commercial Modular Aero-Propulsion System Simulation 40k (C-MAPSS40k) has been modified to operate on three separate computers connected on a dedicated User Datagram Protocol (UDP) Local Area Network (LAN). This dedicated network is characterized by high data throughput, low packet loss, and low latency communications, all of which are essential for hard real-time simulation of this system. Initial comparisons against the typical single-computer C-MAPSS40k setup revealed less than 0.0005 % difference in simulation results for net thrust, suggesting minimal effect from the LAN communication. Additional tests showed that the maximum bandwidth usage is less than 1% of the 1 Gigabit per second (Gbps) LAN. The network communications between the three platforms take less than 1.18 milliseconds (ms), less than 8% of the 15 ms control interval in C-MAPSS40k, leaving ample time for computing the control logic. In a 22 minute simulation, the LAN had no packet losses. This is due to the combination of low network usage and the use of a full-duplex switch as opposed to a router.

These results highlight the strengths associated with using a dedicated UDP LAN network for inter-platform communications in the Hardware-In-the-Loop (HIL) test bed. The HIL system has excellent extensibility because the communication architecture has low latency, packet drop, and bandwidth usage. This enables additional sensors and actuators to be included in the simulation without significantly affecting the network. These traits support future studies that may consider decreased control interval time, increased information transfer due to additional sensors & actuators. Additionally, the networked system will enable incorporation of proprietary engine models into the HIL without compromising the intellectual property of the model.

The HIL test bed is being developed to support Distributed Engine Control (DEC) simulations for turbine engines. These simulations are targeted to include an investigation of control network architectures that use digital rather than analog signaling. DEC for turbine engines has the capability to decrease engine weight and lifecycle costs while maintaining essential characteristics of high-reliability. There are many opportunities for outside entities to collaborate with NASA and test their engine, actuator, or sensor models in the HIL test bed.

Acknowledgments

The authors would like to thank James Liu of N&R Engineering, for his work on splitting the C-MAPSS40k engine and controller into separate model files. Additionally, the authors thank Johnathan Kratz of NASA, for his initial network characterization work.

References

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

[2] Saluru, Deepack C., Yedavalli, Rama, K., "Fault Tolerant Model Predictive Control of a Turbofan Engine using C-MAPSS40k," *Proceedings of the 51st annual AIAA Aerospace Sciences Meeting*, No. AIAA-2013-0128, Grapevine, Texas

[3] Cully, D. E., Thomas, R., and Saus, J., "Concepts for Distributed Engine Control," *Proceedings of the 43rd Join Propulsion Conference and Exhibit*, No. AIAA-2007-5709, Cincinnati, OH, July 2007.

[4] May, R. D., Csank, J., Lavelle, T. M., Litt, J. S., and Guo, T.-H., "A High-Fidelity Simulation of a Generic Commercial Aircraft Engine and Controller," *Proceedings of the 46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, No. AIAA- 2010-6630, Nashville, TN, July 2010.

[5] Csank, J., May, R. D., Litt, J. S., and Guo, T.-H., "Control Design for a Generic Commercial Aircraft Engine," *Proceedings of the 46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, No. AIAA-2010-6629, Nashville, TN, July 2010.

[6] United Electronic Industries, "UEI Application Notes: A Modern Alternative to Reflective Memory and VME," United Electronic Industries, Inc, Walpole, MA, 2010

[7] Dann, James C., Encore Computer, U.S., Inc., U.S. Patent Application for a "Multiprocessor system with reflective memory data transfer device," Patent No. 5581732, filed 13 Oct. 1993.

[8] Sharpe, R., Lamping, U., Warnicke, E., "Wireshark User's Guide, v1.99.0-rc1-261-g4dfbb79 for Wireshark 1.11," Wireshark Foundation, 2013