Design and Benchmarking of a Network-In-the-Loop Simulation for Use in a Hardware-In-the-Loop System

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

George L. Thomas[†]
N&R Engineering, Parma Hts, OH, 44130, USA

and

Dennis E. Culley[‡] and Jonathan L. Kratz[§]
NASA Glenn Research Center, Cleveland, OH 44135, USA

Distributed engine control (DEC) systems alter aircraft engine design constraints because of fundamental differences in the input and output communication between DEC and centralized control architectures. The change in the way communication is implemented may create new optimum engine-aircraft configurations. This paper continues the exploration of digital network communication by demonstrating a Network-In-the-Loop simulation at the NASA Glenn Research Center. This simulation incorporates a real-time network protocol, the Engine Area Distributed Interconnect Network Lite (EADIN Lite), with the Commercial Modular Aero-Propulsion System Simulation 40k (C-MAPSS40k) software. The objective of this study is to assess digital control network impact to the control system. Performance is evaluated relative to a truth model for large transient maneuvers and a typical flight profile for commercial aircraft. Results show that a decrease in network bandwidth from 250 Kbps (sampling all sensors every time step) to 40 Kbps, resulted in very small differences in control system performance.

Nomenclature

Acronyms

ADC	Analog to Digital Conversion
C-MAPSS40k	Commercial Modular Aero-Propulsion System Simulation 40k
CRC	Cyclic Redundancy Check
DAC	Digital to Analog Conversion
DEC	Distributed Engine Control
EADIN Lite	Engine Area Distributed Interconnect Network Lite
EMI	Electromagnetic Interference
FADEC	Full Authority Digital Engine Controller
HIL	Hardware-In-the-Loop
HPC	High-Pressure Compressor
HPT	High-Pressure Turbine

[*] Research AST, Control Systems, eliot.d.aretskin-hariton@nasa.gov

[†] Controls Engineer, george.l.thomas@nasa.gov

[‡] Research Engineer, dennis.e.culley@nasa.gov, AIAA Senior Member

[§] Research AST, Control Systems, jonathan.kratz@nasa.gov, AIAA Member

LIN	Local Interconnect Network
LPC	Low-Pressure Compressor
LPT	Low-Pressure Turbine
ms	millisecond
NSA	National Security Agency
NIL	Network-In-the-Loop
PIL	Processor-In-the-Loop
RMS	Root-Mean-Square
SLS	Sea-Level Static
SHA3-256	Secure Hash Algorithm 3, 256 bits
SXD	Smart Transducer
TEDS	Transducer Electronic Data Sheet

Variables

EPR	Engine Pressure Ratio
FAR	Fuel-to-Air Ratio
Nc	Core Shaft Speed
Nf	Fan Shaft Speed
P2	Engine Inlet Pressure
P25	Pressure at the Exit of the LPC
P50	Pressure at the Exit of the LPT
PLA	Power Level Angle
Ps3	Pressure at the Exit of the HPC
Fnet	Net Thrust
SM	Surge Margin
T2	Engine Inlet Temperature
T25	Temperature at the Exit of the LPC
T30	Temperature at Exit of the HPC
T40	Temperature at the Inlet of the HPT
T50	Temperature at Exit of the LPT
TSFC	Thrust Specific Fuel Consumption
VBV	Variable Bleed Valve
VSV	Variable Stator Vane
Wf	Fuel Flow

I. Introduction

Distributed Engine Control (DEC) is a modular hardware architecture that is being developed to address the more severe system constraints of next generation turbine engines. The temperature and compact size of these new engines makes integrating controls far more difficult. Fortunately, DEC also has the potential to bring new capability and performance to the engine system. Benefits occur through weight reduction, improved fault diagnostics, streamlined engine upgrades, obsolescence management, and implementing conditional control strategy optimization.[1–3] DEC is an extension of the Full Authority Digital Engine Control (FADEC) architecture used today. DEC physically separates the high-performance computing aspect of the controller from the engine system. The analog conversion function of the controller can be embedded into the sensors and actuators which remain on the engine. The connection between the computing, sensing, and actuation elements is bridged by a digital control network. This capability becomes feasible by using embedded high-temperature electronics in the form of smart nodes associated with these control elements. It is essential for future engine design to understand the full implications of DEC technology including: the limitations imposed by high-temperature electronics, and the effect of network communications.

The main focus of this paper is to develop knowledge about control system performance under real-time network constraints which directly affect loop closure as well as being the mechanism for integrating the distributed control system. Previous studies[4–7] have employed a series of increasing levels of network fidelity to improve the understanding of information flow within a distributed control system. The Commercial

Modular Aero-Propulsion System Simulation 40k (C-MAPSS40k) software is used as the engine model for this work. It is a realistic, 0-dimensional dynamic simulation of a 40,000 lb thrust class, high bypass turbofan engine system.[8, 9] The baseline C-MAPSS40k is used as a truth model reference for evaluation of the subsequent model changes developed for distributed control architectures.

In Culley et al.,[4] the baseline model was decomposed into separate model referenced structures in order to reflect the independence of the engine plant model from the control system. Further decomposition of the control system followed in Zinnecker et al.[5] by separating the controller from the sensors and actuators following the idealized hardware architecture of a distributed system. The sensors and actuators were subsequently developed as independent smart nodes. The smart nodes included many features to reflect their implementation as hardware elements, such as limited quantization and noise injection.

Distributed control relies on digital communication networks that require the serial communication of data, meaning the arrival time of information from various components can vary greatly. Furthermore, the throughput capability of the communication media has the potential to impart significant restrictions on the quantity of data transmitted within a specific interval. Traditional aircraft engine control architectures and the modeling tools used to describe their operation do not address these issues. This realization spurred the development of empirical methods using the smart node models embedded on generic microcontroller hardware in a configuration known as Processor-in-the-Loop (PIL).[6] The hardware smart nodes were intended to replicate the 'systems of asynchronous systems' inherent in distributed control.

The concept of a network schedule was introduced in Culley et al.[7] Here, the data flow between individual control elements was throttled by a predefined network schedule which acted like a gate and mask, restricting when new data reached the controller. This configuration is referenced as the ideal Network-in-the-Loop (ideal NIL) model because it represents a control system with an ideal scheduled network without network induced transport delays. In that study, the basis for restricting data throughput was the frequency response of the control elements. Network throughput reductions of as much as 90% were demonstrated in software simulations while still maintaining adequate performance.

In the current study, the PIL hardware configuration was merged with the network scheduling mechanism by using the NASA-developed Engine Area Distributed Interconnect Network Lite (EADIN Lite) communication network.[10] EADIN Lite represents a realistic multi-drop digital communication link between the controller and smart-nodes. This configuration is designated as the Network-in-the-Loop (NIL) model. The purpose of the current study is to analyze control system performance under real-time network constraints and to compare the performance of the physical distributed control hardware versus the ideal NIL model. The modeling and simulating of these networks will be critical in the integration, upgrading, and flight certification of distributed engine control systems and components. What follows in Section II provides details about the HIL simulation system including the models under consideration, the design of the control network, and test conditions evaluated. Section III will describe the results of the benchmarking simulations and discussion. Conclusions and future work are given in Section IV.

II. Simulation Setup

A. Control System

Four configurations of C-MAPSS40k were considered: 'baseline', 'Processor-In-the-Loop' (PIL), 'ideal Network-In-the-Loop' (ideal NIL), and 'Network-In-the-Loop' (NIL). The baseline model is used as the truth model in order to have something to compare the other models against. The baseline model is run using the original C-MAPSS40k model on a single computer. The PIL model contains the same C-MAPSS40k engine and controller, with the exception that the sensors and actuators of the system are replaced by 13 smart transducer (SXD) models executing on microcontrollers. This is enabled by an Ethernet network which is meant to be transparent to the control system. The ideal NIL model uses baseline C-MAPSS40k and a network block which implements a sample and hold operation on sensor and actuator signals to limit communication data flow in the system according to the schedule.

A SXD is defined in this work as a sensor or actuator with integrated processing and digital communications capability. As shown in Fig 1, a SXD contains transducer hardware (the analog sensor or actuator itself), a signal conditioning component, one or more digital-to-analog or analog-to-digital converters (DACs or ADCs) as appropriate, an application processor (containing software that acts on inputs and outputs from the controller network and engine), and a network interface component. A SXD may also contain meta-information in the form of a transducer electronic data sheet (TEDS). SXDs constructed in this way

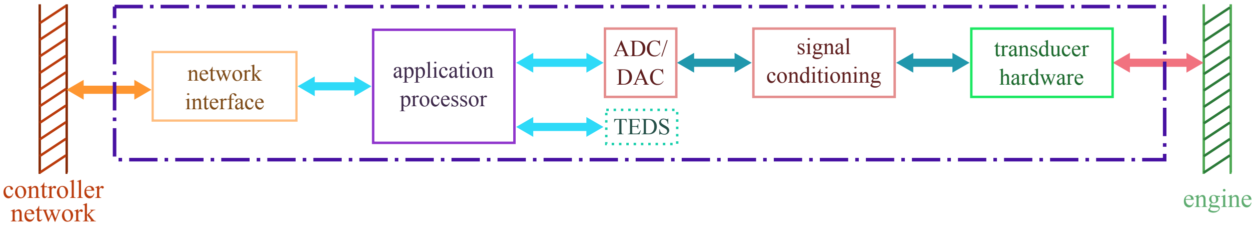

Figure 1. Diagram of a Smart Transducer (SXD) showing internal and external components.

are consistent with the IEEE 1451 standard.[11]

The NIL architecture shown in Fig 2, contains a real-time control network using the EADIN Lite protocol to connect the SXD models to the controller. To connect the C-MAPSS40k model (which was coded in Matlab/Simulink) to the SXD models, a soft-real-time block was used. Without this block, the C-MAPSS40k model would run faster than the real-time network and flood it with data. The soft-real-time block allows C-MAPSS40k to be paced so it can run in time with the network.

The data presented in this report was collected for five simulations of each control model. Initial testing showed strong within-group matching which justifies the small sample size. It should be noted that the control data plots presented in this report are of representative runs, as opposed to an average of all runs, whereas tables contain summary information on all simulations.

B. Control Network

Various network hardware and software layers have been studied for safety-critical and real-time applications.[2,12–15] There are two primary network types being considered by industry for DEC, an Ethernet and an RS-485 multi-drop bus type network. The speed of these networks is expected to be less than 10Mbps. This is due in part to the difficulty of creating complex electronic components that can communicate at high-speed in continuous operating temperatures over 250°C. EADIN Lite was chosen as the protocol implementation for this study because it is similar in structure to the expected final protocol implementation for DEC.

The NIL configuration with the EADIN Lite control network is shown in Fig 2. During the simulation, sensed data and actuator commands must travel over the EADIN Lite network before the controller can receive new sensor data or the engine can respond to actuator manipulation. The Ethernet network shown in Fig 2 is ideally meant to be transparent to the information transfer process because it represents the physical interface between the sensors, actuators, and the engine. However, this communication was implemented in serial fashion which adds in a small delay compared with the ideal case. The Ethernet network is not part of the DEC architecture to be implemented on an actual engine. Details on microcontroller selection, Ethernet back-end network configuration and SXD models have been published.[6]

The test case presented here is implemented on an RS-485 half-duplex bus. The network was operated at 4 Mbps which was the maximum functional rate of the microcontrollers used in this study. RS-485 was selected for its robustness, due in part to large differential voltage used for signaling. Large differential voltage provides improved electromagnetic interference (EMI) resistance characteristics, which is essential for

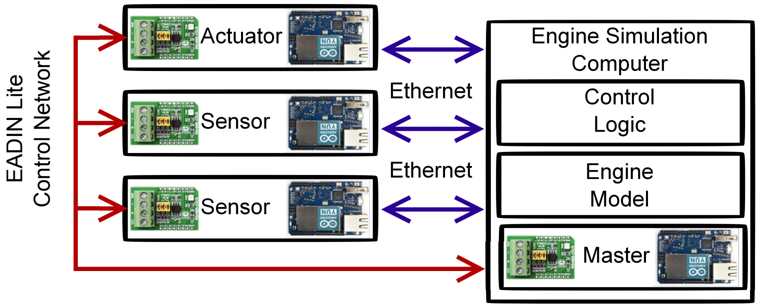

Figure 2. Network-in-the-Loop block diagram showing EADIN Lite control network in red. This network bridges the master controller to the SXD models. This network enables the sensors and actuators to share information between the control logic and the engine model. The Ethernet network shown in blue is not a control network and is ideally transparent to the information transfer process.

Table 1. NIL Schedule: The sampling interval for different sensors and actuators in the C-MAPSS40k system simulation based on Culley et al.[7] Each sensor or actuator is executed on an individual microcontroller and is queried separately. Queries are offset so that at most only two nodes are contacted during a single minor frame which limits peak bandwidth requirements.

Node Type	Wf, VSV, VBV, P2, P25, Ps3, P50, Nc, Nf	T2, T25, T30, T50
Sample Interval	10 minor frames	25 minor frames

robust aircraft engine operations. While communication request retries are often used to add communication robustness in the case of burst EMI, they are less effective in continuous noise environments. EMI resistance or noise immunity on copper-based networks is strongly coupled to differential voltage on the communication line with modifiers based on the implementation of spread spectrum, shielding, receiver sensitivity, and encoding. Copper-based networks are the only option being considered for aircraft engine DEC at this time due to high-temperature compatibility requirements.

The software implemented for the NIL simulation was the EADIN Lite protocol which is loosely based on the Local Interconnect Network (LIN) specification. EADIN Lite uses a fixed length packet size of 18 bytes in a master-slave system. Figure 3 shows the individual elements of a message: three bytes of preamble, five bytes of header information, eight bytes of payload, and lastly a two byte cyclic redundancy check (CRC). The CRC is used to ensure data has not been corrupted in transit. The protocol uses a master-slave architecture where the master node polls the slave nodes for information and sends command information. Slave nodes only exchange information with the master controller. The master must make a request for this exchange to initiate.

EADIN Lite is a master-slave protocol. Information flow in the network is managed by a time-triggered schedule process. The schedule enables each node to communication with the controller at a periodic rate that is commensurate with the frequency content of the information transmitted. Schedule creation is based on the Nyquist sampling theorem which states that in order to be able to reconstruct such a signal from a set of samples, the sampling rate must be at least two times the signal bandwidth of the highest frequency present in the signal.[16] A smart transducer must sense plant outputs or drive plant inputs, and so a sampling rate must be chosen such that this ability is not compromised. Excessive sampling rates impose penalties on network bandwidth which creates the incentive to match the schedule with the transducer response time. Insufficiently large sampling rates will compromised the control system's ability to function.

The schedule uses two constructs, a minor frame and a major frame. A minor frame describes which nodes will communicate during the minor frame interval and the order in which that occurs. In general, the minor frame interval is considered equivalent to the controller's interval, however, it is not necessarily required that the two be synchronized or in phase. A major frame is composed of a set of minor frames that form a repeating sequence. This guarantees periodicity between each individual node and the controller. For simplicity, we assume the major frame consists of 50 minor frames. The scheduling array used for these tests has been described by Culley et al.[7] and includes increased sampling rates for fan speed (Nf) and core speed (Nc). Table 1 shows a shortened version of the schedule and the repetition rate in minor frames for each node in this system configuration. For example, P2 only communicates with the controller every 10 minor frames. To minimize peak network usage, communication is limited to two nodes per minor frame. This is accomplished by offsetting nodes with the same rate to different minor frames.

This network implementation operates in real-time on consumer grade microcontrollers. This software is available for public download.[a] A key characteristic of the network protocol includes round-trip times less than 1 ms when operating at 4 Mbps. A round-trip time is the amount of time required for the master to send a command and process the reply from the slave node. This fast response time is required to ensure that the master can talk to the thirteen nodes of the baseline C-MAPSS40k model during the minor frame period of 15 ms.

It should be noted that the EADIN Lite network does not attempt to inject network errors into the communication between nodes because this study is not focused on control robustness with transmission errors. The Culley et al.[7] software network model also did not attempt to inject network errors. Injecting network errors at specific points in the flight profile is an important area for future research.

[a] https://github.com/nasa/EADINLite

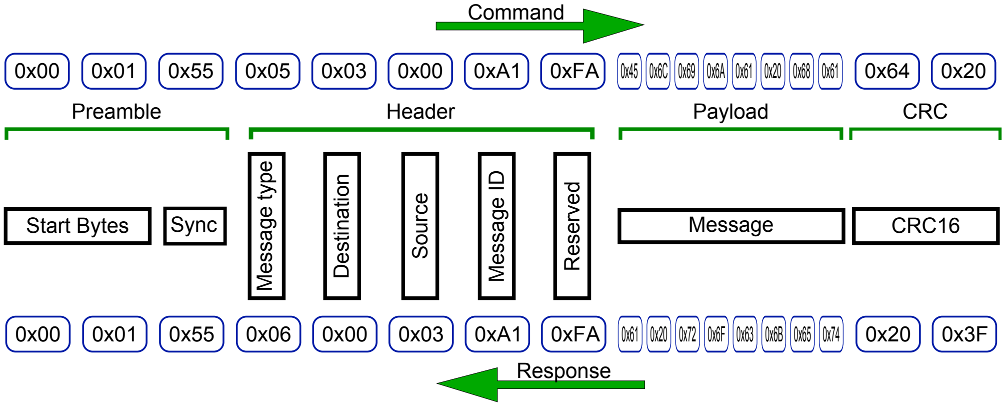

Figure 3. EADIN Lite protocol example message: A message sent from the master (0x00) to a sensor node (0x03) is shown on the top, and the response from the slave node is shown on the bottom.

C. Description of Test Cases

The following tests were conducted in order to observe the effects of adding smart transducer (SXD) logic (PIL case) and the EADIN Lite network (NIL case) to the closed loop C-MAPSS40k system. These tests allow one to determine whether or not the C-MAPSS40k system operates as intended after adding these components into the system. A series of power lever angle (PLA) burst and chop input profiles were fed into the system at seven different points throughout the flight envelope to obtain transient results. PLA is proportional to the throttle or power demand, and burst and chop inputs are sudden steps from low to high power demand, and high to low power demand respectively. These simulations are intended to represent the worst-case or most demanding throttle transients that would occur at each of these flight conditions in terms of engine operability margins. This is because, considering only engine transient effects, an engine will get closest to its minimum operability margins (minimum compressor surge margins and maximum turbine inlet temperatures) during these kinds of step transients. Flight profile evaluation points are listed by reference numbers in Table 2. Also, Fig 4 shows the points evaluated as they lie within the C-MAPSS40k flight envelope.

Table 2. List of ambient test conditions, each with a reference number that is used to refer to those conditions within this text.

Test Condition Reference #	Altitude (ft)	Mach #	Ambient Temperature (°F)
1	0	0.0	59
2	0	0.2	59
3	4,000	0.2	45
4	30,000	0.8	-48
5	4,000	0.2	95
6	0	0.2	109
7	0	0.0	109

To study the steady-state limit regulators between transients, extreme PLA commands were designed. The min and max values in the burst-and-chop PLA profiles were chosen as 40° and 100°, as these lie outside of the range of normal PLA command values (42° to 80°). This allows more explicit testing of transitions between control modes, and it is often during these transitions that the engine transient limits are most likely of being violated.

To verify that fuel economy is not compromised under the different control architecture configurations, a fifty minute flight profile was also run with all three simulation configurations. Engine fuel economy, often quantified as thrust specific fuel consumption (TSFC), is of critical importance because fuel costs are a significant part of total airline operating costs. This flight profile was created from engine data recorded during a real flight. The data was obtained from a NASA sample flight database.[b] Data from this database has been used for other aeropropulsion controls research studies.[17] Specifically, the data used for this test

[b] NASA DASHlink web site, Sample Flight Database, Accessed 10/14/2016, URL: https://c3.nasa.gov/dashlink/projects/85/

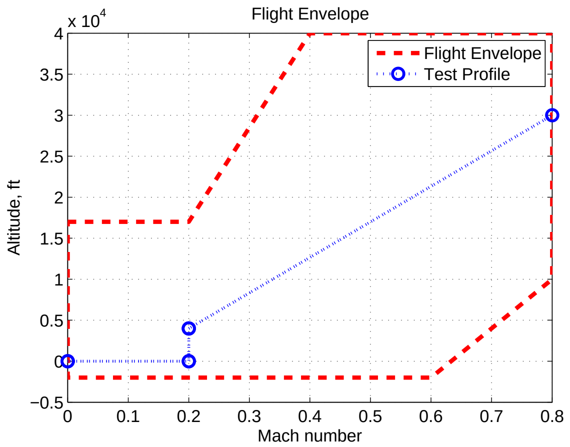

Figure 4. C-MAPSS40k flight envelope and test profile. Points circled represent the reference conditions from Table 2. Each operating point was tested assuming an ambient temperature equal to the standard atmospheric temperature and also a 50°F offset from that temperature to simulate hot day effects.

was obtained from the records of engine one in the file *687200104140748.mat* from the dataset *Tail 687*. Figure 5 shows the ambient conditions specified in this profile (altitude, Mach number, ambient temperature, and PLA specified by the profile).

The effect of smart transducer and digital control network non-idealities on transient system operation must also be studied. These non-idealities include quantization in the SXD ADC and DAC, signal conditioning effects, sensor noise, and network-induced transport delay. Transient data from the simulation run at 59 °F at sea-level static (SLS) will be presented (figs. 6 to 8). SLS corresponds to an engine running at Mach zero at an altitude of zero feet. This paper examines the same quantities that were examined in previous

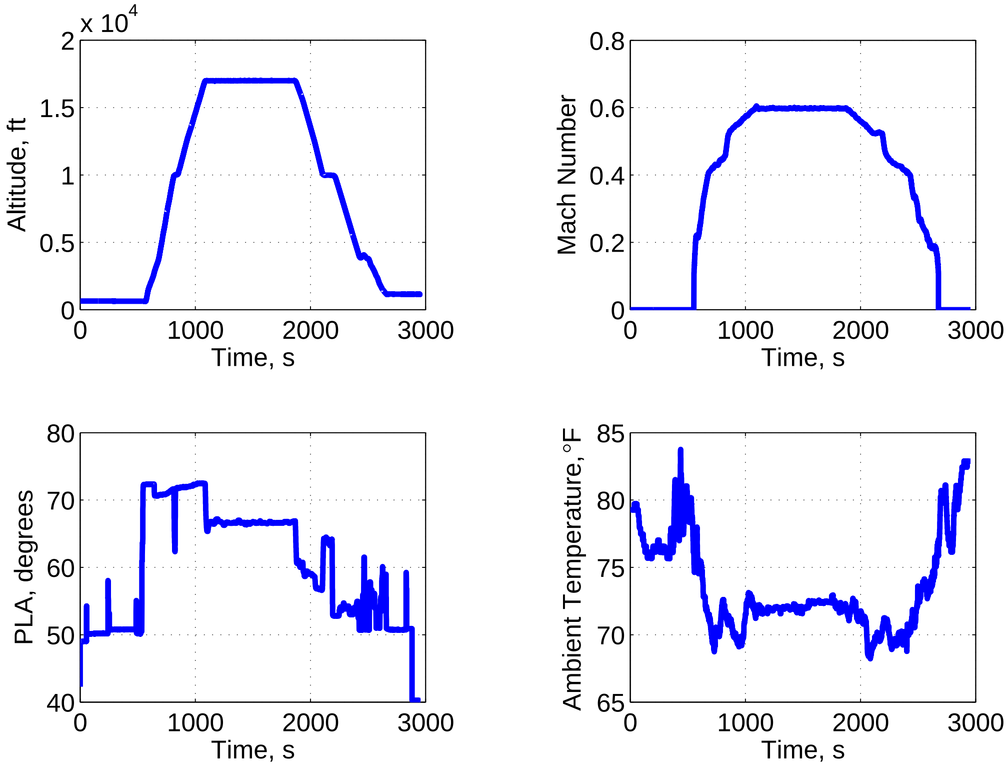

Figure 5. Flight profile for engine data recorded during a real flight and recorded in the NASA sample flight database.

work.[7] The results from this previous work represent the ideal operation of an aeropropulsion control system network with scheduling logic. It is considered ideal because that simulation did not include SXD models and data transfer occurred instantaneously. The NIL simulation represents a hardware implementation of the same kind of control system network with the same scheduling logic. Additionally, the NIL simulation also includes the added smart transducer modeling fidelity. Any differences observed between these cases will highlight differences between ideal and realistic control networks.

III. Results and Discussion

A. Control System Performance

The data extracted from the burst-and-chop simulations is percent root-mean-squared (RMS) steady-state engine pressure ratio (EPR) tracking error, as well as engine 95% rise time, the minimum compressor surge margins, the minimum fuel-to-air ratio (FAR), and the maximum turbine inlet temperature (T40) observed during each of the transients. This data allows one to determine if the non-ideal phenomena represented by the additional transducer and network models affect the engine control system in a substantial way with respect to both its ability to operate within design margins, and its ability to operate as commanded.

The steady-state percent RMS error data for EPR tracking are shown in Table 3. Note that the steady state data were computed over 15 second periods of time after each of the transients settled out. The values presented in the table are the greater of the two RMS errors computed at low and high power. This corresponds to a PLA of 40° for low power and 100° for high power. This data is given as percentages of the baseline average value over their respective 15 second period. Further, error is computed in this analysis by comparing the baseline C-MAPSS40k control system to the PIL and NIL systems.

Table 3. Percent RMS EPR tracking error for the PIL and NIL simulations.

Test Condition Reference #	PIL EPR Tracking Error	NIL EPR Tracking Error
1	0.07%	0.06%
2	0.06%	0.17%
3	0.16%	0.20%
4	0.31%	0.29%
5	0.03%	0.06%
6	0.08%	0.09%
7	0.09%	0.16%

As seen in Table 3, the RMS EPR tracking error is consistently within 0.3% of the baseline steady state EPR value. This indicates that the controller successfully tracks its setpoint throughout the flight envelope. This verifies that the steady-state performance of the control system is not significantly affected by the proposed control system network or the non-idealities imposed by the smart transducers.

It is important to verify that these changes to the control system do not degrade engine performance and TSFC. Based on the flight profile presented in Sec II.C, TSFC was computed at each time instant of each of the 50 minute simulations and then integrated over simulation time. The time integrated TSFC values for the baseline, PIL, and NIL simulations are 0.37503, 0.37493, and 0.37481 respectively. Since these differ less than 0.6%, one can conclude that there is no significant difference in engine TSFC as tested over a realistic flight profile. This, and data in Table 3 show that the PIL and NIL systems perform similar to the baseline system.

The NIL results shown in figs. 6 to 8 agree well with the ideal NIL results from previous work.[7] Figure 6 shows shaft speeds (Nc, Nf), engine pressure ratio (EPR), and the ratio of fuel flow to high-pressure compressor (HPC) exit pressure (Wf/Ps3). The last value is used for purposes of controller limit logic. Fig 7 shows the net thrust produced by the engine and the HPC and low-pressure compressor (LPC) surge margins. Finally, Fig 8 shows actuator response signals from the system. The NIL system and ideal NIL system both achieve a faster response time than the baseline due to the reduced effective sampling rate of the Nc sensor, which causes the acceleration limiter to activate later in the throttle burst transient than it normally would. Further, there is some level of ringing in the fuel flow (Wf) actuator response in both systems. Despite these differences, both systems protect their respective surge margin limits.

The most significant differences between the NIL and ideal NIL systems is that the NIL system achieves

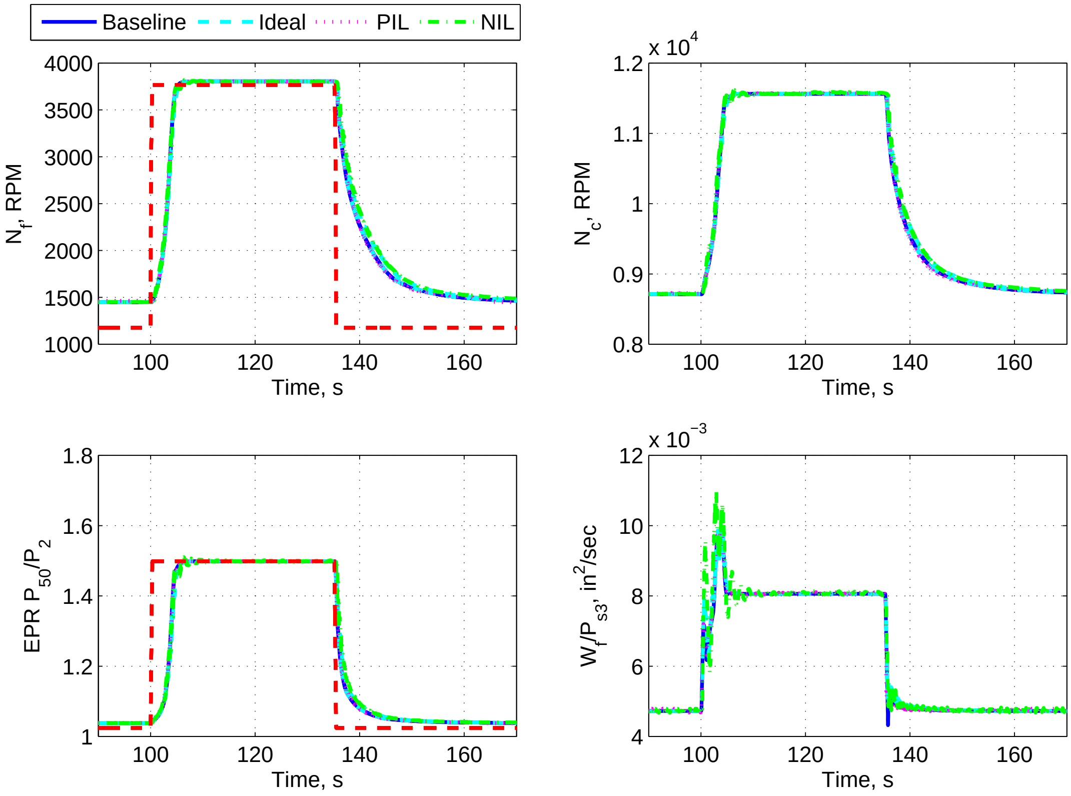

Figure 6. Transient engine control variable results at sea-level static comparing the baseline, PIL, and NIL simulation test cases. The ideal NIL from Culley et al.[7] is also shown. The red dashed line is the set point.

a slower response time than the ideal NIL system, and that the NIL system appears to have a larger degree of Wf actuator response ringing (Fig 6). Both of these differences are attributed to delays inherent in the communication system in the NIL case that are not modeled in the ideal NIL system case. Specifically, there is some message delay in the NIL system, whereas messages were assumed to travel instantaneously in the ideal NIL system. Delays in a closed-loop system can exacerbate ringing and oscillation, especially in a control system that primarily relies on integral control such as the one used in C-MAPSS40k. Delays should also be expected to reduce system response speed, which is consistent with what was observed.

A practical control system must be designed with consideration to non-idealities introduced by the communication networks and smart node electronics. The C-MAPSS40k controller used in this work has not been redesigned to reflect the changes associated with the PIL and NIL models, as this is considered outside of the scope of this work. Thus, techniques for aeropropulsion control system design to address networking issues is an attractive item for future work.

B. Control Network

The results from the NIL model described in Sec III.A closely match that of the baseline, therefore we can investigate network bandwidth requirements for the C-MAPSS40k engine. In the NIL model, the EADIN Lite network moves at maximum 72 Bytes per minor frame (15 ms). This corresponds to two nodes being sampled during a minor frame which is four 18 Byte messages. To transmit this information, a minimum network bandwidth of 40 Kbps is required which includes both message overhead and payload. Thus, to adequately support the control function for the C-MAPSS40K engine model, we expect to use 40 Kbps of network bandwidth. These results are also close to other assessments of 52 Kbps required bitrate for a T700 turbo-shaft engine which includes more instrumentation than the C-MAPSS40k model.[18] In comparison, the C-MAPSS40k control function, for a network where all thirteen sensors and actuators were sampled every minor frame, would require 250 Kbps without lending significant improvements to control capability. The careful choice of a scheduling scheme allowed network bandwidth to be significantly reduced for the NIL model.

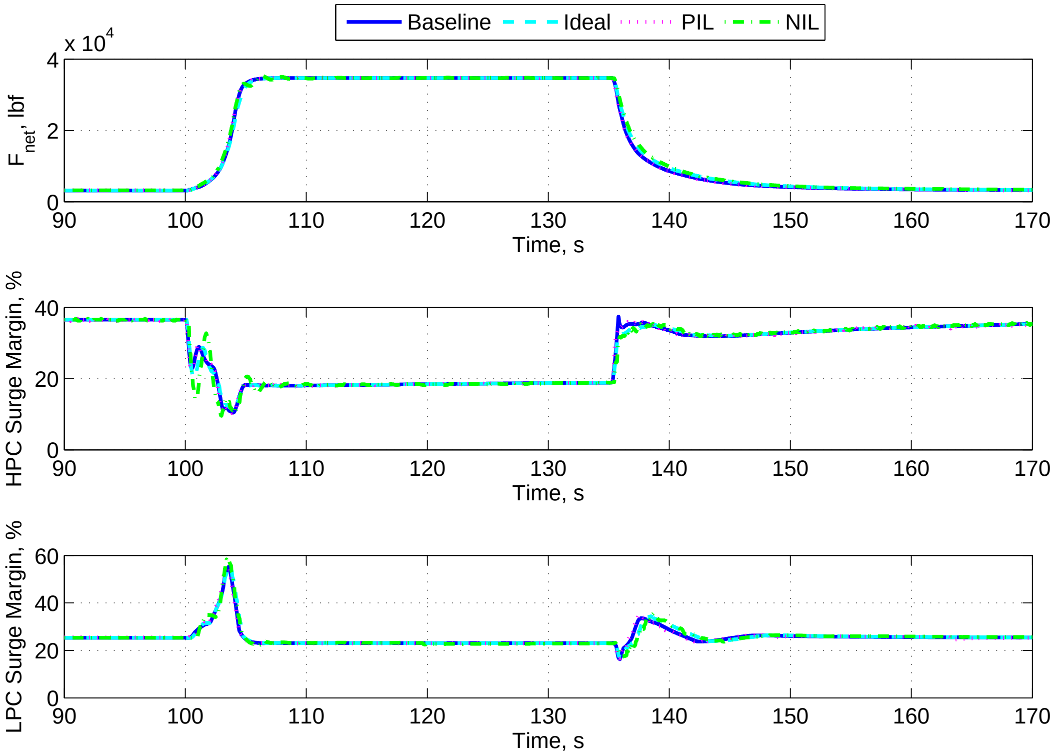

Figure 7. Net thrust produced by engine, and high-pressure compressor (HPC) and low-pressure compressor (LPC) surge margins obtained by running a burst and chop transient at SLS, showing the baseline, PIL, and NIL simulation test cases. The ideal NIL from Culley et al.[7] is also shown.

Some mechanisms to secure digital networks against hacking may increase the required control network bandwidth. National Security Agency (NSA) message authentication in the form of a 256 bit secure hashing algorithm three (SHA3-256) signature including random number generator and timer could push message size from 18 Bytes to 96 Bytes. To append a SHA3-256 signature to messages in the NIL would require network bandwidth to increase from 40 Kbps to 205 Kbps. The bandwidth overhead for authentication is very high when compared to the control bandwidth requirements. Other methods that do not require additional network bandwidth include encryption[19,20] and SXD anomaly monitoring.[21] Using all these methods to secure the network to ensure secure communications over the short and long term would be prudent.

In addition to security, scaling the network to handle future capability must also be considered when designing an engine control architecture. The C-MAPSS40k model implemented in this study had a total of 13 SXD nodes communicating with a master node on a set schedule. However, the number of SXDs and the sampling frequency of those nodes may increase in the future. Further, the desire to transfer health monitoring and fault code data as well as adding the capability of sending communication retries may also increase network bandwidth demand. Adding 10 Bytes per message of health or fault data and increasing the number of individual messages per time step from 4 to 8 allows for additional sensors, faster sensor sampling, or additional retries. The overall network bandwidth then becomes 450 Kbps.

To execute the C-MAPSS40k control system on a authenticated and encrypted network, with margin to add nodes or sample nodes faster, in addition to providing for health monitoring data, a network with a minimum network capacity of 450 Kbps is required. If message authentication is not desired, lower bandwidth networks of 120 Kbps will be able to deliver the same performance.

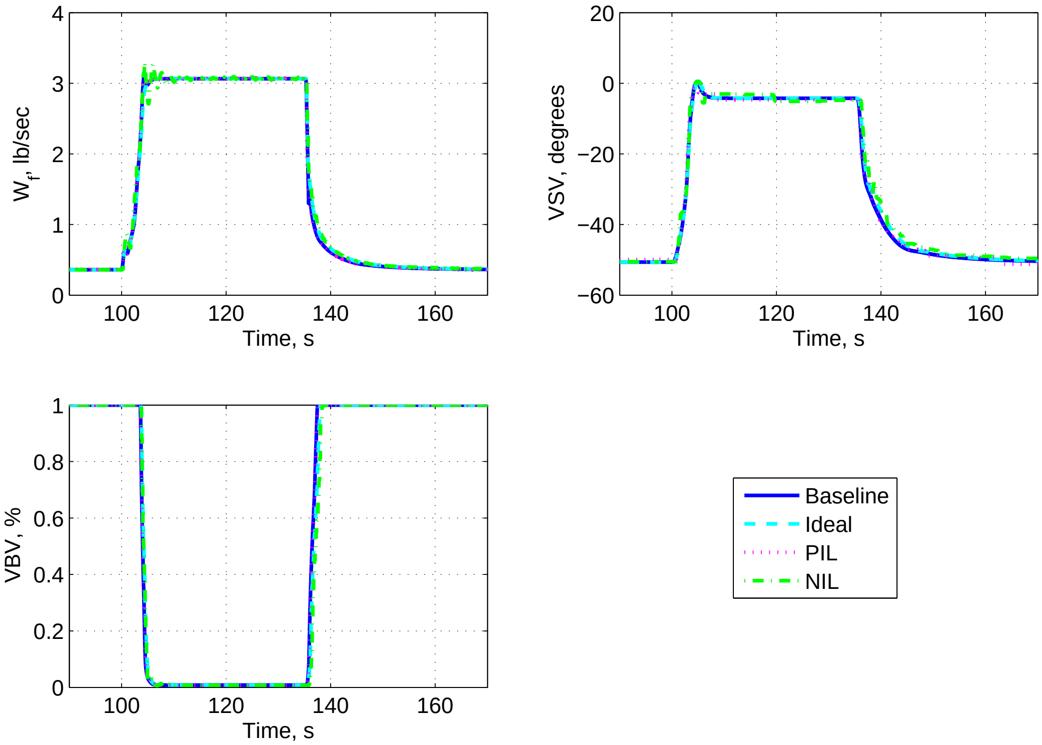

Figure 8. Actuator response obtained by running the baseline, PIL, and NIL simulation test cases. The ideal NIL from Culley et al.[7] is also shown.

IV. Conclusion

Performance characteristics of an engine system simulation in several configurations were explored using the Commercial Modular Aero-Propulsion System Simulation 40k (C-MAPSS40k). A real-time network using the NASA-developed Engine Area Distributed Interconnect Network Lite (EADIN Lite) protocol was described and incorporated into C-MAPSS40k to create a Network-In-The-Loop (NIL) model. This model limits the amount of communication with each smart node based on a defined schedule in order to limit network usage while maintaining controllability. NIL and baseline C-MAPSS40k models were assessed during slow and fast transients throughout the flight envelope. The NIL controller successfully tracked its setpoint throughout the flight envelope. This shows that network and smart transducer (SXD) non-ideality effects on engine performance were small. Non-ideality effects include: quantization, signal conditioning, sensor noise, and network-induced transport delay. The NIL model also did not experience significantly degraded engine performance or thrust specific fuel consumption (TSFC) when compared to the baseline. The NIL network was observed to have minor differences from an ideal network model due to message latency effects. Improved network modeling will allow for the integration, upgrading, and flight certification of distributed engine control systems and components. The NIL model demonstrated a control architecture for C-MAPSS40k that required only 40 Kbps of network bandwidth at peak. This represents a significant decrease from the baseline C-MAPSS40k model which requires 250 Kbps. The difference between the two is a result of decreasing scheduling node communication frequency. Actual data throughput requirements for an engine may vary depending on smart node count, network security schemes, health monitoring, and fault reporting requirements. Future work may include control system retuning and development of a network model that can simulate specific bus network error types.

V. Acknowledgments

This work is being performed at NASA Glenn Research Center under the Transformative Aeronautics Concepts Program, Transformational Tools and Technologies Project.

References

[1] Behbahani, A. R., "Achieving AFRL Universal FADEC Vision with Open Architecture Addressing Capability and Obsolescence for Military and Commercial Applications," AIAA 2006-4302, Sacramento, California, July 2006.

[2] Yedavalli, R., Belapurkar, R., and Behbahani, A., "Stability Analysis of Distributed Engine Control Systems Under Communication Packet Drop," *Proceedings of the 44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, AIAA 2011-6145, Hartford, CT, July 2008.

[3] Culley, D. E., Thomas, R., and Saus, J., "Concepts for Distributed Engine Control," *Proceedings of the 43rd Joint Propulsion Conference and Exhibit*, AIAA-2007-5709, Cincinnati, OH, July 2007.

[4] Culley, D., Zinnecker, A., and Aretskin-Hariton, E., "Developing an Integration Infrastructure for Distributed Engine Control Technologies," *AIAA Propulsion and Energy Forum and Exposition 2014: 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, AIAA-2014-3532, Cleveland, OH, July 2014.

[5] Zinnecker, A. M., Culley, D. E., and Aretskin-Hariton, E. D., "A modular approach to modeling hardware elements in distributed engine control systems," *AIAA Propulsion and Energy Forum and Exposition 2014: 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, AIAA-2014-3530, Cleveland, OH, July 2014.

[6] Aretskin-Hariton, E. D., Zinnecker, A. M., Kratz, J. L., Culley, D. E., and Thomas, G. L., "Benchmarking model variants in development of a hardware-in-the-loop simulation system," *AIAA Science and Technology Forum and Exposition*, AIAA-2016-1425, San Diego, CA, January 2016.

[7] Dennis E. Culley, George L. Thomas, E. D. A.-H., "A Network Scheduling Model for Distributed Control Simulation," *52nd AIAA/SAE/ASEE Joint Propulsion Conferencen*, Salt Lake City, Utah, July 2016.

[8] May, R. D., Csank, J., Litt, J. S., and Guo, T.-H., *Commercial Modular Aero-Propulsion System Simulation 40k (C-MAPSS40k) User's Guide*, NASA/TM-2010-216831, September 2010.

[9] 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*, AIAA-2010-6630, Nashville, TN, July 2010.

[10] Aretskin-Hariton, E. D., "EADIN Lite Communication Network," *NASA Tech Briefs*, LEW-19264-1, Cleveland, Ohio, December 2015.

[11] Zinnecker, A. M., Chapman, J. W., Lavelle, T. M., and Litt, J. S., "Modeling a twin-spool turbofan engine using the Toolbox for Modeling and Analysis of Thermodynamic Systems (T-MATS)," *AIAA Propulsion and Energy Forum and Exposition 2014: 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, AIAA-2014-3930, Cleveland, OH, July 2014.

[12] Moll, A. V. and Behbahani, A., "Comparison of Communication Architectures and Network Topologies for Distributed Propulsion Controls," *59th International Instrumentation Symposium*, WPAFB, OH 45433, 2013.

[13] Yedavalli, R., Willett, M., and Behbahani, A., "The Role of Various Real-time Communication Data Bus for Open System Distributed Engine Control Architectures for the Future," *Proceedings of the 47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, AIAA 2011-6145, San Diego, CA, July 2011.

[14] Gwaltney, D. and Briscoe, J., "Comparison of Communication Architectures for Spacecraft Modular Avionics Systems," June 2006, NASA/TM-2006-214431.

[15] Rushby, J., "A Comparison of Bus Architectures for Safety-Critical Embedded Systems," SRI International, September 2001.

[16] Smith, S. W., *The Scientists and Engineer's Guide to Digital Signal Processing, 2nd Edition*, California Technical Publishing, San Diego, California, 1999.

[17] Jonathan Kratz, Dennis Culley, J. C., "Approximation of Engine Casing Temperature Constraints for Casing Mounted Electronics," *52nd AIAA/SAE/ASEE Joint Propulsion Conferencen*, Salt Lake City, Utah, July 2016.

[18] Belapurkar, R. K., *Stability and Performance of Propulsion Control Systems with Distributed Control Architectures and Failures*, Ph.d. thesis, The Ohio State University, 2012.

[19] Driscoll, K., *Fast Software Encryption: 9th International Workshop, FSE 2002 Leuven, Belgium, February 4–6, 2002 Revised Papers*, chap. BeepBeep: Embedded Real-Time Encryption, Springer, Berlin, Heidelberg, 2002, pp. 164–178.

[20] Venugopalan, R., Ganesan, P., Peddabachagari, P., Dean, A., Mueller, F., and Sichitiu, M., "Encryption Overhead in Embedded Systems and Sensor Network Nodes: Modeling and Analysis," *Proceedings of the 2003 International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, CASES '03, ACM, New York, NY, USA, 2003, pp. 188–197.

[21] Shila, D. M., Venugopalan, V., and Patterson, C. D., *Unraveling the Security Puzzle: A Distributed Framework to Build Trust in FPGAs*, Springer International Publishing, Cham, 2015, pp. 95–111.