AFRL-RQ-WP-TP-2013-0209

COMPARISON OF COMMUNICATION ARCHITECTURES AND NETWORK TOPOLOGIES FOR DISTRIBUTED PROPULSION CONTROLS (PREPRINT)

Alex Von Moll and Alireza R. Behbahani

Engine Systems Branch
Turbine Engine Division

MAY 2013

Approved for public release; distribution unlimited.

See additional restrictions described on inside pages

STINFO COPY

AIR FORCE RESEARCH LABORATORY
AEROSPACE SYSTEMS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the USAF 88th Air Base Wing (88 ABW) Public Affairs Office (PAO) and is available to the general public, including foreign nationals.

Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RQ-WP-TP-2013-0209 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//
ALIREZA R. BEHBAHANI
Project Manager
Engine Systems Branch
Turbine Engine Division

//Signature//
CARLOS A. ARANA, Chief
Engine Systems Branch
Turbine Engine Division
Aerospace Systems Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

*Disseminated copies will show "//Signature//" stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE ***(DD-MM-YY)*** May 2013	**2. REPORT TYPE** Conference Paper Preprint	**3. DATES COVERED** ***(From - To)*** 01 December 2012 – 16 May 2013

4. TITLE AND SUBTITLE COMPARISON OF COMMUNICATION ARCHITECTURES AND NETWORK TOPOLOGIES FOR DISTRIBUTED PROPULSION CONTROLS (PREPRINT)	**5a. CONTRACT NUMBER** In-house
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 62203F
6. AUTHOR(S) Alexander Von Moll and Alireza R. Behbahani	**5d. PROJECT NUMBER** 3066
	5e. TASK NUMBER N/A
	5f. WORK UNIT NUMBER Q0K4
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Engine Systems Branch (AFRL/RQTE) Turbine Engine Division Air Force Research Laboratory, Aerospace Systems Directorate Wright-Patterson Air Force Base, OH 45433-7542 Air Force Materiel Command, United States Air Force	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFRL-RQ-WP-TP-2013-0209
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Aerospace Systems Directorate Wright-Patterson Air Force Base, OH 45433-7542 Air Force Materiel Command United States Air Force	**10. SPONSORING/MONITORING AGENCY ACRONYM(S)** AFRL/RQTE
	11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) AFRL-RQ-WP-TP-2013-0209

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

PA Case Number: 88ABW-2013-2323; Clearance Date: 13 May 2013. This paper contains color. This is a work of the U.S. Government and is not subject to copyright protection in the United States.

The conference paper was presented at the 59th International Instrumentation Symposium and MFPT 2013 Joint Conference, held in Cleveland, Ohio on May 13 through May 17, 2013.

14. ABSTRACT

A distributed engine control system (DECS) offering flexibility and scalability is envisioned for the next generation of propulsion controls. Perhaps the most touted benefit of a DECS is the potential to reduce the amount of harnessing which connects throughout the engine. Such a system is comprised of several network sections incorporating control nodes or data concentrators (DCs). These DCs contain control logic to perform control function computations and are connected to the full authority digital engine control (FADEC) via a high-speed data communication bus. A novel approach for analyzing and evaluating three topologies–ring, star, and bus–in the context of a relevant military engine was described. In this study, the algorithm uses a particle swarm optimization process to evolve solutions to a multi-objective optimization problem. The results of this study indicate there is potential for large wire length savings in a distributed control architecture.

15. SUBJECT TERMS

network topologies for distributed propulsion controls; control of high-performance military aircraft; propulsion systems of the future; communication network; communication architecture; optimization algorithm; ring, star, and bus topologies

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** (Monitor)
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	SAR	48	Alireza R. Behbahani **19b. TELEPHONE NUMBER** *(Include Area Code)* N/A

Comparison of Communication Architectures and Network Topologies for Distributed Propulsion Controls

Alex Von Moll [*]   Al Behbahani [†]

Air Force Research Laboratory
WPAFB, OH 45433

Abstract

A distributed engine control system (DECS) offering flexibility and scalability is envisioned for the next generation of propulsion controls. Perhaps the most touted benefit of a DECS is the potential to reduce the amount of harnessing which connects throughout the engine. Such a system is comprised of several network sections incorporating control nodes or Data Concentrators (DCs). These DCs contain control logic to perform control function computations and are connected to the Full Authority Digital Engine Control (FADEC) via a high speed data communication bus. The short term distributed engine control configurations will be core-mounted, uncooled data concentrator with high temperature electronics with relatively low speed data communications between smart effectors and the data concentrator; and high temperature electronics, high speed communication bus between the data concentrator and the control law processor master FADEC. This type of configuration is referred to as partially distributed control. In the long term, all smart components will be embedded in their respective locations, and the master FADEC will be in the flight controller in a cooled environment. This is considered a fully distributed control.

The DCs are connected to many smart components (sensors, actuators, pumps, etc.) via a communication data bus which has lower bandwidth. Network topology describes the way in which the smart components, DCs, and FADEC are connected to each other. Communication constraints like limited bandwidth, time delays and packet dropouts can limit the performance of a distributed engine control system. The distributed engine control system must be made robust enough to meet these constraints and handle failures in the network. In a basic sense, the loss of one or several nodes should not interrupt the system control. In this paper, we present a brief overview of several network topologies and report the advantages and disadvantages for each topology. A method to quantitatively assess the system benefits of each topology will be explored last.

[*] Aerospace Engineer, AFRL / RQTE, 1950 5th St. WPAFB, OH
[†] Senior Aerospace Engineer, AFRL / RQTE, 1950 5th St. WPAFB, OH

1 INTRODUCTION

FADECs (Full Authority Digital Engine Control) have been in service since as early as 1982, making hydromechanical controllers a thing of the past [1]. Of course, technologies and equipment like the FADEC had to buy their way into aircraft platforms. The need for higher reliability, maintainability, and performance was and is the main objective of these technologies, as well as the primary criteria by which they were compared with existing technology. While these areas have been addressed, to some extent, by digital control and electronics, the overall complexity of the controls systems have increased. In addition, new problems have arisen including limitations due to thermal considerations, increased weight, and electromagnetic interference (EMI) susceptibility. As modern jet engines continue to evolve, several newer trends have emerged: variable cycle technology and adaptive engine technology. The main idea is to optimize the performance of the propulsion system over the entire flight envelope [2], as well as diagnose and adjust to engine wear. This creates additional onus for an already over-burdened controller infrastructure as it must handle more control variables, effectors, and data with limited resources and a shrinking design envelope.

The concept of a distributed control architecture has been proposed [3, 4, 5] as a possible answer to some of the issues mentioned including performance, wider operability, and weight as well as reduced life cycle cost [5]. Currently, centralized controllers require a physical point-to-point connection to every sensor, pump and actuator. Thus the placement of the controller has a major influence on the overall wiring length, which is a major weight factor. Moreover, the location of the controller is constrained by the environment (e.g. temperature, vibration, etc.), the engines volumetric envelope, and accessibility for maintenance. By distributing functions such as analog to digital conversion (ADC), data processing, and diagnostics throughout the engine many point-to-point connections can be eliminated. For a large aircraft engine, this networked approach has been estimated to save 45 kg of weight in wiring [6]. However, there are many undeveloped, requisite elements which inhibit the implementation of distributed engine controls systems (DECS). They have been identified as [5]:

1. **High Temperature Electronics** – smart sensors, actuators, and pumps with embedded electronics that can withstand temperatures from -70°F to 600°F

2. **Collaboration & Standards** – definitions of specifications for common building blocks (smart nodes, interfaces, connectors, etc.) and collaboration between government, industry, and academia

3. **Communication Network** – a network to facilitate resource sharing and data passing for control and prognostic functionality

This paper seeks to explore the communication network by comparing both network topologies and communication architectures with DECS in mind. Network topology describes the specific physical or logical arrangement of elements (i.e. nodes) in a network [7]. Physically speaking, the topology can have a significant impact on system weight, reliability and redundancy, and maintainability. As previously mentioned, weight is affected by wiring which runs throughout the engine. By changing how the various components (and smart components) are connected, the wiring weight can be changed. Smart components can communicate bi-directionally in networked control systems. Smart components are data concentrators (DCs), valves, pumps, actuators, and sensors with expanded functionality. Obviously, we are mainly interested in configurations that reduce the system weight compared to the current centralized architecture. Thus sparse networks with fewer connections will tend to reduce more wiring weight. Reliability and redundancy, on the other hand, is improved by increasing the number of connections between nodes. This is especially important in terms of system availability, fault isolation, and robustness. If one node fails, for example, all other nodes in the system must be able to continue functioning. Currently, FADECs present a single point of failure for engine control systems, which is why they are typically dual-redundant systems. Network topology can also affect the system maintainability. The more connectors and cables there are, the costlier the system will be to maintain. In addition, the location of each node and cable affects maintainability in the sense of physical access to the maintainer (i.e. proximity to engine maintenance doors) [8]. The basic network topologies considered in this paper are [7]:

1. **Ring** – each node is connected to the node just before and just after itself
2. **Star** – each node is connected to a central node or hub
3. **Bus** – each node is connected to shared backbone medium
4. **Tree** – nodes are arranged in a hierarchical fashion, there is a root node, interior nodes, and leaves (nodes with no children)

There are advantages and disadvantages of each topology with regards to the aforementioned design considerations. In some cases, these topologies can be combined to form a sort of hybrid topology [9]. For network designs specifying actual components and their estimated mean time between failures (MTBF), fault-tree analysis can be used to compute system availability, which is a useful metric for comparing network topologies [10].

In addition to the network topology, a DECS implementation must also specify a communication architecture. The communication architecture describes both the communication protocol and physical medium of the network. Much work has been done already to compare various communications architectures for aerospace applications [11, 12, 13]. In general, the communication architecture must handle or specify node connectivity, message handling, timing and synchronization, media access control (MAC), error checking, and fault detection, isolation, and recovery (FDIR). There are two main schemes for determining when messages can be sent across the network. Time triggered systems define specific time slots for communication, thus each node has a particular time to send messages across the network. Event triggered systems, like the CANbus (controller area network bus), which is popular in the automotive industry, do not send messages at specified times, but rather rely on external events. Several protocols have provisions for both synchronous and asynchronous communication (i.e. time-triggered and event-triggered). The overall system-level communications requirements for a DECS application are given as [13]:

1. Real-time communication and determinism
2. Support for architectural composability

3. Synchronization
4. Dependability
5. Low complexity
6. Efficient redundancy management
7. Scalability (in terms of data transfer, number of nodes, etc.)

Determinism is important from the standpoint of the controller; the stability of the system must be able to be analyzed and verified. Architectural composability is important to the modularity of the system and how easily changes can be made to the system architecture. For example, adding or removing a node should not result in a complete reassessment of the communication performance and should have minimal impact on ex isting communication patterns. Systems that are highly modular and reusable help to mitigate issues related to recertification and change management. In addition, dependability and redundancy are desired, but they ought not consume too many resources. At the communication architecture implementation level there are even more challenges [14, 15]:

1. **Network-induced time delay** – delay inherent to message transport across the network
2. **Jitter** – variability in message delivery times
3. **Packet dropouts** – messages that are not delivered due to packet collision, unavailability of the network, or a transmission error
4. **Bandwidth** – the maximum rate at which data can be transferred is constrained both by the medium and, in some cases, the protocol

Section 2 of this paper contains a more thorough discussion on topology. Section 3 elaborates on some of the design considerations and past work for the communication architecture. A quantitative study of applicable topologies using a stochastic, multi-objective optimization algorithm for the placement of DCs on an engine is contained in Section 4. Lastly, some concluding remarks on the study and this paper as a whole are included in Section 5.

2 TOPOLOGY DISCUSSION

As previously mentioned, this paper will discuss the basic ring, star, bus, and tree topologies in the context of a DECS implementation. They will be analyzed in terms of their potential impact on weight, network reliability, and maintainability. It is also important to note the effect a given topology may have on the average message transmission length. This discussion will be primarily qualitative, as the quantitative case studies will be presented later in Section 4.

2.1 RING

The ring topology describes a network in which each node has two connections. Generally these connections are to the node that comes before and the node that comes after. This is simple to visualize in the case where nodes are already arranged in a ring-like shape, and each node connects to its two nearest neighbors. However, the physical shape of the network could end up looking quite contorted depending on the locations of the nodes. The ring topology is considered to be weakly connected since for a fully connected network of n nodes there are $\frac{1}{2}n(n-1)$ connections while there are only n connections in a ring; thus it may have a large potential for weight savings. In some simple ring networks, data is passed in only one direction (clockwise, or counterclockwise), but the authors will consider a ring that is counter-rotating (i.e. data may be passed in either direction). This means that the failure of a single node or link in the ring would not cause an entire system failure since there are always two paths between each node. Since the connections in a ring network do not terminate at a central location, accessibility to the maintainer might be difficult. The average message transmission length is relatively high for the ring topology since a message may have to traverse a sizeable portion of the ring to reach its destination. In light of this issue, there has been some research on how to improve scalability in terms of transmission time for ring networks [16].

2.2 STAR

In a star configuration, each node connects directly to a central node, generally referred to as the hub. The current centralized control architecture may be interpreted as a physical star topology where the FADEC represents the central node. In the case of a distributed architecture, there may be several stars chained together. While, in principle, the total number of links in a star topology is similar to the ring topology, the wire length tends to be larger. This is the result of the point-to-point nature of the star topology; on average, the exterior nodes are closer to each other than the hub. The hub represents a single point of failure for all nodes connected to it. However, the loss of any exterior node has no effect on the rest of the network. In this sense, the need for reliability (strictly in terms of network connectivity) is shifted away from the exterior nodes towards the nodes acting as hubs. Maintaining a DECS in a star topology could prove to be easier than a ring topology since the connections all terminate at the hubs. These hubs could be placed purposefully for ease of access to the maintainer. Though each exterior node has only one connection in this configuration, the hub has many. Exterior nodes are logically connected via the hub, so there are generally only two links that a message must traverse.

2.3 BUS

For the purposes of this paragraph, the word bus refers to the physical bus network topology as opposed to the data transfer mechanism (i.e. databus) meaning, which is taken throughout much of this paper. The bus topology is characterized by a shared communication backbone. Each node connects exclusively to the bus, instead of to other nodes, like in the other topologies. The physical bus topology shares many advantages and disadvantages with the star in terms of maintainability and fixed message transmission length. However, there is a potential to reduce the number of connectors, compared to the star case. Consider the case where there is a single hub in the center of the engine. A multitude of links converge on the location of the hub. Then consider a bus, which could extend along the length of the engine. Here, each node requires just one connector. Although both the bus and the star configurations present a single point of failure, the bus may be more susceptible to physical damage because it spans over a much greater portion of the engine.

2.4 TREE

The tree topology consists of a root node that is connected to each of its children, who may have their own children. Thus each node is connected to a single parent and each of its children (except for the root, which has no parent). Trees are often classified by the number of children allowed. For example, a binary tree is a tree in which each node may have up to two children. A tree structure is inherently hierarchical, which could prove useful as control systems become more hierarchical. This is particularly appropriate for supervisory and control tasks in which a high-level controller sends commands to low-level controllers [17]. The tree structure may also be suitable in cases where the controller is partitioned in a functional manner. In [18] the authors propose using localized control nodes, one for the inlet/fan, compressor, combustor, and turbine/nozzle. Thus it is clear that the tree architecture has a clear advantage in scalability. In other words, it has the ability to add nodes without increasing the capacity requirement of each node and without significantly lengthening message transmission delays [19]. Performing maintenance on a network arranged as a tree could be difficult if the network reaches over a large area. In many ways, the tree topology is comparable to a chain of stars.

2.5 PREVIOUS WORK, CONSIDERATIONS GOING FORWARD

It is clear that each basic topology has its strengths and weaknesses. The merit of each topology depends heavily upon the application in addition to whatever system-level assumptions can be made at this point. Table 1 summarizes the analysis presented in this section.

Table 1: Summary of qualitative topology analysis – green markers indicate potential benefits, red markers indicate potential challenges, and yellow markers are neutral

	Ring	Star	Bus	Tree
Weight	🟩	🟨	🟩	🟨
Reliability	🟥	🟨	🟥	🟥
Maintainability	🟥	🟨	🟨	🟥
Msg TX Length	🟥	🟩	🟩	🟩

There is a potential to employ a hybrid topology by combining multiple topologies in a way that captures their advantages while avoiding their drawbacks. One example, that has been mentioned already, is a multi-star configuration where two or more stars are chained together. In [20] the author considers a dual star configuration along with single star, single ring, and dual ring in terms of system unavailability, initial and life cost, diagnostic ease, and data transfer for electric power substation communications. Though the initial cost for the dual star configuration was estimated to be much higher than the others, it won in every other category. The application for the power substation study is far different from the application being considered in this paper; however, the methods and analyses used to compare the topologies as well as the results are, to some extent, pertinent.

In [13] the authors consider a particular network topology for the DECS application. They describe a braided-ring topology in which a node is connected to its neighbors in addition to its neighbors neighbors along the ring. The reported advantages of this type of topology are given as follows:

1. High reliability in a deployment scenario
2. Robustness against physically localized damage
3. Robustness against node dropout
4. Mitigation of physical layer composability

It is also mentioned that it is possible to change the physical medium between ring segments. This last statement is potentially true for almost any network topology. The topic of physical medium is typically more closely coupled with communication architecture considerations, which will be discussed in more detail in Section 3. It is clear, given the two topology examples above, that the best topology largely depends on the application and its system requirements. In general, the best topology will be one which establishes the best balance between weight (or cost), reliability, redundancy, and maintainability.

3 COMMUNICATION ARCHITECTURE

The question of which communication architecture to use for a distributed engine control scenario appears in almost every paper concerning DECS. The discussion of communication architecture generally includes the topics of media access control, protocol, and physical medium as well. In essence, these discussions seek to define the elements of the Open System Interconnection (OSI) network model for the engine control application. Figure 1 summarizes the OSI network model, which is comprised by 7 layers [21].

In addition to the requirements and constraints enumerated in the Introduction of this paper it is also important to consider the failure scenarios for the network communication. Understanding the failure scenarios and their relation to the DECS application is important for assessing the merit of the various architectures and protocols. In [22], the author identifies the following failure scenarios:

1. **Outgoing link failure** – node can't send a message (either failure of logic or aging of components)
2. **SOS failure** – (Slightly-Off-Specification) node produces an output signal slightly outside the specified window (i.e. on the edge of what the receiver understands as 1 or 0)

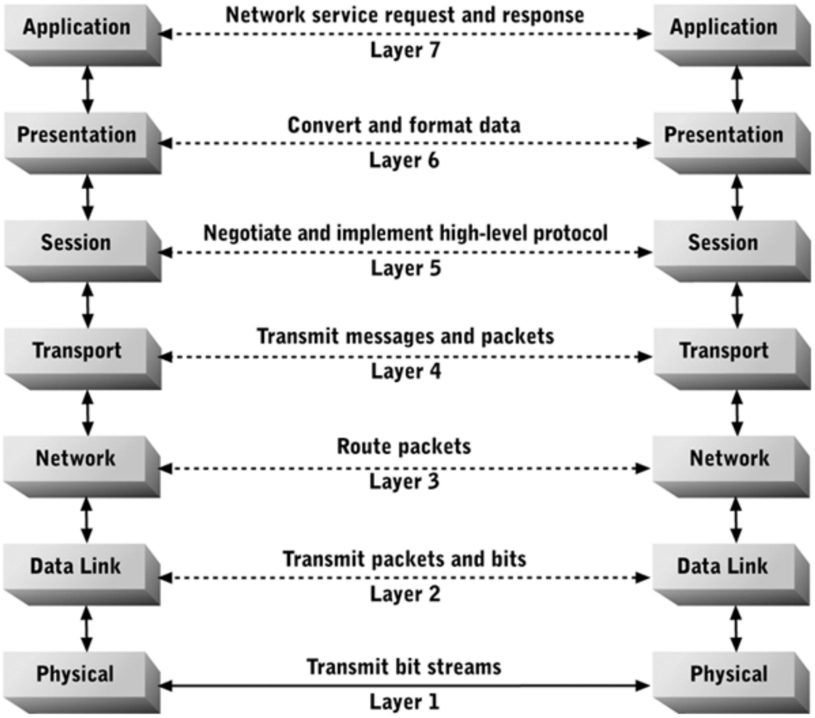


Figure 1: OSI Network Model [21]

3. **Spatial proximity failure** – nodes in the same spatial region affected be some physical damage

4. **Masquerading failure** – a node takes the identity of another node, which may be undetected

5. **Babbling idiot failure** – node doesnt observe the rules of the protocol, may attempt to send messages at arbitrary times; bus guardians are typically used in safety critical applications to mitigate this type of failure

A DECS must be able to handle these types of failures and continue to function.

In the past, much consideration has been given to whether the system should be event-triggered or time-triggered. Several time-triggered protocols (TTP) have been implemented and considered for a DECS application including SAFEbus [23], TTP [24], SPIDER [25], and TTCAN [26] [12]. In general, TTPs require a significant amount of initial design to create the message schedule model and do not allow the addition of nodes or messages without redesigning the message and task schedule [12]. However, the topic of fair and efficient scheduling for time-division multiple access (TDMA) transmission is becoming more prevalent in the research community [27]. Event-triggered protocols have also been considered, including CANbus[28], Ethernet, and AFDX [29] (which is currently in service on the Airbus A380). These protocols must incorporate some sort of contention resolution mechanism since it is possible for multiple nodes to attempt to transmit messages simultaneously [11]. In [12], NASA considered the Honeywell SAFEbus (the backplane data bus used in the Boeing 777 Airplane Information Management System) and the NASA SPIDER (an architecture being developed as a demonstrator for certification under the new DO-254 guidelines) as well as the TTTech Time-Triggered Architecture (TTA) and FlexRay [30] architectures. The latter two have been used by the automobile industry. Since the NASA studies, Honeywell and Hamilton Sundstrand have used TTP technologies in the MAC FADEC, and Boeing 787 applications, respectively [4]. In both [12] and [22] TTP/C (an implementation of TTP) was identified as having many advantages over the alternatives. The TTP/C protocol is expected to be considerably less expensive to implement due to reduced number of components required afforded by its overall robustness [22] (since fewer or no bus guardians would be needed). Additionally, TTP/C is deterministic, supports upgrading and swapping of modules, is masterless (allowing for single-fault tolerance), is independent of the physical layer, and implements the protocol components on the hardware simplifying software development [12]. Physical layer independence is an important concept for the DECS application since different stretches of cabling may be immersed in different EMI, vibration, and temperature environments. Thus, at the time of writing, a time-triggered architecture similar to the TTP/C protocol seems to be the forerunner, though there are many alternatives that could prove to be useful. It should be noted that it's the overall system architecture, not necessarily the physical topology, that drives the selection of a communication protocol. More elaboration on the architecture could enable a more detailed analysis of the various protocols.

4 DATA CONCENTRATOR LOCATION CASE STUDY USING MULTIOBJECTIVE OPTIMIZATION ALGORITHM

While the discussion, up to this point, has been mainly qualitative, this section is devoted to quantifying and visualizing the comparison between the different topologies. The application considered in this study is a relevant (or what may be considered to be current) military engine. Conceptually, the study estimates the effect of adding data concentrators, in a particular topology, to the existing engine configuration while attempting to minimize several different objectives. Overall wiring length, number of components placed in high temperature regions, and number of data concentrators added are the objectives that the study seeks to minimize. This minimization entails finding the optimal location to place each data concentrator. Below, the assumptions that are built into the study are enumerated; in Section 4.2 the algorithm is described.

4.1 ASSUMPTIONS

There are many assumptions that are factored into this study, many of which are for the sake of simplicity. In some cases, the assumptions detract from the realness of the study; however, with such a complicated system, with so many variables, it is easier to generate and interpret the results by using this simplified approach. Existing control components, including sensors, pumps, actuators, and the FADEC, are assumed to have known and fixed locations. In other words, the study does not seek to optimize the locations of components already present in the configuration under consideration. The physical topology, number of DCs, and the locations of the DCs are variable. Whether an existing control component is smart or not (i.e. has its own controller or signal processing capability onboard) bears no significance for the purposes of this study. It is assumed that each component must have a point-to-point connection to a FADEC or DC. Thus the differences in topologies only occur in the DC-to-DC, DC-FADEC connection style. In order to build in some redundancy, each component will connect to the two nearest DCs (or the nearest DC and the FADEC, if the FADEC is closer than the next nearest DC). The true wiring length of this particular engine configuration is not known. A wiring length baseline, for the current centralized control architecture, is estimated by summing the distance from each component to the FADEC twice (once for each channel) and scaling by the number of conductors. Although the surface temperature distribution of this particular engine is known, the exact relation between engine surface temperature and the required electronics temperature limits is unclear. Thus data concentrators that are placed on hotter parts of the engine have a higher penalty associated with them. In reality, this higher penalty could be associated with higher weight for possible thermal insulation or fuel cooling, or higher cost for developing and using higher temperature electronics, or a combination of these factors. A detailed analysis of the temperature capability for the data concentrator electronics and an estimate of the actual cost and/or weight added to the system is out of the scope of this study. Because the engine under consideration is a military engine with a small bypass ratio, the engine case is approximated using a cylinder. Many components are embedded deeper inside the engine or even in the gas path, but since each component must connect to a DC or FADEC on the engine case, this radial variation would be common among any configuration. In addition, it is assumed that radial distance would not contribute greatly to the overall cable length.

Another big assumption lies in how wiring is routed, and thus how wiring length is estimated. In reality, cables merge and split from various wiring harnesses, which must be routed around control components and external plumbing (hydraulic, fuel, bleed air, etc.). This can, in turn, affect which locations are considered optimal for component placement. However, component proximity to what it is connected to is one of the biggest location drivers. Thus the wiring length estimation follows a very simple approach: calculating the city block, or *Manhatten Distance* between the components. This distance is then scaled by the number of conductors required for the component being connected. In the case of a cylinder, with all the components located at the same radius, this means the cabling travels only circumferentially or axially, but never both simultaneously. Although the shortest path between two points on a cylinder is helical, it is obvious that engine cabling does not follow this approach. This assumption also benefits the computation side of the study, since the city block distance is less computationally intensive to calculate.

It is believed that reducing the number of direct connections and overall processing burden of the FADEC would also reduce the size and weight of the FADEC, however, this potential benefit is not directly considered in this study. As DCs are added to the configuration throughout the study, there is a point where the cost of adding an additional DC to the configuration will not be offset by a large enough savings in wiring length. This optimum number of DCs is a strong function of the actual cost, weight, and size of that DC, which does not currently exist. Thus the optimum number suggested by this study may not necessarily reflect the best possible configuration for a real-world implementation.

4.2 OPTIMIZATION ALGORITHM

The overall approach for this multi-objective optimization is similar to the one used in [31], which describes a tool for determining spacecraft avionics box placement. Like the avionics box placement problem, the DC placement problem is shown to be a variant of the classical Traveling Salesman Problem (TSP), which is of the class NP-hard, meaning it cannot be solved to optimality in polynomial time due to the enormity of the solution space [31]. Thus, problems that are considered NP-hard are generally good candidates for using an evolutionary or iterative algorithm to find an optimal solution in a reasonable amount of time.

Genetic algorithms, like the one used in [8] have gained a lot of popularity for a wide variety of different applications. However, encoding a problem for use with a genetic algorithm involves devising analogous operations for genetic crossover and mutation. There did not appear to be an intuitive way of capturing these kinds of operations for the DC placement problem. A more recently developed algorithm called Particle Swarm Optimization (PSO) [32] is chosen instead. PSO involves keeping track of many different potential solutions, or particles, as they fly around the solution space. Each particle has a position and velocity and is drawn towards the best solution found. The PSO combines elements of global search and local search – the particles remember their own best solution as well as the best solution found by any particle in the swarm. The value of a solution, or position, is determined by a fitness function, which is a mapping from positions in the solution space to scalar values. Depending on the number of particles used, PSO can explore the search space more thoroughly than some of the alternatives (e.g. simulated annealing, genetic algorithms). For the DC placement problem, each particles position encodes a configuration of DCs on the engine. A configuration consists of the position, in three dimensions, of each data concentrator.

$$x = [z_1,\ \theta_1,\ r_1,\ ...,\ z_n,\ \theta_n,\ r_n] \tag{1}$$

Where z is the axial position, θ is the angular position, and r is the radial position (considered constant in this study). Also, n is equal to the number of data concentrators; thus the solution space is $3n$-dimensional. Similarly, the particle has a velocity in each dimension:

$$v = [\dot{z}_1,\ \dot{\theta}_1,\ \dot{r}_1,\ ...,\ \dot{z}_n,\ \dot{\theta}_n,\ \dot{r}_n] \tag{2}$$

A major benefit of the PSO is that it does not require differentiability or knowledge of the derivatives. Instead, the PSO essentially uses a difference equation to update a particles current position, while the velocity is updated using information about the best solutions found so far [33].

$$x_i = x_i + v_i \tag{3}$$

$$v_i = \omega v_i + \phi_p r_p (p_i - x_i) + \phi_g r_g (g - x_i) \tag{4}$$

Where

- subscript $i \equiv$ reference to a particular particle
- $\omega \equiv$ inertial weighting
- ϕ_p, $\phi_g \equiv$ cognitive and social weighting, respectively

- p_i, g $\equiv$ 'personal' and global best solutions found so far, respectively
- r_p, r_g $\equiv$ random numbers on the interval $[0, 1)$

At the end of each update, the position is fed into the fitness function:

$$f_i(x_i) = \sum_{k=1}^{c} r_k(\min_1(d_k) + \min_2(d_k)) + \sum_{j=1}^{n} w_j \tag{5}$$

Where

- subscript k $\equiv$ reference to a particular component (sensor or actuator)
- subscript j $\equiv$ reference to a particular DC
- d_k $\equiv$ a vector containing the distances from component k to each DC and the FADEC
- $\min_1$, $\min_2$ $\equiv$ the minimum and next smallest, respectively
- r_k $\equiv$ the number of conductors for component k
- w $\equiv$ DC cost term incorporating temperature penalty and DC connections
- c, n $\equiv$ the total number of DCs and sensors/actuators, respectively

Since this fitness is really a measure of overall wiring length, weight, and temperature penalties, the algorithm seeks to minimize these values. It should be noted that the overall wiring length includes the DC databus length in addition to the wiring length of all the sensors and actuators.

An important observation to be made at this time is the fact that the number of dimensions increases rapidly with the number of DCs; a configuration with 4 DCs is encoded by particles with 12 dimensions, for example. Thus the shape of the solution space, with regards to fitness, may be very complex with many geographic features, requiring fine granularity. In this situation, there is a potential for an abundance of local optima. If one of these local optima is found before the true global optimum is explored (which is a highly likely scenario) then the particles may converge to this location without fully exploring the search space and reaching the global optimum. This raises the question of how to scale the velocity of the particles, or, in other words, how to adjust the w, ϕ_p, and ϕ_g terms from Equation (5). If these terms are very small then the algorithm must perform many iterations to explore enough of the search space, taking up a lot of time. On the other hand, if the velocity is too large particles may miss a lot of the finer features of the solution topology. In order to avoid some of the problems associated with premature convergence and local optima the baseline PSO is extended to a more robust Local Optima Avoidable PSO (LOAPSO) [34]. At each iteration, a random subset of the swarm population is *repulsed* instead of attracted to the personal and global best solutions. This effectively adds more randomness to the algorithm and encourages exploration to avoid premature convergence.

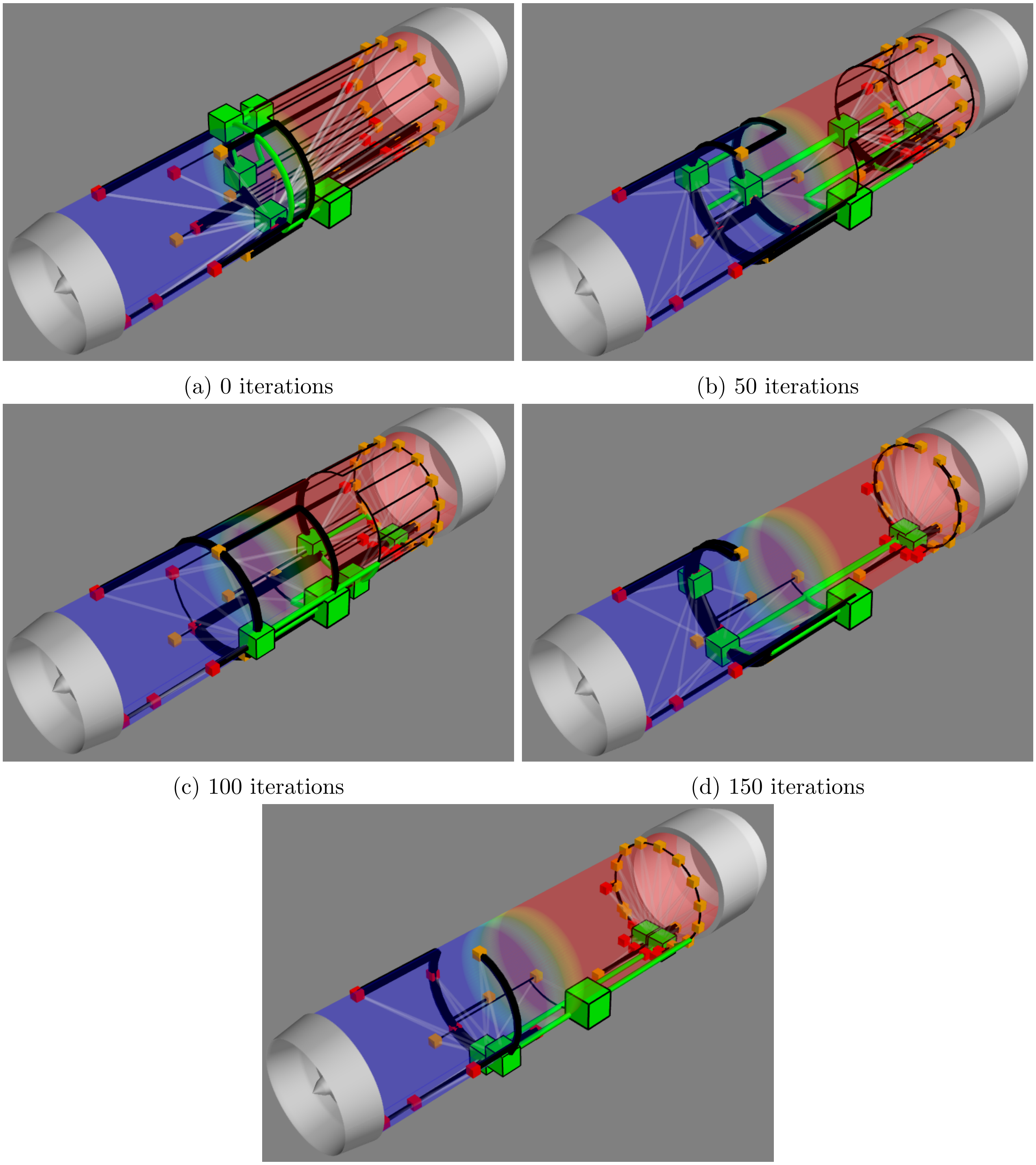
4.3 RESULTS

As previous sections allude to, the result of the algorithm is a configuration of DCs for a military engine with sensors and actuators already in place. Figure 2 shows the evolution of a configuration of four DCs connected in a ring topology for one particle in the swarm. In each figure, the front of the engine is on the left and the red and blue sections of the semitransparent cylinder represent the hot and cold portions of the engine case, respectively. The large green box in the center of the engine represents the FADEC while each of the smaller green boxes represents a DC. Sensors and actuators are represented by small red and orange boxes, respectively. The wires connecting these sensors and actuators are represented by black tubes around the outside of the cylinder. Each tubes thickness is based on the number of conductors needed for that component the more conductors, the thicker the tube. Faint white tubes show a straight line connection for some added clarity on which component is connected to which DC – they do not represent any physical entity. Finally, there are thick green tubes representing the high speed databus which connects the DCs and FADEC to each other in the chosen topology – a ring in this case. Note the similarity between the 150th and 200th iterations, this shows the algorithm converging to an optimal configuration. Figure 3 demonstrates this as well by showing the overall swarm's minimum fitness, wiring length, and average temperature penalties over each iteration.

The most important results of this study are contained in Figure 4. These plots illustrate how the total system is impacted by the number of DCs and the topology. One observation that can be made immediately is the relative system cost of the bus is consistently higher than the ring and star topologies. This is mostly due to a special constraint made for the bus topology which forces each DC to lie on the same angular position as the FADEC. Without this constraint, the bus would behave quite similarly to the ring, but with one less link between DCs. However, including this constraint shows that the controller network can be constrained spatially (a big advantage for accessibility and maintainability) and still offer cable length savings. It will become clear how the bus configuration looks in Figure 5.

Another observation that can be made on Figure 4 is the fact that the data for the ring architecture appears to be less smooth than the other lines. This brings to light a specific issue that is unique for the ring topology. In the ring topology, the order in which the DCs and FADEC are connected is important. Since the bulk of the fitness function is determined by the wiring between the DCs and the sensors and actuators, the wiring between the DCs themselves plays a very small part. Thus it is quite possible that the algorithm could converge on a configuration in which the green lines in the diagram (Figures 2, 5) are anything but optimal. Optimizing the order of connections for the ring topology is, in fact, analogous to the TSP. Coming up with a solution to this problem for each particle after each iteration was considered to be too computationally intensive and time consuming to warrant implementation. Instead, the baseline algorithm was run many times for each case, and the configuration with the minimum fitness was kept.

Based on the results in Figure 4 the star topology does not scale quite as well as the ring topology for larger numbers of DCs. For few DCs (up to about 4), the ring and star behave very similarly, however as more DCs are added, the relative system cost is slightly higher for the star topology. It is also evident that the best configuration uses 4 DCs. At this point, the wire length savings heavily outweigh the cost, weight, and temperature penalties. Overall, the total wiring length is shown to be reduced by nearly 60% when 7 or more DCs are used.



(a) 0 iterations

(b) 50 iterations

(c) 100 iterations

(d) 150 iterations

(e) 200 iterations

Figure 2: Evolution of one particle's configuration for 4 DC ring topology

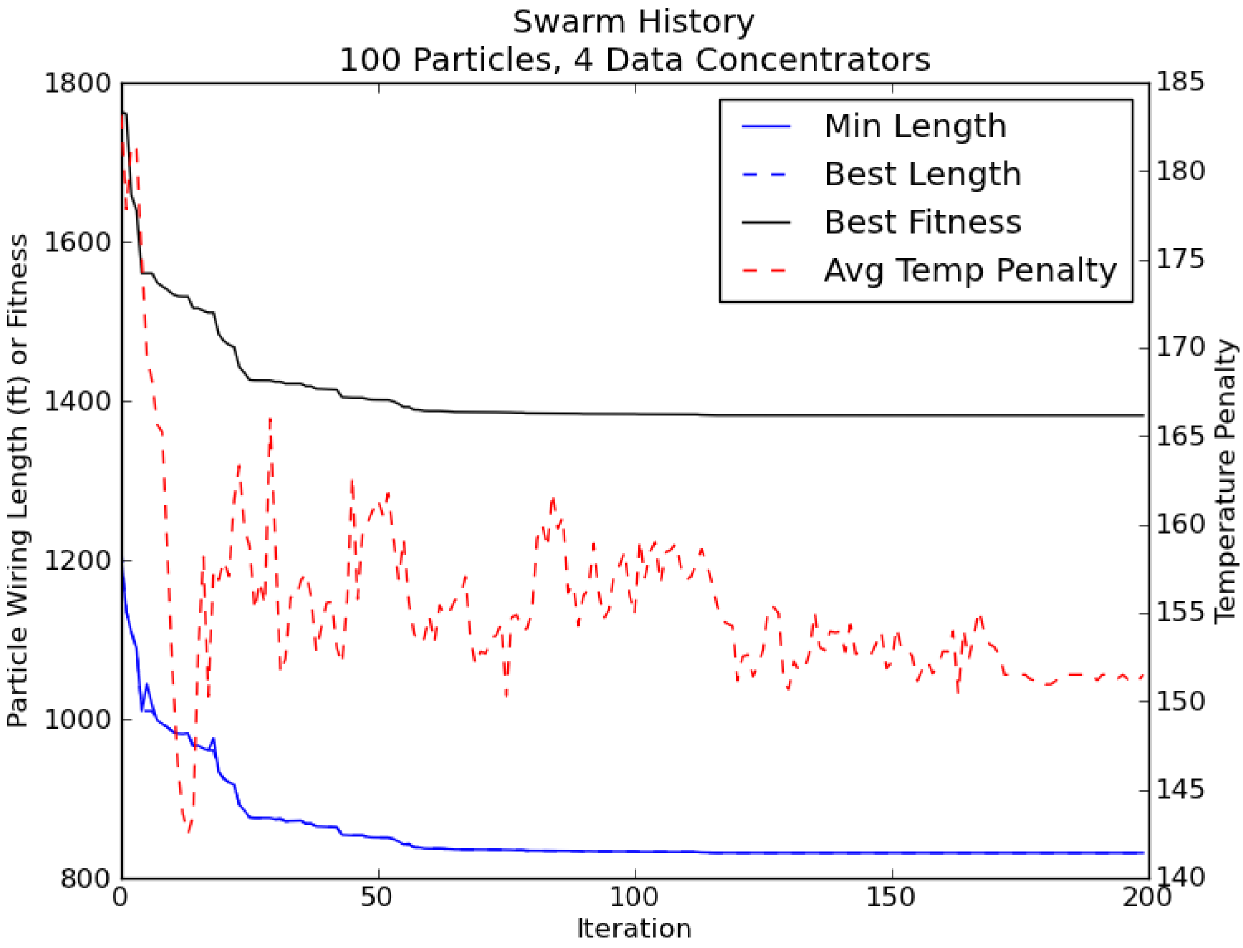


Figure 3: Iteration history for the entire swarm with 100 particles in a 4 DC ring topology

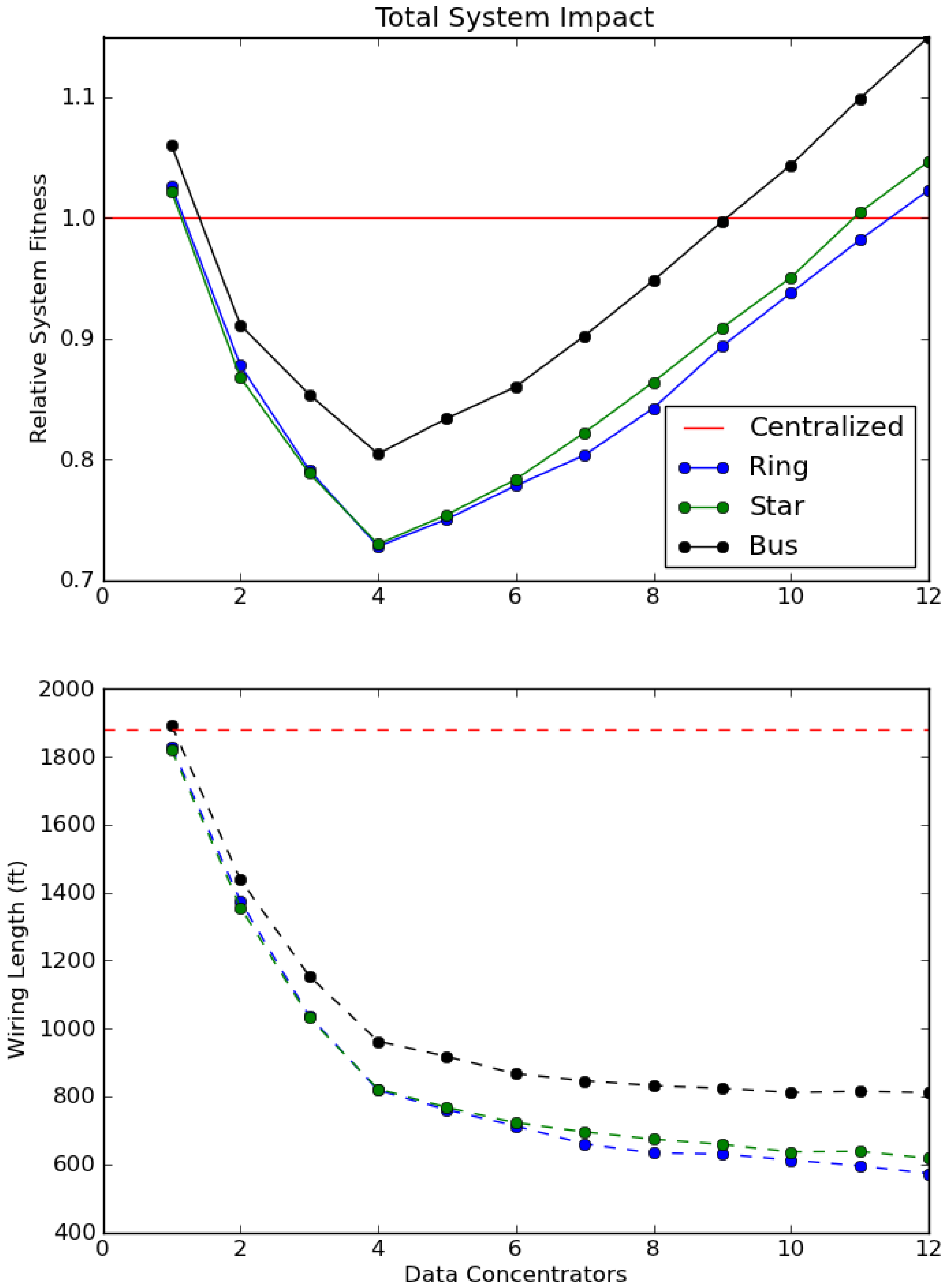
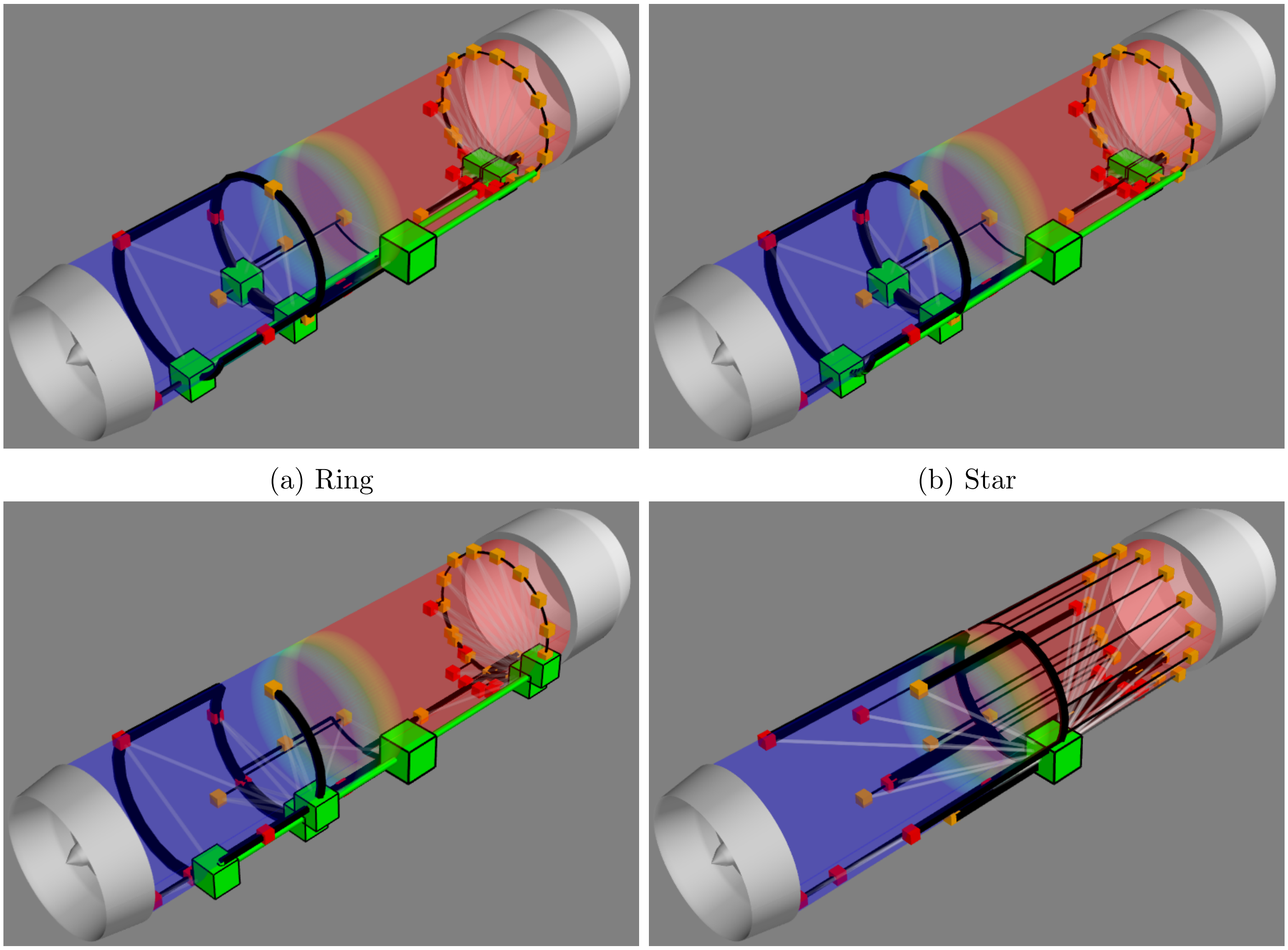


Figure 4: System level results – fitness and wiring length as a function of the number of DCs in the configuration



(a) Ring

(b) Star

(c) Bus

(d) Baseline (centralized)

Figure 5: Distributed engine control configurations with 5 data concentrators (a–c) and a baseline configuration (d)

5 CONCLUSION

A distributed engine control system is envisioned for the next generation of propulsion controls. Although the concept of distributed control, from a functional standpoint, has been explored many times in recent literature there are still many unanswered questions associated with the implementation of such a system. In this report, we sought to shed some light on two of the requisite items: the physical topology of the networked control system, and the communication architecture. The discussion on communication architecture summarizes much of the work that has been done in this area, both inside and outside of the engine controls community. This report focused much more on the topology considerations. A novel approach for analyzing and evaluating three topologies: ring, star, and bus in the context of a relevant military engine was described. The algorithm uses a Particle Swarm Optimization process to evolve solutions to a multi-objective optimization problem. The results of this study indicate there is potential for large wire length savings in a distributed control architecture even when reliability (in this case, connections to two different data concentrators) is considered. It is shown that the ring architecture scales better with larger numbers of DCs than the star and bus. Based on the assumptions of the study, including estimating the system cost of each DC and temperature penalties, it is shown that a configuration using 4 DCs is the most advantageous. In the future, as more information becomes available, this type of study could be updated to have more fidelity. For example, the impact of these configurations on weight should be investigated. In addition, the size of the engine, number of sensors and actuators, and their locations could be varied to investigate potential benefits for different types of engines. There is a potential for even more savings if the actual locations of sensors and actuators themselves could be optimized. This will require higher temperature capable electronics. Lastly, as more components on the engine begin to incorporate more signal processing capability, the style of the sensor/actuator-to-DC connections could be varied as a part of the study on topology.

ACKNOWLEDGMENTS

The authors would like to thank the following individuals for their input in the form of discussions, comments, and technical data: John Fisher, Brendan Hall, Pat Ethier, John Oerther, Jack Hoying, and Ken Semega.

REFERENCES

[1] Harrier flies with digitally controlled pegasus. *FLIGHT International*, 1982.

[2] Larine Barr. Air Force plans to develop revolutionary engine. *USAF News*, 2007.

[3] Bruce Wood, Derek Weber, Bill Rhoden, Bill Mailander, Gary Hunter, Dennis Culley, Casey Carter, Dewey Benson, and Al Behbahani. Transition in gas turbine engine control system architecture: Modular, distributed, embedded. Technical report, DTIC Document, 2009.

[4] Al Behbahani. Adaptive distributed intelligent control architecture for future propulsion systems. In *61st Meeting Of The Society For Machinery Failure Prevention Technology, Virginia Beach, Virginia, Paper: Session 5B-Cape Colony B*, 2007.

[5] Alireza Behbahani, Dennis Culley, BJ Smith, et al. Status, vision, and challenges of an intelligent distributed engine control architecture. *SAE Paper*, pages 01–3859, 2007.

[6] Mirko Jakovljevic, Leonard Fulcher, and Dewey Benson. Expectations and vision for true modular distributed engine control–beyond 1st project. In *44th AIAA Joint Propulsion Conference & Exhibit*, 2008.

[7] ATIS. ATIS Telecom Glossary 2012. http://www.atis.org/glossary/definition.aspx?id=3516.

[8] Duncan Bourne, Roger Dixon, and Alex Horne. Architectural design of distributed control systems for aero gas turbine engines using genetic algorithms. In *Systems Engineering (ICSEng), 2011 21st International Conference on*, pages 3–7. IEEE, 2011.

[9] LAN Network Topologies. http://www.firewall.cx/networking-topics/general-networking/103-network-topologies.html.

[10] Gary W Scheer. Comparison of fiber-optic star and ring topologies for electric power substation communications. In *proceedings of the 1st Annual Western Power Delivery and Automation Conference, Spokane, WA*, 1999.

[11] TTTech Computertechnik AG. Protocols for aerospace controls. Technical report, 2005.

[12] DA Gwaltney and JM Briscoe. Comparison of communication architectures for spacecraft modular avionics systems. *Marshall Space Flight Center, NASA/TM2006-214431*, 2006.

[13] Brendan Hall, Michael Paulitsch, Dewey Benson, and Alireza Behbahani. Jet engine control using ethernet with a brain (postprint). Technical report, DTIC Document, 2008.

[14] Joao P Hespanha, Payam Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.

[15] Rama Yedavalli, Rohit Belapurkar, and Alireza Behbahani. Stability analysis of distributed engine control systems under communication packet drop. In *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, 2008.

[16] Jae Young Choi. Evaluation of ethernet based control network used in the distributed control system. In *Proceedings on the 15th CISL Winter Workshop*, volume 1, page 2. Control Information Systems Lab.

[17] Esteban Gutiérrez Fernández et al. Management system for unmanned aircraft systems teams. 2011.

[18] Dennis Culley and Alireza Behbahani. Communication needs assessment for distributed turbine engine control. In *44th AIAA Joint Propulsion Conference & Exhibit, Hartford, CT*, 2008.

[19] Min Huang and Brett Bode. A performance comparison of tree and ring topologies in distributed systems. In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, pages 8–pp. IEEE, 2005.

[20] Gary W Scheer and David J Dolezilek. Comparing the reliability of ethernet network topologies in substation control and monitoring networks. In *Western Power Delivery Automation Conference, Spokane, Washington*, 2000.

[21] Stephen D Burd. *Systems Architecture*. Course Technology Ptr, 2005.

[22] Hermann Kopetz. A comparison of ttp/c and flexray. *Institut für Technische Informatik, Technische Universität Wien, Austria, Research Report*, 10:2001, 2001.

[23] K. Hoyme and K. Driscoll. Safebus. In *Digital Avionics Systems Conference, 1992. Proceedings., IEEE/AIAA 11th*, pages 68–73, 1992.

[24] Hermann Kopetz and Günter Grunsteidl. Ttp-a time-triggered protocol for fault-tolerant real-time systems. In *Fault-Tolerant Computing, 1993. FTCS-23. Digest of Papers., The Twenty-Third International Symposium on*, pages 524–533. IEEE, 1993.

[25] Alfons Geser and Paul Miner. A new on-line diagnosis protocol for the spider family of byzantine fault tolerant architectures. *NASA Technical Memorandum*, 2003.

[26] Gabriel Leen and Donald Heffernan. Ttcan: a new time-triggered controller area network. *Microprocessors and Microsystems*, 26(2):77–94, 2002.

[27] Sinem Coleri Ergen and Pravin Varaiya. Tdma scheduling algorithms for wireless sensor networks. *Wireless Networks*, 16(4):985–997, 2010.

[28] Robert Bosch. Can specification. 1991.

[29] AIM GmbH. Afdx training. 2010.

[30] FlexRay Consortium et al. Flexray communications system. *Protocol Specification Version*, 2, 2005.

[31] Bill Jackson. A stochastic optimization tool for determining spacecraft avionics box placement. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 5, pages 5–2373. IEEE, 2002.

[32] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.

[33] James McCaffrey. Artificial Intelligence - Particle Swarm Optimization. *MSDN Magazine*, 2011.

[34] SMA Salehizadeh, P Yadmellat, and MB Menhaj. Local optima avoidable particle swarm optimization. In *Swarm Intelligence Symposium, 2009. SIS'09. IEEE*, pages 16–21. IEEE, 2009.

Setting the Standard for Automation™

16 May 2013

Instrumentation - Aerospace Applications

Intelligent Turbine Engine Controls

Session E4 - Roxy

Comparison of Communication Architectures and Network Topologies for Distributed Propulsion Controls

Alex Von Moll

Alireza R. Behbahani, Ph.D.

ADVANCED INTEGRATED CONTROLS AND PROGNOSTIC /
DIAGNOSTIC SYSTEMS, Engine Systems Branch Turbine Engine Division
Aerospace Systems Directorate U.S. Air Force Research Laboratory
WPAFB, Ohio

Standards

Certification

Education & Training

Publishing

Conferences & Exhibits

59th International Instrumentation, Symposium – Cleveland, OH

Approved for public release; distribution unlimited.

Objectives

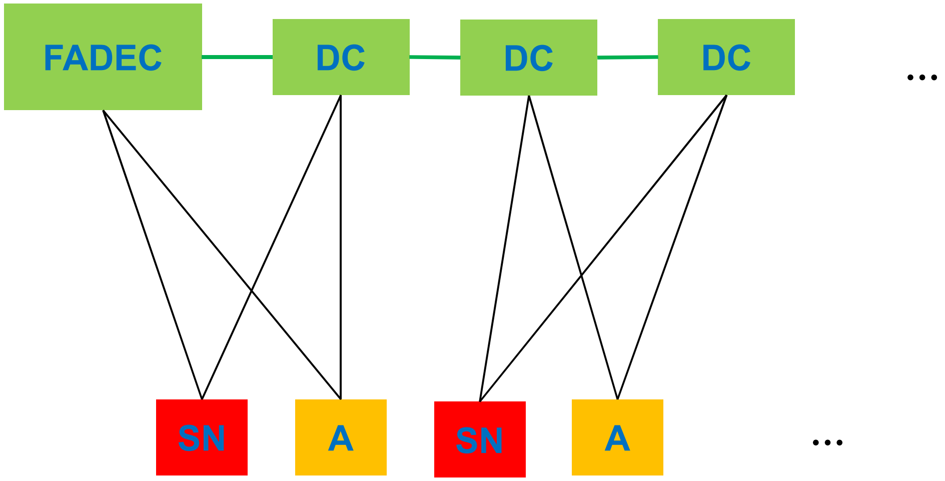
- Analyze several topologies with regard to distributed propulsion control considerations
- Review, compare, and summarize analysis on communication architectures and protocols
- Model, analyze, and optimize a potential architecture design

Approved for public release; distribution unlimited.

Potential Architecture

High Speed Databus

Low Speed Databus & Analog Wiring



FADEC

DC

DC

DC

…

SN

A

SN

A

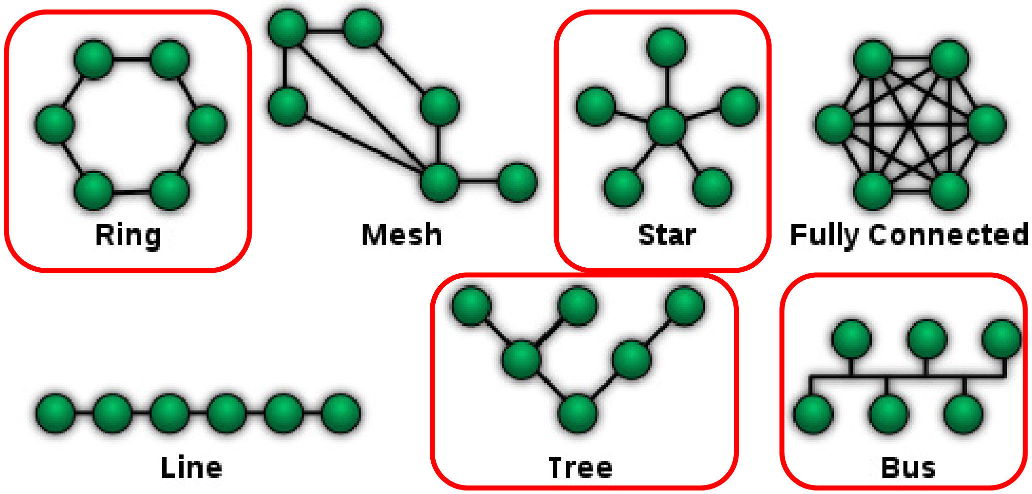
…

- Partially distributed control architecture comprised of an existing FADEC and sensor/actuator suite along with added Data Concentrators (DCs)
- Topological considerations only for the upper level

Topology

- Focus on physical topology
  - i.e. what is the physical structure of the network
- Establish criteria for the distributed propulsion controls application
- Evaluate (qualitatively) select topologies based on these criteria

Topology



□ Considered

- Network configuration should
  - reduce weight (from central architecture)
  - exhibit reliability / redundancy
  - be maintainable
- 4 topologies seen as relevant to the application

http://en.wikipedia.org/wiki/File:NetworkTopologies.svg

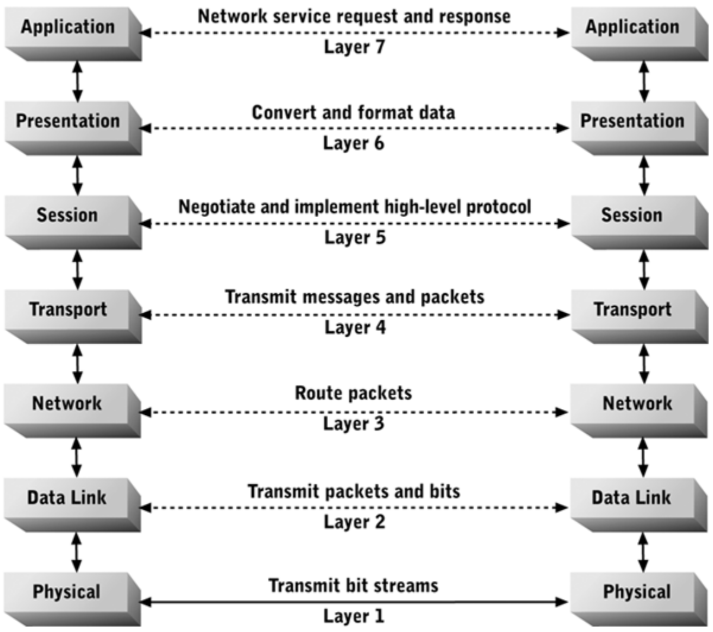
Topology
~ Results ~

	Ring	Star	Bus	Tree
Weight	🟩	🟨	🟩	🟨
Reliability	🟥	🟨	🟥	🟥
Maintainability	🟥	🟨	🟨	🟥
Msg TX Length	🟥	🟩	🟩	🟩

Summary of qualitative topology analysis -- green markers indicate potential benefits, red markers indicate potential challenges, and yellow markers are neutral

Communication Architecture

Network Layers



These layers must be specified and implemented for the DEC application

Failure Scenarios

1. Outgoing Link Failure
2. SOS Failure (Slightly-Off-Specification)
3. Spatial Proximity Failure
4. Masquerading Failure
5. Babbling Idiot Failure

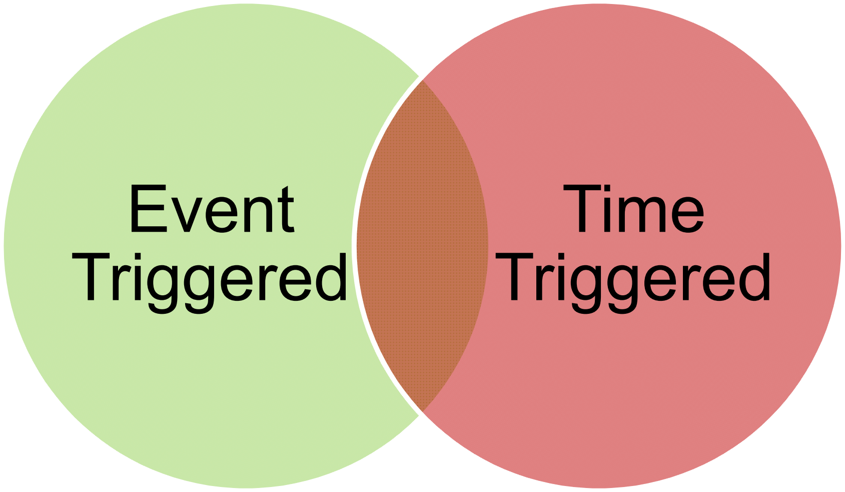
A distributed engine control system must be able to handle these types of failures and continue to function

Stephen D Burd. *Systems Architecture*. Course Technology Ptr, 2005

Approved for public release; distribution unlimited.

Protocols for Aerospace Applications

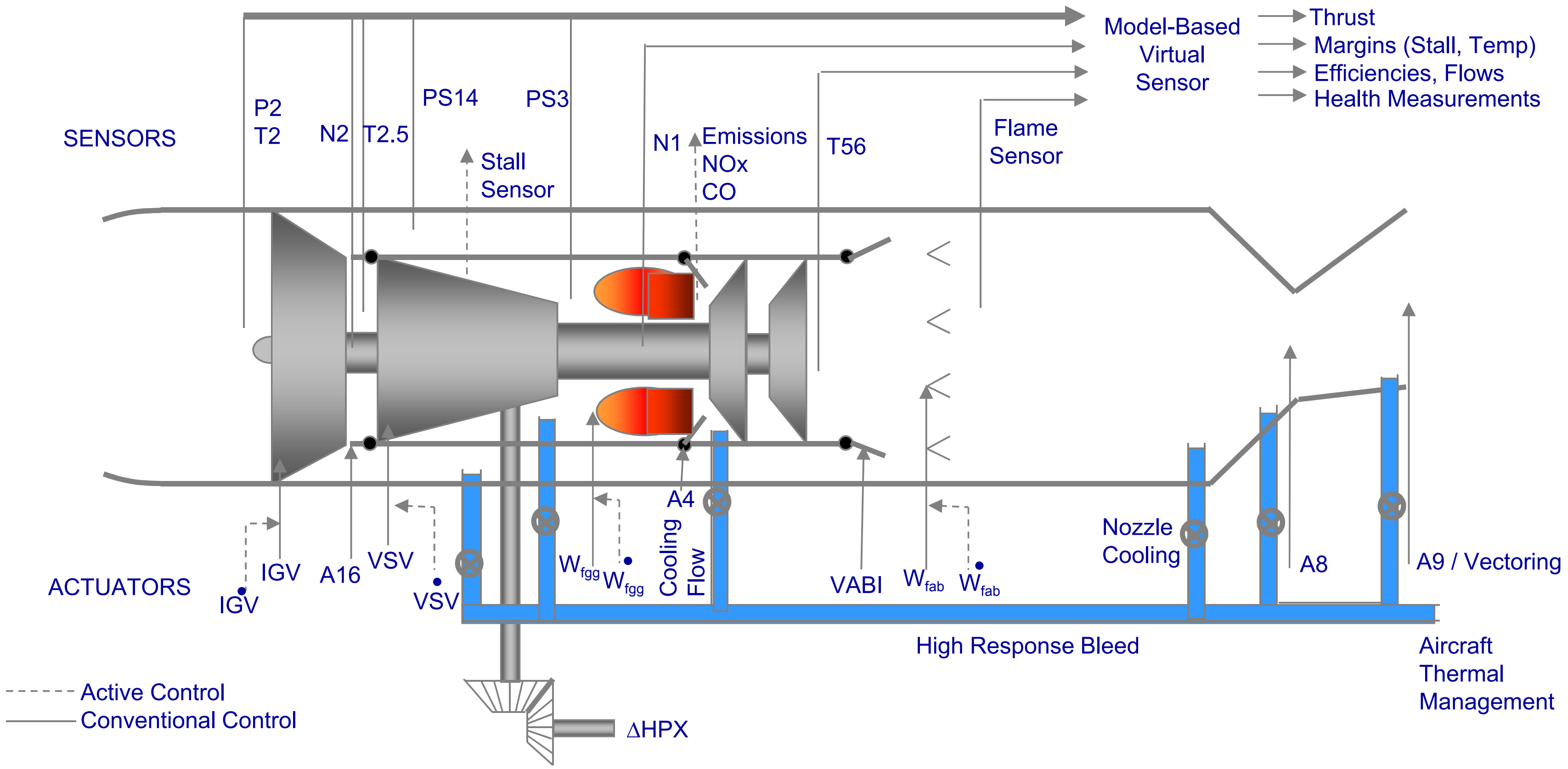
- SAFEbus
- TTP, TTP/C
- SPIDER
- TTCAN
- CANbus
- Ethernet
- AFDX
- FlexRay



Approved for public release; distribution unlimited.

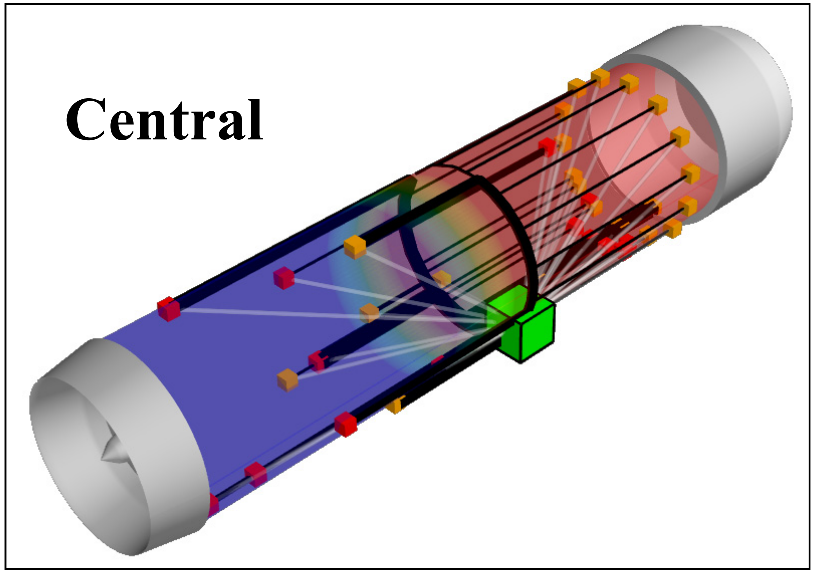
TYPICAL CONTROL SENSING SYSTEM

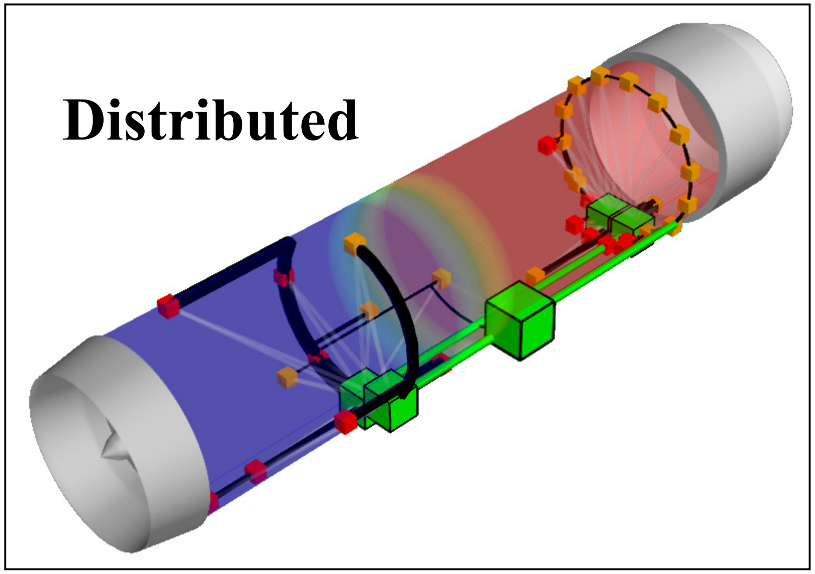
Including Electronic Control & Electric Actuation



This system can be extended for distributed control

Physical Network Model





🟩 FADEC

🟩 Data Concentrator

🟥 Sensor

🟧 Actuator

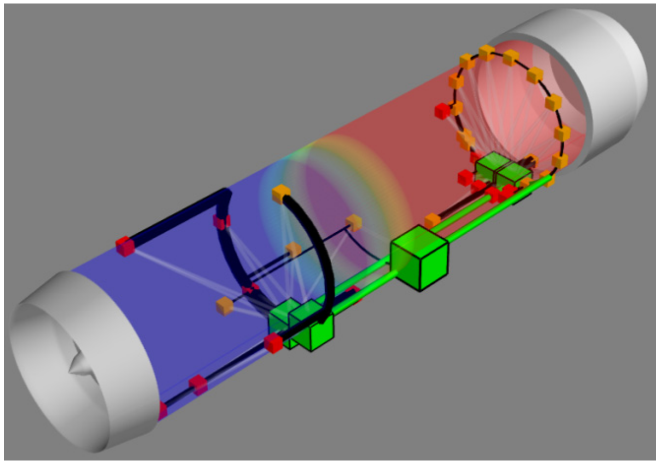
— Cable – few conductors

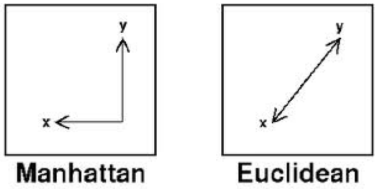
▬ Cable – many conductors

▬ High-Speed Databus

Assumptions

- Sensor & Actuator locations fixed
- Relative case temperature profile (blue & red areas)
- Sensors & Actuators must connect to 2 nearest data concentrators (or FADEC)
- Wiring is routed simply using *Manhattan Distance* approach





Particle Swarm Optimization ~ Formulation ~

Each ‘particle’ has a position and velocity in the solution space

$$x = [z_1, \theta_1, r_1, \dots, z_n, \theta_n, r_n]$$
$$v = [\underbrace{\dot{z}_1, \dot{\theta}_1, \dot{r}_1}_{\text{cylindrical coordinates for each DC}}, \dots, \dot{z}_n, \dot{\theta}_n, \dot{r}_n]$$

Update Rule (applied iteratively)

$$x_i = x_i + v_i$$
$$v_i = \omega v_i + \phi_p r_p (p_i - x_i) + \phi_g r_g (g - x_i)$$

- subscript $i \equiv$ reference to a particular particle
- $\omega \equiv$ inertial weighting
- $\phi_p, \phi_g \equiv$ cognitive and social weighting
- $p_i, g \equiv$ ‘personal’ and global best solutions found so far
- $r_p, r_g \equiv$ random numbers on the interval [0,1)

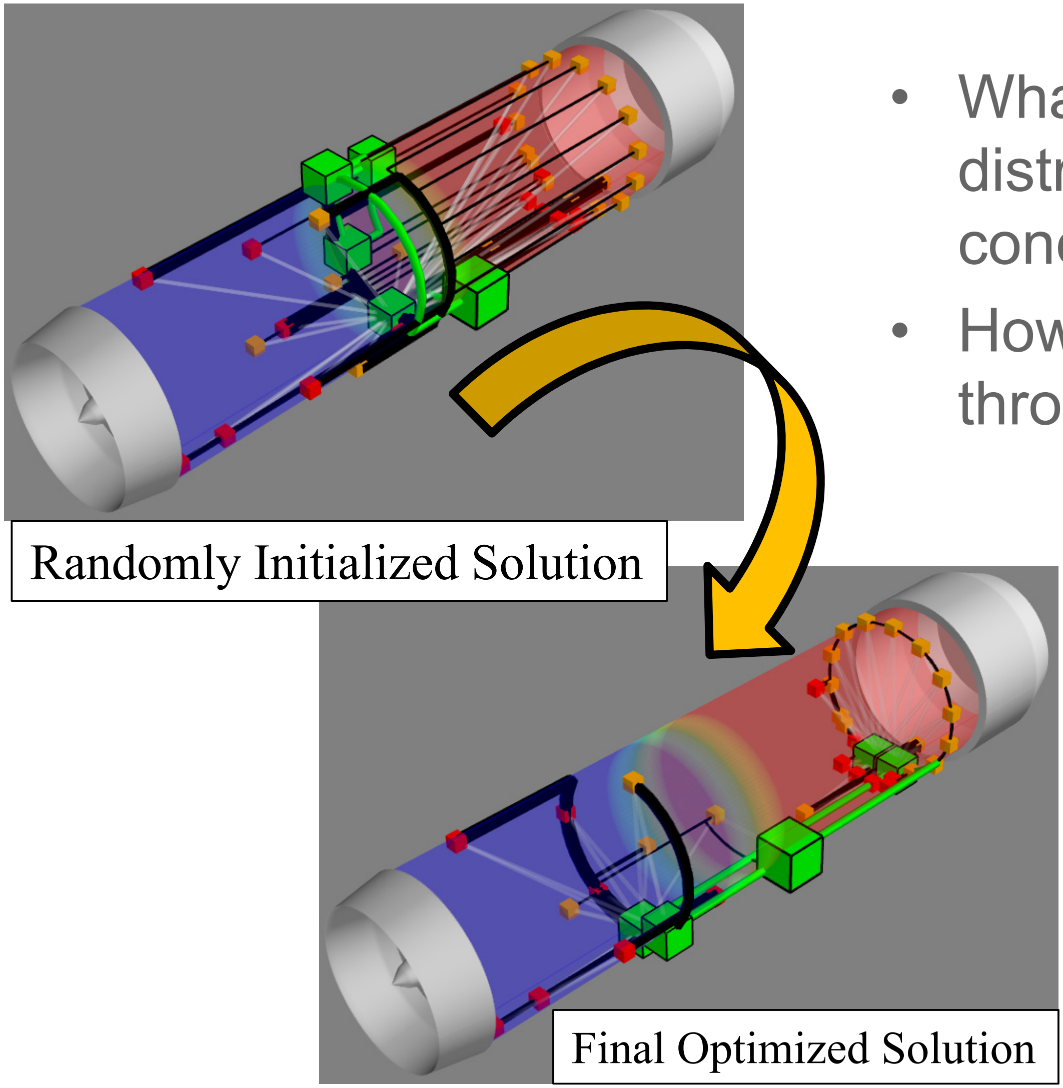
Particle Swarm Optimization ~ Fitness Function ~

$$f_i(x_i) = \sum_{k=1}^{c} r_k(\min_1(d_k) + \min_2(d_k)) + \sum_{j=1}^{n} w_j$$

- subscript $\boldsymbol{k}$ ≡ reference to a particular component (sensor or actuator)
- subscript $\boldsymbol{j}$ ≡ reference to a particular DC
- $\boldsymbol{d_k}$ ≡ a vector containing the distances from component k to each DC and the FADEC
- $\min_1 \boldsymbol{d_k}$, $\min_2 \boldsymbol{d_k}$ ≡ the minimum and next smallest, respectively
- $\boldsymbol{r_k}$ ≡ the number of conductors for component k
- $\boldsymbol{w}$ ≡ DC cost term incorporating temperature penalty and DC connections
- $\boldsymbol{c}, \boldsymbol{n}$ ≡ the total number of DCs and sensors/actuators, respectively

Approved for public release; distribution unlimited.

Finding the Optimum

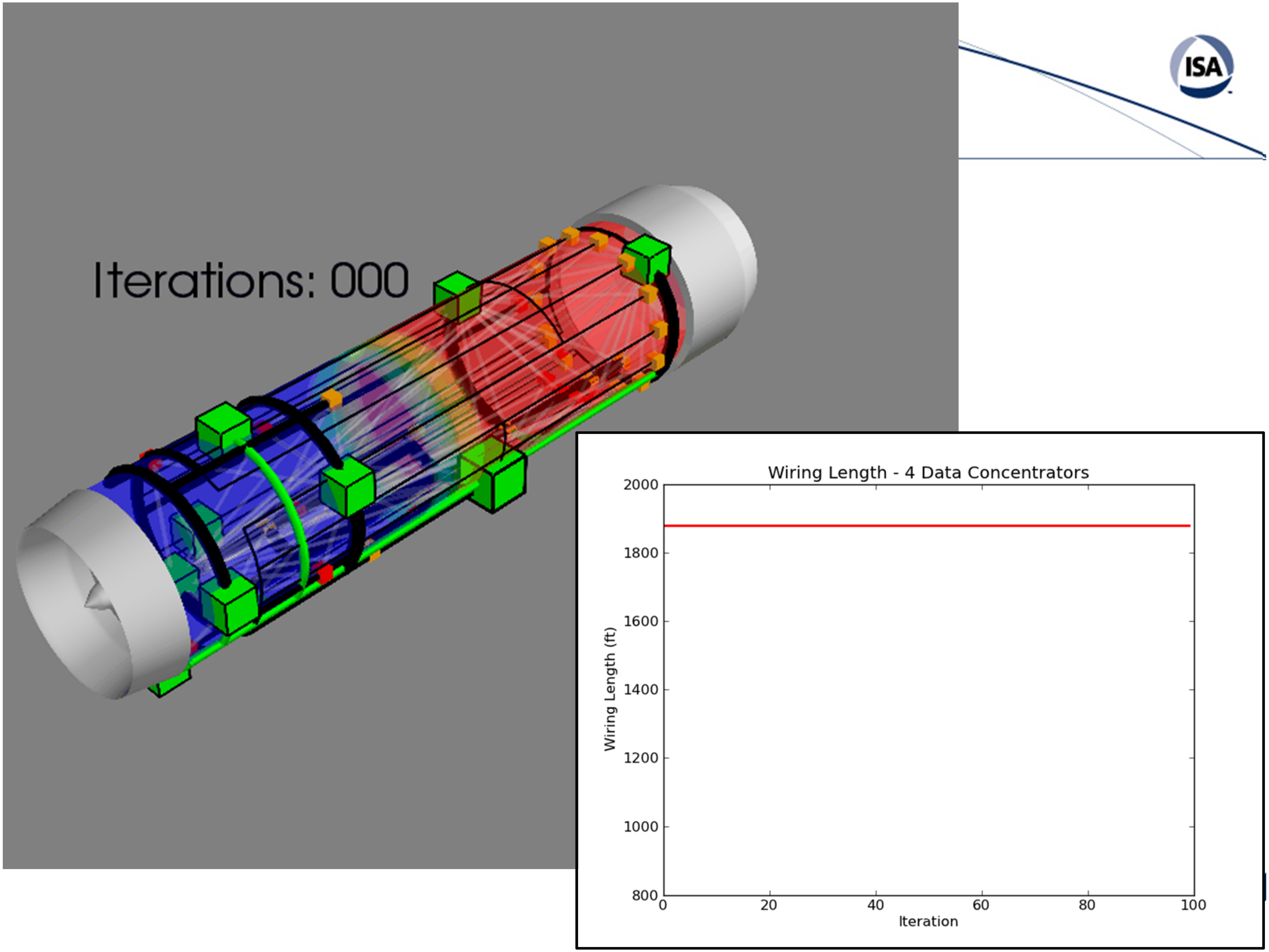


Randomly Initialized Solution

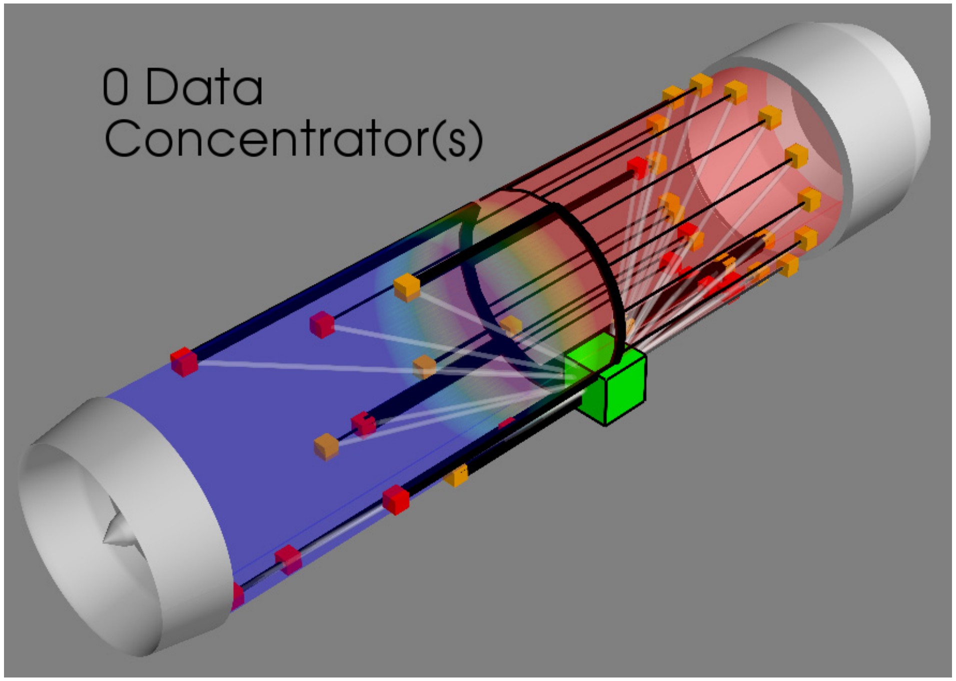
Final Optimized Solution

- What is the best spatial distribution of data concentrators?
- How does the solution evolve through each iteration?

Best fitness for wiring length, temperature penalties, etc.

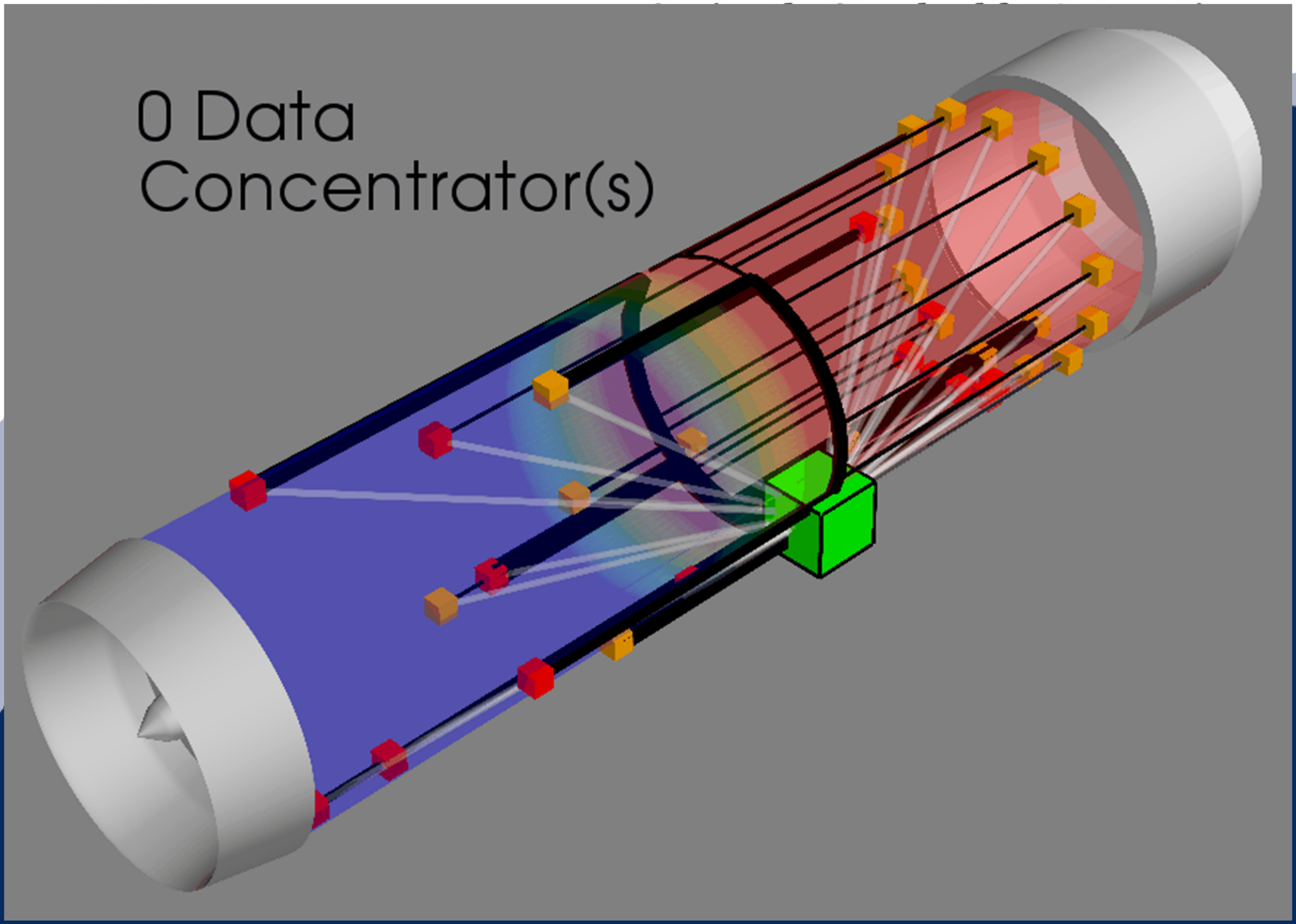

Iterations: 000
Wiring Length - 4 Data Concentrators
Wiring Length (ft)
2000
1800
1600
1400
1200
1000
800
0
20
40
60
80
100
Iteration

Adding Data Concentrators

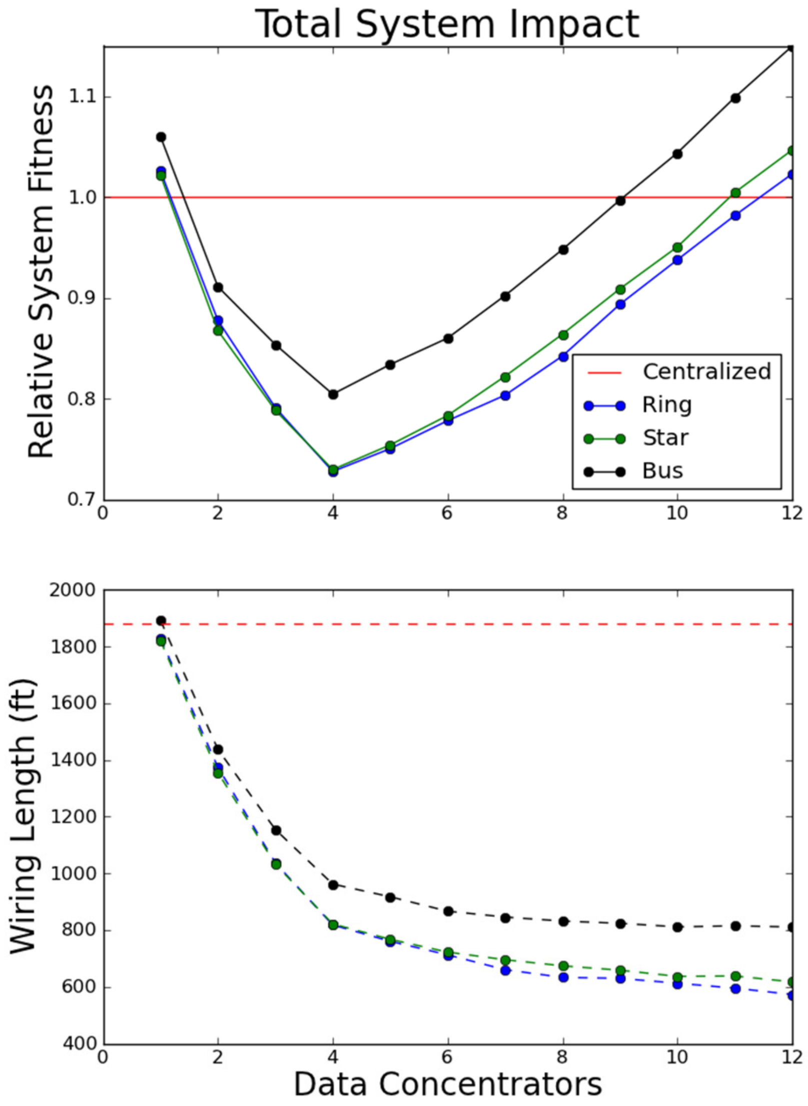


- How is the wiring affected by the number of Data Concentrators?

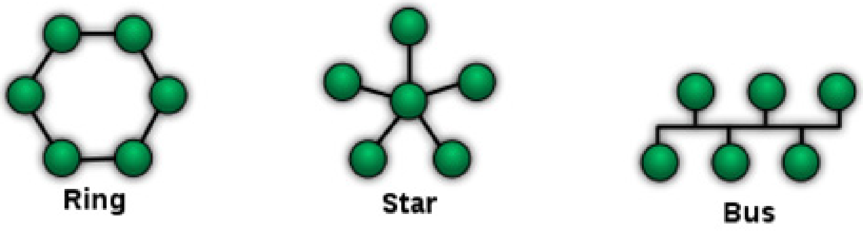
Approved for public release; distribution unlimited.


0 Data
Concentrator(s)

Results



Topologies Investigated



System Fitness

- Overall Wire Length
- Weight & Cost of Added Boxes
- Temperature Penalties

Payoff

Using 7 or more data concentrators, approximately **60%** of total wiring length can be saved

Conclusion

- This study was intended to provide data to aid in the selection of a network topology for future distributed turbine engines.
- A novel approach for analyzing and evaluating three topologies: ring, star, and bus in the context of a relevant military engine was described.
- Robustness and redundancy must be a consideration when designing these networks, a single failed node or communication link should not be able to take down the entire distributed control system
- There is potential for large wire length savings in a distributed control architecture.
- This is the tip of the iceberg: many extensions to this study are possible, including
  - Impact on TMS system and cooling requirements
  - Different constraints (i.e. spatial constraints, access panels)
  - Project results to cost and weight savings
  - Prioritize sensors and actuators and vary how many connections each require
  - Couple topology model with various network models to investigate performance

Approved for public release; distribution unlimited.

Questions?

Thank you!

Powering the Air Force

Developing practical engine control and Prognostics health management technologies

Approved for public release; distribution unlimited.