STABILITY AND PERFORMANCE OF PROPULSION CONTROL SYSTEMS WITH DISTRIBUTED CONTROL ARCHITECTURES AND FAILURES

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Rohit K. Belapurkar, M.S.

Graduate Program in Aeronautical and Astronautical Engineering

The Ohio State University

2012

Dissertation Committee:

Prof. Rama K. Yedavalli, Advisor

Prof. Meyer Benzakein

Prof. Krishnaswamy Srinivasan
Copyright by
Rohit K. Belapurkar
2012
ABSTRACT

Future aircraft engine control systems will be based on a distributed architecture, in which, the sensors and actuators will be connected to the controller through an engine area network. Distributed engine control architecture enables the use of advanced control techniques along with achieving weight reduction, improvement in performance and lower life cycle cost. The performance of distributed engine control system is predominantly dependent on the performance of communication network. Due to serial data transmission, time delays are introduced between the sensor/actuator nodes and distributed controller. Although these random transmission time delays are less than the control system sampling time, they may degrade the performance of the control system or in worst case, may even destabilize the control system. Network faults may result in data dropouts, which may cause the time delays to exceed the control system sampling time. A comparison study was conducted for selection of a suitable engine area network. A set of guidelines are presented in this dissertation based on the comparison study. Three different architectures for turbine engine control system based on a distributed framework are presented. A partially distributed engine control system for a turbo-shaft engine is designed based on ARINC 825, a general standardization of Controller Area Network (CAN). The effect of the addition of an engine area network on the stability and
performance of the proposed partially distributed engine control system is presented. Stability conditions are presented for the proposed control system with output feedback under the presence of network-induced time delay and random data loss due to transient sensor/actuator failure. A control design is proposed to stabilize the distributed turbine control system under these communication constraints. A fault tolerant control design is proposed to benefit from the additional system bandwidth and also from the broadcast feature of the data network. It is shown that a reconfigurable fault tolerant control design in which several control input values are transmitted and stored at the actuator can help to reduce the performance degradation in presence of node failures. It is also demonstrated that the best nominal performance of the distributed engine control system can be obtained by designing the control law based only on the network induced time delay. However, the performance of this optimum control design degrades in presence of sensor or actuator faults. A performance comparison is presented between zero-input, hold-input and reconfigurable data loss compensation strategy. It is shown that the optimum overall performance of the turbine engine control system based on a partially distributed architecture was obtained when the controller was designed based only on networked induced time delay with a reconfigurable data loss compensation strategy. A T-700 turbo-shaft engine model is used to simulate and validate the proposed control theory based on both single input and multiple-input multiple-output control design methodologies.
Dedicated to my wife, parents and grandparents for their love and support
ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my adviser, Dr. Rama K. Yedavalli, for his support and guidance during my Ph.D. research. He created an ideal environment which gave me freedom to explore new ideas and an opportunity to work on industry-relevant projects. I am grateful to Dr. Meyer Benzakein and Dr. Krishnaswamy Srinivasan for serving on my dissertation committee.

I would also like to acknowledge the support of Goodrich Engine Control Systems, Air Force Research Laboratory (AFRL) and National Aeronautics and Space Administration (NASA) through Small Business Innovation Research (SBIR) and Small Business Technology Transfer (STTR) programs. I would like to thank Mr. Paul Paluszewski for his help, advice and more importantly, for his trust and believing in me with the research responsibilities. I would also like to thank Dr. Alireza Behbahani for his help and encouragement during my graduate studies. I would also like to thank Mr. Stephen Reeps, Ms. Deb Trailor, Mr. Bill Storey, and Mr. Jonathan DeCastro for their support and technical guidance. I would like to express my gratitude to Prof. Rajkumar S. Pant for his encouragement to pursue a Ph.D. in dynamics and control systems.

I would also like to thank my colleagues, Dr. Praveen Shankar, Dr. Wenfei Li, Dr. Hsun-Hsuan Huang, Dr. Nagini Devarakonda, Mr. Deepak Saluru and Mr. Jonathan
Kratz for their discussions and for creating a positive research environment. I would like to offer my special thanks to my parents, Mr. Kishore and Mrs. Ashwini and my sister, Ruchita, for their love, support and tremendous encouragement over the years. Most of all, I deeply appreciate the love and support of my wife, Uttara, who has been a constant source of encouragement. Special thanks to my friends; Adwait, Anup, Kanika, Ninad, Parth, Pinak, Rahul, Renuka, Shubhanan, Suraj and Tejas for supporting me get through this process.
VITA

2006.................................................................B.E., Mechanical Engineering, University of Pune, India

2008.................................................................M.S., Aeronautical & Astronautical Engineering, The Ohio State University

PUBLICATIONS


FIELDS OF STUDY

Major Field: Aeronautical and Astronautical Engineering
TABLE OF CONTENTS

Abstract......................................................................................................................................... ii

Acknowledgments ....................................................................................................................... v

Vita........................................................................................................................................ vii

List of Tables ............................................................................................................................ xii

List of Figures ............................................................................................................................ xiii

1. Introduction............................................................................................................................... 1

   1.1. Current Gas Turbine Engine Control System ................................................................. 2

   1.2. Distributed Engine Control Systems............................................................................. 7

   1.3. Problem Statement ......................................................................................................... 10

   1.4. Dissertation Contribution ............................................................................................ 12

   1.5. Dissertation Organization ............................................................................................. 14

2. Turbine engine control systems ............................................................................................ 16

   2.1. Control System Architecture......................................................................................... 17

   2.2. Thermal Management .................................................................................................. 21
2.3. Cockpit Engine Indications ................................................................. 22

2.4. Control System Design ................................................................. 23

  2.4.1. Variable bleed valve control ......................................................... 24

  2.4.2. Variable stator vane control ......................................................... 24

  2.4.3. Fuel Control System ................................................................. 25

  2.4.4. Set-point Control Design ............................................................ 26

  2.4.5. Transient Control Design ............................................................ 27

  2.4.6. Limit Controllers ................................................................. 28

2.5. Advanced Control Methods ........................................................... 30

  2.5.1. Engine Degradation and Health Management ......................... 30

  2.5.2. Multivariable Control ................................................................. 30

  2.5.3. Active Control ........................................................................... 31

  2.5.4. Model-based Control ................................................................. 32

3. Distributed engine control systems .................................................... 35

  3.1. Introduction .................................................................................. 35

  3.2. Engine Area Network ................................................................... 39

    3.2.1. Flexray .................................................................................. 42

    3.2.2. AFDX ................................................................................... 43

    3.2.3. TTEthernet ........................................................................... 44

    3.2.4. ARINC Specification 825 ......................................................... 45

  3.3. Data Bus Requirements ................................................................ 48

  3.4. Communication Constraints ......................................................... 49
3.4.1. Constraint on Channel Bandwidth ................................................................. 50
3.4.2. Network-induced Time Delays ................................................................. 50
3.4.3. Network-induced Data Loss ..................................................................... 51

3.5. Sensor Requirements ..................................................................................... 52

3.6. Distributed Engine Control Architectures .................................................. 55
   3.6.1. Partially Distributed Architecture with Data Concentrator .................... 55
   3.6.2. Fully Distributed Architecture with Smart Nodes ................................. 58
   3.6.3. Fully Distributed Architecture with Intelligent Nodes and Decentralized
           Controller ....................................................................................................... 59

3.7. Partially Distributed Engine Controller for Turbo-shaft Engine ................. 61

3.8. Estimation of Bandwidth Utilization ............................................................. 70

3.9. Estimation of Network-induced Time Delay ............................................... 73

3.10. Network Faults ............................................................................................ 74

4. Distributed engine control design ..................................................................... 76

4.1. Gas Turbine Engine Control System ............................................................ 76

4.2. Networked Control Systems ........................................................................ 79

4.3. Communication Constraints ........................................................................ 82
   4.3.1. Bit-rate Constraints .................................................................................. 82
   4.3.2. Network-induced Time Delay ............................................................... 83

4.4. Control System Design .................................................................................. 88
   4.4.1. Output Feedback NCS Design with Constant Time Delay ................. 90
   4.4.2. Stability Under Random, Bounded Jitter ............................................. 96


4.5. Node Failures.............................................................................................................. 98

4.5.1. Single-Input Single-Output (SISO) Control System ........................................... 100

4.5.2. Single-Input Multiple-Output (SIMO) Control System ....................................... 105

4.5.3. Multiple-Input Multiple-Output (MIMO) Control System ................................. 107

4.6. Simulation:............................................................................................................... 113

4.6.1. SISO T700 Engine Simulation ........................................................................ 114

4.6.2. SIMO T700 Engine Simulation ....................................................................... 118

4.6.3. MIMO T700 Engine Simulation ...................................................................... 125

5. Conclusions.................................................................................................................. 132

6. Bibliography................................................................................................................ 135
LIST OF TABLES

Table 3.1: Minimum required bit-rate for engine control system sensors ....................... 54
Table 3.2: ARINC 825 data frame size and network transmission delay .......................... 71
Table 3.3: Bandwidth utilization for individually transmitted data packets ....................... 72
Table 3.4: Bandwidth utilization for proposed turbo-shaft engine control architecture ... 72
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Centralized turbine engine control system</td>
<td>7</td>
</tr>
<tr>
<td>1.2</td>
<td>Distributed turbine engine control system</td>
<td>8</td>
</tr>
<tr>
<td>2.1</td>
<td>Turbine engine station numbering</td>
<td>20</td>
</tr>
<tr>
<td>2.2</td>
<td>Schematic of turbine engine control system</td>
<td>26</td>
</tr>
<tr>
<td>2.3</td>
<td>Transient control design schematic</td>
<td>28</td>
</tr>
<tr>
<td>2.4</td>
<td>Limit control design schematic</td>
<td>29</td>
</tr>
<tr>
<td>2.5</td>
<td>Active clearance control system</td>
<td>32</td>
</tr>
<tr>
<td>2.6</td>
<td>Model-based control system with health management</td>
<td>34</td>
</tr>
<tr>
<td>3.1</td>
<td>Dual-redundant partially distributed FADEC with data concentrator</td>
<td>56</td>
</tr>
<tr>
<td>3.2</td>
<td>Triple-redundant partially distributed FADEC with data concentrator</td>
<td>58</td>
</tr>
<tr>
<td>3.3</td>
<td>Dual redundant fully distributed FADEC with smart nodes</td>
<td>59</td>
</tr>
<tr>
<td>3.4</td>
<td>Dual redundant fully distributed FADEC with active control sub-systems</td>
<td>60</td>
</tr>
<tr>
<td>3.5</td>
<td>T700 Hydromechanical Unit [61]</td>
<td>64</td>
</tr>
<tr>
<td>3.6</td>
<td>T-700 Electrical Control Unit [61]</td>
<td>65</td>
</tr>
<tr>
<td>3.7</td>
<td>T-700 engine control system [61]</td>
<td>66</td>
</tr>
<tr>
<td>3.8</td>
<td>Proposed partially distributed turbo-shaft engine control system</td>
<td>69</td>
</tr>
<tr>
<td>4.1</td>
<td>Network diagram showing all time delays</td>
<td>91</td>
</tr>
</tbody>
</table>
Figure 4.2: Simulink model for SISO T700 engine

Figure 4.3: Output response with network dynamics

Figure 4.4: Output response when $\eta_{max} = 10$

Figure 4.5: Output response with $K_{new}$

Figure 4.6: Output response without scheduling

Figure 4.7: Simulink model for SIMO T700 engine

Figure 4.8: Output response without any failures

Figure 4.9: Output response with single data packet transmission

Figure 4.10: $N_g$ sensor output response

Figure 4.11: $N_p$ sensor output response

Figure 4.12: $Q_{mr}$ sensor output response

Figure 4.13: $N_{mr}$ sensor output response

Figure 4.14: $w_f$ control input

Figure 4.15: Comparison between LQR and reconfigurable control gains

Figure 4.16: Simulink model for MIMO T700 engine

Figure 4.17: Output response with gain $K$

Figure 4.18: Output response with transmission faults

Figure 4.19: Output response with increase in sensor faults

Figure 4.20: Output response with increase in actuator faults

Figure 4.21: Output response with a reconfigurable control gain

Figure 4.22: Output response with node recovery time of 2 seconds
CHAPTER 1

INTRODUCTION

Gas turbine engines are one of the most efficient internal combustion engines offering very high power to weight ratio, high operational reliability and low emissions. The main uses of gas turbine engines are for propulsion and power generation with applications ranging from ground military vehicles, marine ships, military and commercial aircrafts, helicopters to industrial gas turbines for land-based power generation and auxiliary power generation for aircraft and ground vehicles. Gas turbine engines can be powered by commercial fuels like aviation fuel (jet fuel), natural gas, and propane as well as by renewable fuels like E85, biodiesel and plant-based biofuels. Micro gas turbine engines are one of the promising technologies for recharging batteries of hybrid electric vehicles. The next generation of power plants will be based on advanced gas turbines to allow for fuel-flexibility, low life-cycle cost, high reliability and low emissions. There were more than 30 million commercial flights in 2010 and it is projected that the airline passenger travel will double in next 20 years. Global air freight is also predicted to double in next 20 years. The increasing use of gas turbine engines highlights the importance of increasing the reliability and efficiency along with a reducing both the life cycle cost and
emissions. The use of an advanced, intelligent gas turbine engine control systems is identified as one of the key technologies necessary to achieve these objectives.

1.1. Current Gas Turbine Engine Control System

Gas turbine engines were developed independently by Hans von Ohain in Germany and by Sir Frank Whittle in England in late 1930s. The world’s first jet-powered flight was made on 27 August 1939 by a turbo-jet powered German Heinkel He 178 aircraft followed by the test flight of Gloster E.28/39, the first British jet engine aircraft, on 15 May 1941. The early jet engine control systems used a hydromechanical governor to meter the fuel flow rate based on a proportional controller, i.e., the fuel flow was proportional to the difference between the desired and measured turbine shaft speed. A physical stop on the throttle position ensured limits on the engine speed and a minimum flow stop on the fuel metering valve prevented the engine from flame-out. Over-temperature protection was provided by the use of a maximum flow schedule. The control system design was based on frequency response methods and the performance analysis was focused mainly on steady-state response. Time domain control design methods were gradually implemented along with variable geometry like inlet guide vanes (IGV) by early 1950s. This was followed by the use of transient performance analysis techniques in mid-1960s via analog computer simulations. Analog and digital electronic control units (ECU) were developed in early 1970s to provide high-level supervisory or trim functions. The early full-authority ECUs, which controlled the entire operation of an engine based on the pilot’s thrust setting were analog and helped to achieve an
unrestricted power lever movement, thereby reducing the pilot workload, eliminating the trim which is required after component replacement and improving the component fault diagnostic capabilities. Although digital electronic engine control systems and advance control design techniques like multi-variable control, optimal control design methodologies like linear quadratic regulator (LQR), integrated flight and propulsion control (IFPC), etc. were flight tested, they were not used for production engines.

Advancement in the capabilities and reliabilities of digital electronics enabled the use of a full authority digital engine control (FADEC) along with electric actuators, use of multi-variable control design and improved engine health diagnostics and fault detection techniques. The working of turbine engines and the turbine engine control system are well discussed in [1]. The historical progress of turbine engine control systems, engine modeling, design and analysis of turbine engine control systems and the future directions for engine control system design are presented in [2]. The Pratt & Whitney F135 turbofan engine developed for the F-35 Lightning II is the most advanced jet engine to be flight tested in 2000 and entered into production in late 2010. According to the 2011 Flight International annual airline census, there are approximately 25,000 commercial aircrafts in-service worldwide with a backlog of nearly 8,000 airframes. Most of these aircrafts have 2 or more engines per airframe. More than 2000 engines were delivered to power Airbus and Boeing commercial aircrafts in 2011 with total backlog of more than 16,000 engines. The global military fleet is nearly 52,000 aircrafts which include combat aircrafts, special mission, transport/tanker aircrafts, combat helicopters and training helicopters/aircrafts. Although the United States Department of Defense (DOD) fuel
consumption decreased by 4% between FY2005 and FY2011, the spending on fuel rose by 381% in inflation-adjusted terms from $4.5 billion to about $17.3 billion. According to US Air Transport Association (ATA), fuel cost account to more than 40% of the operating cost of US airlines and an increase of the oil barrel cost by 1 dollar adds $456m a year in jet-fuel costs for US airlines. Due to the high fuel prices and large fuel requirements, a small improvement in fuel efficiency can result in significant saving to each airline. Hence, there is a huge monetary incentive to improve engine fuel efficiency without reducing the engine output (thrust for turbo-fan and torque output for turbo-shaft engines), reliability and safety and also to decrease the engine emissions. Significant savings in engine operating cost can be achieved by reducing the engine life-cycle and maintenance cost along with the engine fuel consumption. The Versatile Affordable Advanced Turbine Engines (VAATE) program aspires to achieve 200% increase in engine thrust-to-weight ratio, 25% reduction in engine fuel consumption, 60% reduction in engine development, procurement and life cycle maintenance cost with the help of collaborative effort between National Aeronautics and Space Administration (NASA), Department of Defense (DOD), Department of Energy (DOE), academia and industry. Improvements in the technical areas such as advanced materials, mechanical systems including compressors, turbines, engine control and health management systems are needed to achieve the aggressive goals of the VAATE program. Use of active control technologies like active stall control, active combustion control, and active tip clearance control can significantly improve the performance and efficiency of turbine engines. Life cycle maintenance cost can be reduced by the use of engine health management systems,
which will improve the fault detection and diagnostic capabilities of turbine engine control system.

Due to the difference in manufacturing tolerances and material properties, the performance of any two identical new engines will differ under same operating conditions. Also, the engine will deteriorate with time and will result in a degraded performance. Traditionally, time based scheduling methods are used to repair or overhaul turbine engines, which results in a mismatch in component health conditions, further increasing the engine performance variation. Since a new engine will have a large stall margin as compared to a deteriorated engine which operates at low or zero stall margin, the engine acceleration times are determined from the stall margins of a deteriorated engine. The current FADEC has auto-trim capabilities, which allow controller calibration to match the actual engine health condition and the pilot gets the desired performance irrespective of the engine health condition. Use of adaptive control design methods or intelligent, active control techniques will allow the controller to operate the engine with the optimum efficiencies and extend the engine component life by operating them at more benign conditions. Some of the promising active control technologies as discussed in [2], are as below:

1. Active clearance control
2. Active combustion control
3. Active compressor stability control
4. Model-based control design
The tip clearance in compressors and turbines depends on the operating condition, external loads and component deterioration. A large tip clearance results in significant losses, while, a small tip clearance can result in shroud rubbing. Active clearance control systems use tip clearance sensors to reduce tip clearances by using fast acting actuators, which essentially act as a local control sub-system with a very high sampling rate. Active compressor stability control system will result in an improvement in both engine pressure ratio and surge margin. Active combustion control will help to reduce emissions, improve the turbine inlet temperature distribution, avoid blowouts and control combustion instabilities. The active control techniques can be viewed as additional sub-systems with local loop closures and the FADEC will perform the supervisory duties to ensure the desired engine performance and will access the overall health of the engine. The current generation of production turbine engines have a digital controller (FADEC) connected to analog sensors and analog actuators via individual wire harnesses. As shown in Figure 1.1, each sensor or actuator is connected to the FADEC by an individual wire harness and the sensor data acquisition, data processing and control law calculation is executed on a centralized level by the FADEC. Most of the active control systems require high bandwidth sensors and actuators along with high computational capabilities. Due to the analog nature of sensors and actuators, the current engine control architecture has limitations on the type of sensors and actuators which can be connected to the FADEC. Hence, the use of advanced control and health management techniques require a transition from the current legacy analog and digital control system architecture to digital and embedded control system architecture.
1.2. Distributed Engine Control Systems

Turbine engine control system based on a distributed architecture were proposed in 1990s in [3], [4] and [5] and more recently in [6], [7] and [8]. The benefits of designing a turbo-shaft engine control system based on a partially distributed architecture were discussed in [9]. The distributed engine control system architecture will act as a foundation for implementing the advanced, adaptive active control systems. In this architecture, the legacy analog sensors and analog actuators will be replaced by digital smart sensors and smart actuators. As shown in Figure 1.2, the point-to-point analog wire harness based connections between the sensors/actuators and FADEC will be replaced by a digital, serial data bus. The data acquisition, analog-to-digital converter (A/D), digital-to-analog converter (D/A) and signal conditioning functionality of the legacy FADEC will be
distributed to the smart sensor/actuator nodes. The smart nodes will also possess the electronics required to encode/decode the data based on the communication protocol and local fault detection capability. If the smart sensor has additional functions such as performing local loop closures or fault compensation mechanisms, it is identified as an intelligent node.

Figure 1.2: Distributed turbine engine control system

Distributed engine control system architecture will also allow the transition to all electric engine control system, in which, the hydro-mechanical actuation systems will be replaced by electric actuators, leading to huge savings in weight and improvement in system reliability. The use of fiber optic based data network instead of the heavy, shielded wire
harnesses will contribute to additional weight savings. Furthermore, use of multi-mode fiber optic cable will help to achieve high data transmission rate along with an improvement in reliability.

Recent advances in fiber optic sensors have led to the development of sensors for high temperature measurements. Such sensors have several advantages including low weight, resistance to electromagnetic interference, small packaging size and non-electrical conductivity. The integration of such high temperature fiber-optic sensors with fiber optic data transmission will allow the placement of sensors close to the desired sensing location in combustors. Distributed engine control architecture will also allow the use of mission-specific sensors, for example, sensors to detect ash ingestion for flight operations near high volcanic activity regions or sand ingestion detection sensors for deployment of military aircrafts in a desert environment. Since the average lifespan of an engine is more than 40 years, scheduled maintenance, repair, and overhaul (MRO) activities have to be carried out to maintain the engine performance. This calls for maintaining an inventory for all the turbine engine components throughout the engine life. Rapid advancements in electronics have led to short product life cycles of components like sensors, micro-controllers, etc. Such obsolescence issues increase the inventory management cost and also necessitates the development of new hardware components for retrofitting the engine control system. Since the current control system architecture is based on a centralized architecture, both the software and hardware of the new control system has to be re-certified for flight use. This not only increases the development cost but also the
development and certification time. Since the new control system will be based on a
distributed architecture, it will be more modular than the current architecture and will
facilitate the retrofit process. It will be also easier to integrate the flight and propulsion
control system design thereby improving the flight handling capabilities under loss of
flight control surfaces. Variable turbo-shaft rotor speed control system, which will
improve the helicopter maneuverability and reduce the rotor noise will also benefit from
the distributed nature of the engine control systems. Distributed engine control
architecture will also allow better communication between the flight control system and
engine control systems of all the engines installed on the aircraft. Some additional
benefits of the distributed engine control system include real-time processing of sensor
data for fault-diagnostics, improved post-flight engine health management or even in-
flight data transmission for engine fleet management. Advances in data transmission
security, efficient data transmission methods and energy-harvesting technology will allow
use of wireless sensors, initially for health monitoring followed by a full distributed
wireless engine control system.

1.3. **Problem Statement**

Although there are many benefits of transitioning from a centralized to a distributed
architecture for turbine engine control systems, there are several technological challenges
which have to be met first. The main stumbling block in implementation of distributed
engine control system is availability of high temperature electronics. Since distributed
engine control system requires the use of smart sensors which will be often located in high temperature regions like combustor or turbines, smart sensor electronics should be able to withstand high temperatures and extreme environmental conditions. Electronic components based on silicon on insulator (SOI) technology can operate at 225-250 °C. Silicon carbide (SiC) based electronics, which are anticipated to operate at temperatures greater than 500 °C are being developed by the industry in collaboration with federal agencies. Electronic packaging techniques to provide the necessary thermal management, weight reduction and protection from harsh environmental conditions should be developed for design of smart nodes. Along with modular hardware, control design software developed using a modular approach will facilitate the validation and certification process.

The selection of the appropriate communication protocol and physical data bus is one of the most important initial design phase decision. The communication protocol should provide deterministic communication, high reliability and sufficient data bandwidth. The physical data bus should be able to tolerate high temperatures, extreme vibrations, lighting strikes and should be immune to electromagnetic interference (emi). The physical data bus interface connectors should be standardized designed for mistake proofing and to offer ease of access, serviceability and flexibility in routing. The data bus and communication protocol should also provide both backward and forward compatibility with sensors, actuators and other control system components. The addition of data bus will impart additional design constraints such as limited data bandwidth, data scheduling and redundancy management, and will impose communication constraints.
such as transmission time delays and data loss due to message dropouts. These constraints can have a significant impact on stability and performance of control systems and should be included in the control design process. The turbine engine control system should also benefit from the features of the distributed communication architecture, such as improved fault detection and compensation, improved health management strategies, etc. Since the estimator or tracking filter will form the basis for implementing model-based control methodologies and engine health management strategies, the impact of these communication constraints on the accuracy of tracking filters should be studied in detail. In addition, new verification and validations tools need to be developed for turbine engine control systems based on a distributed architecture.

1.4. Dissertation Contribution

The performance of the turbine engine is predominantly based on the performance of the control system. The turbine engine model along with the control law design methodology is considered proprietary by the engine manufactures. Hence, this dissertation is based on turbine engine models and turbine engine control system design methodology available in public literature. Since the distributed architecture will serve as a foundation for implementation of next generation turbine engine control systems, a common set of guidelines are required to design and implement distributed engine control system. This dissertation aims to present a study on the impact of addition of the data network and presents guidelines for designing the basic distributed architecture and for scheduling the
data transmission. Most of the current time delay control design methodologies are based on sufficient conditions and do not guarantee the existence of the controller. Also, these stability conditions and control design techniques require heavy computational capabilities and are not suitable for gain scheduled control design with min-max architectures. Most of the existing networked control design methodologies also assume single data packet transmission policy. Although the control design methodology for networked control systems with multiple packet data transmission is discussed in [10], [11] and [12], they were applicable for only full state feedback control systems. In this dissertation, a control design methodology is proposed for output feedback control systems with constant time delay. This dissertation presents both necessary and sufficient conditions for analyzing the stability under transmission delays, sampling jitter and data loss due to packet dropouts. A control design methodology is presented to stabilize the time delayed control system. Stability conditions are presented for random, bounded jitter introduced either due to loss in clock synchronization or due to the communication protocol. It is shown that random, transient network faults or communication failures can degrade the system performance or even destabilize the control system. A reconfigurable active fault tolerant control design is proposed to make the system robust against network failures. The proposed control system design takes advantage of the additional unused communication bandwidth to transmit control input values based on reconfigurable controllers. The broadcast nature of the data protocol ensures that the node failure is communicated to all active nodes, thereby allowing them to compensate for the failed node even if the engine area network is inactive. This active fault tolerant control design
helps to reduce performance degradation in presence of transient node failures and random data packet dropouts. A non-proprietary T-700 turbo-shaft engine model is used to validate the proposed methodology for both single input and multiple input control system. Results of comparison study between zero-input, hold-input and reconfigurable input strategy are presented and it is demonstrated that the best performance of turbine engine control systems based on partially distributed architecture is achieved when the controller is designed based only on network-induced time delay and a reconfigurable control input strategy is implemented to compensate for the presence of node failures.

1.5. Dissertation Organization

The current turbine engine control systems are discussed in detail in chapter 2. Chapter 2 also contains description of the current architecture based on analog wire harness connections between FADEC and sensors/actuators. Several turbine engine sub-systems which interact with FADEC along with the FADEC functions are discussed in Chapter 2. Along with the control system architecture, the control system design and advanced control methodologies are also discussed in brief in section 2. Section 3 presents a brief overview of turbine engine control systems based on a distributed architecture. The benefits and technical challenges for implementation of the distributed engine control systems are also discussed in this chapter. A comparison between several communication protocols is presented along with data bus selection guidelines. The various communication constraints like network induced time delay; limited bit-rate and network
induced data loss are briefly discussed in chapter 3. Chapter 3 also contains a brief comparison between 3 different architectures for distributed engine control systems. A partially distributed engine control system is proposed for turbo-shaft engine and the effect of the above mentioned communication constraints on the proposed architecture are briefly discussed. Chapter 4 contains the stability and performance analysis along with a robust control design methodology. The various failure scenarios for SISO, SIMO and MIMO control systems are also discussed in chapter 4 along with a discussion on failure compensation strategies. A Simulink model is developed for the T-700 turbo-shaft engine model and the proposed controller is validated under the presence of all the communications constraints and random, transient network failures. Chapter 5 presents the conclusion of this study along with the recommendations for future research.
CHAPTER 2

TURBINE ENGINE CONTROL SYSTEMS

Gas turbine engine is a type of an internal combustion engines based on thermodynamic working cycle. The main components of a gas turbine engine are compressor, combustor and turbine. A typical gas turbine engine will have a low pressure compressor, high pressure compressor, high pressure turbine and low pressure turbine, each having one or more stages. A turbofan also has a multiple-stage fan in addition to the above components. Some additional components to ensure safe operation of gas turbine engine include fuel distribution system, engine control system, lubrication system, air cooling system, ignition system, accessory gearbox module, etc.[13]. Depending on the application, the turbine engine system can be also fitted with a thrust reverser system, an afterburner, anti-icing system, starting system, bleed system and transmission gearbox. In the next section, a brief review of the gas turbine engine control system architecture is presented.
2.1. **Control System Architecture**

Gas turbine engine control system ensures that the engine provides the pilot requested performance while maintaining its operational limits. In addition, the control system must also provide a safe operation with high reliability, be economically viable and should meet the emission, noise and airworthiness standards.

A gas turbine engine control system consists of one or more control modules and is connected to several sensors to measure the environmental parameters like temperature and pressure, and actuators to control parameters like fuel flow rate, compressor air bleed, etc. A full authority digital engine control (FADEC) unit is used to control the engine using the data from the sensors. Turbine engine control systems also include additional sensors such as oil temperature and level sensor, vibration monitoring, etc. required for health monitoring of engine components. The basic FADEC functions are

1. Provide the thrust requested by the pilot through the power lever. Since thrust is not directly measurable, the pressure ratio or fan speed is used to indicate the thrust. For a turbo-shaft engine, the control objective is to maintain a constant power turbine speed.

2. Maintain the engine components within its operational limit. The limit controllers avoid compressor surge and stall, prevent fan blade breakage and ensure that the turbine temperatures do not exceed the material temperature limits of the turbine blades.

3. Engine component health monitoring and performing fault detection, isolation and repair.
4. Ignition control to ensure safe engine start-up and shut down.

5. Control the thrust reverser.

6. Variety of other functions like controlling the deicing system, interfacing with avionics to provide engine health data and integration with the flight control systems.

Since turbine engine is a safety-critical system, the engine control system design should guarantee high reliability of operation and should contain fault-tolerant and fail-safe mechanisms. High operational reliability is usually achieved by the use of redundant sensors and controllers. Dual-redundant FADEC channels are used to improve the engine reliability. Each FADEC channel is connected to an individual sensor suite. The control-critical sensors are redundant per FADEC channel. Although there is a single actuating mechanism per engine, it is designed to be operable by either of the FADEC channels.

Both the FADEC channels are housed in one unit and are physically separated by a modular design with each channel possessing its independent power supply, sensor inputs and control outputs. Both the FADEC channels are connected by a data link for sensor data sharing. Health monitoring sensors are not redundant and are connected to either of the FADEC channels.

Some of the measured engine parameters which are measured are as follows

1. Pressure

   Both total and static pressures can be measured depending on the sensor requirement. Compressor static discharge pressure $P_{s3}$ is a common sensor
location which is used to limit the compressor pressure. Other sensed values include $P_1, P_2$ pressure sensors. Pressure sensor values are used for engine condition monitoring, calculating engine pressure ratio (EPR) and by FADEC for fuel flow rate calculation.

2. Temperature

Only total temperatures can be measured by the sensors. Inter-turbine temperature $T_{35}$ is measured to limit the turbine inlet temperatures and also as cockpit indicator. The exhaust gas temperature (EGT) can be estimated based on the inter-stage turbine temperature (ITT). Thermocouple type sensors or resistance temperature detector (RTD) type sensors are used to measure the temperature, depending on the temperature range.

3. Rotational speed

For a two spool turbine engine, $N_f$ or $N_l$ is the low pressure compressor or fan speed. The high pressure compressor speed is denoted by $N_c$ or $N_2$. For turboshift engines, the gas generator speed is denoted by $N_g$ and power turbine speed by $N_p$. These speeds are used as cockpit indicators as well as by FADEC for engine control and health monitoring.

4. Engine vibrations

Vibration sensors are attached to engine casing and used for monitoring purpose only. Vibrations sensors may be also installed on load bearings of an engine gearbox to assess its health.

5. Engine torque
Power turbine torque sensors provide the true indication of engine output power for turbo-shaft engines. These sensors are not used by the turbofan engine control systems.

FADEC is also connected to the several other sensors like thrust lever (power lever angle, PLA) sensor, engine start switches, $N_1$ mode selector, etc. FADEC is powered by its own independent power supply, which is a small alternator housed on the accessory gearbox. In case of FADEC alternator failure or during engine starting, a backup power supply is used. This backup power is supplied by the electrical system of the aircraft. A typical turbine engine station numbering is shown in Figure 2.1.

![Figure 2.1: Turbine engine station numbering](image_url)
2.2. Thermal Management

Turbine engine thermal management system performs two main functions; cooling of engine lubrication oil and FADEC unit cooling. Engine lubrication system provides lubrication for rotor bearings, gearbox bearings, gears and also supplies oil for oil damped bearings. Due to the heat transferred from the engine components, the lubrication oil gets hot and its internal temperature must be maintained below the oil flash point. Since the fuel stored in the fuel tanks is at cold temperatures, it must be heated before it is distributed to the fuel nozzles. The heating of fuel and cooling of lubrication oil is achieved through a fuel-oil heat exchanger. The heat exchanger heats the fuel before it enters the fuel metering device and the cooled lubrication oil is fed back to the oil storage tank. The engine fuel is also used for FADEC thermal management. The electronics required for signal data acquisition and condition, control law processing and engine health management, which are all enclosed in a single FADEC unit, generates large amount of heat. Since the FADEC based on the centralized architecture is connected by individual wire harnesses to all sensors and actuators, it is mounted on the engine case or fan case to minimize the wire harness length. This location exposes the FADEC to a very hostile environment of temperatures exceeding 200 °C and extreme vibrations. Hence, cooling of FADEC electronics is required to keep the internal components below the electronic operating temperature limits of 125 °C. FADEC unit cooling is achieved by using the jet fuel as a coolant. Ground idle operations on a hot day often results in large amount of heat dissipation, which increases the fuel temperature above the fuel temperature limit. In order to maintain the fuel temperature below its flash point, engine
shut-down is initiated. Also, the limited heat carrying capacity of the fuel imposes strict 
FADEC cooling restrictions during long flights in extreme hot weather and severely 
limits the total flight time.

2.3. Cockpit Engine Indications

According to the Federal Aviation Administration (FAA) Federal Aviation Regulations 
(FAR) section FAR 25.1305, following cockpit engine indicators are required to be 
installed for each engine.

1. Oil pressure warning
2. Oil temperature indicator
3. Oil filter clogging or oil debris monitoring indicator
4. Oil quantity indicator for each oil tank
5. Exhaust gas temperature (EGT) indicator
6. Rotor Speeds $N_1$ and $N_2$
7. Engine pressure ratio (EPR)
8. Fuel pressure warning
9. Fuel flow-meter indicator
10. Ice protection system functioning indicator
11. Ice clogging heater system of fuel systems indicator
12. Engine torque indicator
13. FADEC status
14. Thrust reverser system status
15. Starting and ignition system status

Since these parameters are either sensed or calculated by the FADEC, data transmission between each engine FADEC unit and cockpit avionics is required. Safety, reliability, efficiency and maintenance functions play an important role in design of FADEC. In next section, a brief review of turbine engine control system design is presented.

2.4. Control System Design

The main function of the Full Authority Digital Engine Control (FADEC) is to provide the thrust level demanded by the pilot while maintaining high engine efficiency and preventing the engine parameters like shaft speeds, temperature and pressure from exceeding their limits. These limits are imposed either to prevent compressor stall/ surge, for example, the compressor static discharge pressure has both lower and upper limits, to reduce the frequency of turbine overhaul by limiting the turbine inlet temperatures or to ensure safety by limiting the fan and core shaft speeds. The compressors of a gas turbine engine are designed to operate with maximum efficiency at the design point, which is the cruise condition. For operation on a power setting other than the cruise condition, the compressor may experience stall or surge leading to engine shutdown or loss in aircraft maneuverability. To prevent the compressor stall or surge, variable bleed valves (VBV) and variable stator vanes (VSV) are operated by the FADEC based on compressor temperature and pressure inputs. A non-linear, physics based turbine engine control simulation tool called as Commercial Modular Aero-Propulsion System Simulation 40K (C-MAPSS40k) is discussed in detail in [14] and [15]
2.4.1. Variable bleed valve control

As discussed earlier, the variable bleed valves (VBV) are used by the engine control system to recover from a compressor stall or surge condition. VBV are located at the exit location of low pressure compressors (LPC) and when opened, they bleed a portion of airflow out of the LPC, decreasing the airflow angle of attack at the rotor blades. The VBV position is typically scheduled based on low pressure compressor speed, $N_1$, altitude, Mach number, power lever angle (PLA) and $T_2$. Bleed valves are also located on the high pressure compressor (HPC) to prevent compressor stall at low speed operation, acceleration or during engine start. They are also used to recover from compressor surge. They are either called as HP bleed valves or start bleed valves, depending on their main purpose. Unlike the LPC bleed valves, they have only two positions, either open or close.

2.4.2. Variable stator vane control

The HPC inlet and stator vanes of first few HPC stages are designed as variable stator vanes. They are operated by the hydraulic actuators between maximum low speed positions to maximum high speed position. This ensures that an optimum angle of attack at the rotor blades is maintained at all operating conditions. The variable stator vane (VSV) position angle is scheduled based on high pressure compressor speed ($N_2$) and inter-compressor temperature $T_{25}$. A sufficient surge margin is maintained by using acceleration schedules and hence altitude, PLA and $N_1$ are also used for scheduling the VSV angle.
2.4.3. Fuel Control System

As discussed in the previous section, the main objective of the turbofan control system is to provide the thrust requested by the pilot through the thrust lever. For a constant thrust lever angle position (PLA), the engine thrust is dependent on the fan speed, pressure ratio and ambient operating conditions including ambient temperature, air density, etc. Since thrust is not directly measureable, alternate engine parameters must be used to estimate the engine thrust. Since a majority of the turbo-fan engine thrust is produced by the fan, fan speed correlates well with the thrust output and can be used as a control variable. The Engine Pressure Ratio (EPR), which is the ratio between low pressure turbine discharge pressure ($P_{50}$) and inlet pressure ($P_2$), can be also used by the FADEC as a control variable. General Electric (GE) and CFM both design the FADEC for fan speed control while Rolls-Royce and Pratt & Whitney design the controller based on engine pressure ratio control. Since the helicopter lift and maneuverability is controlled by changing the main rotor blade angle, the turbo-shaft engine is always maintained at a constant designed rotor speed. Hence, the turbo-shaft control design objective is to maintain a constant power turbine speed in presence of external disturbances like wind gusts. The turbine engine control design can be separated into three control design tasks; set-point control design, transient control design and limit control design. Figure 2.2 presents a schematic of the integration of all three different controllers. The parameter limits used for the limit controller are engine specific and determined based on worst-case operation of a deteriorated engine.
2.4.4. Set-point Control Design

Design of a controller for a steady-state operating condition is known as a set-point control design. The common operating conditions used for set-point controller design are idle, cruise and maximum power and takeoff condition. Due to the complex interactions between the thermal, mechanical and fluid effects, the turbine engine exhibits non-linear behavior. Since the non-linear control design process is too complex and is dependent on initial and current operating conditions, the non-linear engine model is linearized around its operating condition to obtain a steady-state linear model. For a turbo-fan engine, the operating condition around which the non-linear model is linearized is dependent upon the thrust demand (PLA), Mach number and ambient temperature and pressure. The non-linear turbo-shaft engine model is linearized around an operating point which is
expressed as a percentage of the gas generator shaft design speed. Once the linear model is obtained, either frequency-domain or time-domain controls design methodologies can be used to design a control law to regulate the engine performance in the neighborhood of the operating condition. A proportional–integral–derivative controller (PID controller), with gains scheduled based on the altitude and Mach number can also be used as a set-point controller. The fuel ratio, \( RU = \frac{W_f}{P_{s3}} \) is a widely accepted fuel control variable because it provides a good control of turbine gas temperature along with simplifying the control law design process. In an event of an engine stall or surge, the compressor exit pressure drops abruptly reducing the fuel flow rate going into the main burner.

2.4.5. Transient Control Design

The objective of the transient controller is to design a controller that will aid the transition of the engine from one steady state to the other while maintaining the entire engine operational limits and by providing an acceptable engine performance. Development of the transient schedules accounts for a significant amount of control design effort because of the non-linear nature of the engine. Gain scheduling, integrator wind-up protection and N-dot control are some of the transient control design methods [2]. Figure 2.3 shows the schematic of the transient controller.
2.4.6. Limit Controllers

The operating limits of the turbine engine include physical limits on shaft speed, maximum turbine blade temperature, maximum combustor pressure and compressor surge and stall limits. Mismatch between low speed and high speed shafts due to engine deterioration, speed sensor failure or fuel metering valve failure can cause the low or high speed shaft to exceed its limit. Shaft overspeed can result in disk burst or break-off of the compressor or turbine blade which can compromise both the engine and aircraft safety. Engine shaft overspeed also results in an increase in compressor exit static pressure. Hence, an upper limit on the compressor exit static pressure is necessary to prevent damage to the combustor liner and a low limit is necessary to maintain engine performance at idle conditions. High turbine temperatures cause erosion of turbine blades requiring an expensive turbine overhaul or replacement. There are several limits of maximum turbine temperatures. The upper limit is a strict material limit which cannot be exceeded under any conditions. The lower limits can be exceeded by the engine during extreme maneuvers like missile avoidance or aborted landing attempts, but necessitate immediate turbine overhaul. Acceleration limiter prevents high pressure compressor stall...
of a deteriorated engine during sudden, large changes in thrust demand. Limit controllers are generally designed using a PID based control law. Turbine engine control systems designed using both classical and modern control theory is discussed in detail in [16] along with the use of advanced control design techniques like sliding mode control, model predictive control and linear regulators for engine limit management.

As shown in Figure 2.4, set-point, limit and transient controllers are integrated using a min-max logic in which, the lowest or the highest values of the control input, i.e. fuel flow rate is used. This results in a sub-optimal engine performance for a new engine, but guarantees that the engine parameters do not exceed their limits during the entire engine life operation.

![Figure 2.4: Limit control design schematic](image-url)
2.5. **Advanced Control Methods**

2.5.1. **Engine Degradation and Health Management**

Since turbine engine is a safety-critical application, safety and reliability are the two most important aspects which must be considered during the design of FADEC. The FADEC performance must be independent of the engine health, i.e. performance of a deteriorated engine should similar to the performance of a new engine. This can be achieved by implementing an engine monitoring and health management system. Use of health management systems can help in transitioning from time-based maintenance to condition-based maintenance, resulting in an increased time-on-wing and reduced maintenance cost. The engine monitoring and health management systems should consist of diagnostic algorithms to detect and isolate faults and prognostic to predict failures and to estimate the remaining component life. These faults can be abrupt transient component failures or due to slow component degradation. A survey of advanced, intelligent control and health management technologies to improve engine performance while reducing life-cycle cost, noise and emissions is discussed in [17].

2.5.2. **Multivariable Control**

As discussed earlier, a typical gas turbine engine has three main control inputs: fuel flow rate, VBV and VSV. VBV and VSV are open-loop scheduled based on compressor temperatures and pressures while fuel flow rate is designed using traditional single-input, single-output (SISO) control design techniques to achieve the desired system performance. In multivariable control design, an optimum control law is calculated which
uses all the available control inputs (actuators) to achieve the desired control objective along with minimizing (or maximizing) a performance function. The benefits of multivariable turbine engine control design are well discussed [18] and [19].

2.5.3. Active Control

Active turbine engine control strategies can be broadly divided into active clearance control, active stall/surge control and active combustor control. Tip clearance during cold operation and hot operation of compressors and turbine often reduces the component efficiency. Active clearance control will help to minimize the loss by actively reducing the tip clearance during cold operations. Active clearance control can also help in reducing tip rubs, thereby improving the component life. Active clearance control will use high response actuators to modulate the axial tip clearance of compressors and turbines. As shown in figure 2.5, close loop control will be provided by using high bandwidth tip clearance measurement sensors. Ref. [20] presents a study on the requirements and design of active tip clearance control systems for turbine engines while Ref. [21] discusses the current and future research issues for turbine clearance control systems. Figure 2.5 demonstrates the coupling between active clearance control subsystem and turbine engine control system [20].

High thermal and aerodynamic loading of compressors cause rapid, unsteady pressures and temperature fluctuations. These compressor instabilities degrade the component life and can result in compressor surge or stall. Active stall and surge control avoid engine flameouts by preventing compressor stall/surge conditions. The unsteady interactions
between high pressure and heat release in the combustion chamber induce rapid thermo-acoustic oscillations, also known as thermo-elastic instabilities. These combustion instabilities can be minimized by the use of active combustion control. In active combustion control, either the entire or fraction of total fuel flow is controlled by using high-response fuel actuators.

![Figure 2.5: Active clearance control system](image)

Figure 2.5: Active clearance control system

### 2.5.4. Model-based Control

The conventional engine control system is designed based on the worst-case operation of a degraded engine, i.e. based on stall/surge margins of a degraded engine. Although this ensures safe and reliable operation of the deteriorated engine, a new or partially degraded engine does not operate at its optimum efficiency leading to increase in emissions and high specific fuel consumption. Replacing the sensor output based control system by a
model-based control system can result in an efficient engine operation at all levels of engine degradation. An aero-thermodynamic model of an engine is used along with a tracking filter to estimate and directly control the thrust and also to maintain the stall/surge margins. The tracking filter is used for real-time updating of the engine model based on the sensor data and can be designed using a Kalman filter, an extended Kalman filter or classical observers. Model predictive control (MPC) is one of the promising model-based control design techniques which is based on iterative, finite horizon optimization to minimize a performance criterion subject to both input and output constraints. Use of MPC will also replace the min-max control architecture resulting in more efficient engine operation at all degradation conditions and also in presence of sensor/actuator faults [22]. The advantages of using a model-based control methodology for both engine health management and control design is presented in [23].

Figure 2.6 shows an integrated flight and propulsion control system based on model-based control design methodology. A tracking filter is used to update the on-board engine model as well as for engine health management. The tight integration between flight and propulsion control improves the maneuverability of the aircraft in an event of either flight or propulsion control system failure [24].
Figure 2.6: Model-based control system with health management
CHAPTER 3

DISTRIBUTED ENGINE CONTROL SYSTEMS

3.1. Introduction

The next generation turbine engines are expected to deliver high thrust-to-weight ratio along with a significant reduction in fuel consumption and carbon emissions. Use of advance materials, improved turbine engine component designs and advanced, intelligent control design methodologies will help to achieve these objectives. However, along with these objectives, there is a strong incentive for reducing the development, procurement and life cycle cost of a gas turbine engine. As discussed in chapter 2, the performance of a gas turbine engine can be improved by implementing intelligent control methodologies like model-based control, multi-variable control, active control systems and advanced health management techniques. However, implementation of these advanced and intelligent control systems necessitate a transition from the current analog control system architecture to a new distributed control system architecture. In an engine control system based on distributed architecture, the legacy sensor/actuator will be replaced by a smart sensor/actuator node. These smart nodes will be connected to the FADEC via a serial, digital data bus. The smart sensors will consist of the baseline analog sensor and the
electronics required for signal conditioning to convert the analog sensor data to digital data and data network interface unit to transmit the data using the communication protocol. All smart nodes will be designed to include a common data network interface unit, thereby, reducing the smart node design cost. If the smart sensor/actuator includes extra electronics to perform additional functions like fault compensation or local loop closures, then it will be called as an intelligent node. The engine area network (EAN) will be used to connect all smart/intelligent nodes and will house the data bus required for serial communication as well as the power bus to supply power to all of the nodes. The benefits of distributed engine control are widely discussed in [8],[7], [6], [25] and [26]. Some of the perceived benefits of the distributed engine area network are:

1. Enhanced thermal management

To minimize the wire harness length, the FADEC unit is often housed on the engine fan casing or near the combustor and is subjected to extreme vibrations and high temperatures. Since, the maximum junction temperature of current FADEC is limited to 125 °C, jet engine fuel is used to cool the FADEC systems and is either used up in the combustor or circulated back to the fuel tank. The FADEC cooling requirements often limit the ground idle operations and flight time for high performance engines. The use of an engine area network removes the location restrictions and allows it to be housed in more benign environment. The use of airframe mounted FADEC based on a low power, high performance multi-core processors along with electro-mechanical actuators will further
enhance the thermal management systems, thereby increasing the flight times and ground idle operation times.

2. Improved control performance

The reduced vibrational and thermal loads on FADEC housing will allow the use of high performance multi-core processors. This will facilitate the use of resource heavy control design methodologies like model based control, model predictive control as well as the use of improved health management and diagnostics algorithms. The use of electro-mechanical actuators, smart sensors and intelligent actuators will also facilitate the implementation of active control systems.

3. The availability of digital sensor data, improved data sharing between multiple FADECs and availability of high processing power will improve the aircraft handling qualities through the use of an integrated flight and propulsion control design.

4. Due to the modular nature of the distributed engine control system, rapid control system reconfiguration will help to improve operational flexibility by allowing the ground crews to reconfigure the engine or the aircraft to accomplish its new mission objective.

5. The improved fault diagnostic and prognostic capabilities will facilitate condition based maintenance thereby increasing the on-wing time and reducing the overhaul and maintenance costs.

6. Use of improved fault detection capabilities, advanced control and diagnostics algorithms and improved engine control system design will improve the engine
safety and reliability by reducing the mean time between failures (MTBF) and increasing the engine component service life.

7. The modular design of the distributed engine control system will simplify the certification process by reducing the time and effort required for the verification and validation of the FADEC as well as the turbine engine.

8. Turbine engine life cycle cost can be reduced significantly by designing the distributed control systems based on planned obsolescence. In this strategy, the product obsolescence aftermath is considered during the initial product design phase. This reduces the product inventory which has to be maintained throughout the engine life cycle. Also the user can replace the faulty component by a new improved component instead of remanufacturing the component based on an old design.

9. Use of modular design, simplified certification process, reduced logistics footprint and use of open standards for component design will lower the design, procurement as well as life-cycle cost for the turbine engine control systems.

Since the engine area network will be the core module of the distributed control system, the engine area network design will be one of the crucial steps of the engine control system design. The communication protocol of the engine area network should not only meet the design requirements of the current control system, it should also be future-proof and allow easy integration with the engine control system technologies which are currently in the initial development phase. The engine area network should also supply
primary regulated power supply and backup power supply to all sensor and actuator
nodes. New certification guidelines for the distributed engine control system should also
be developed to standardize the component design and testing process as well as
developing component design and engine area network interface standards. One of the
main technical challenges for implementing the distributed turbine engine control system
is the availability of high temperature electronics. Since the A/D, D/A, signal
conditioning and other FADEC functions will be distributed to the smart nodes,
additional electronics should be embedded into the smart nodes. These smart nodes,
which are often placed in high temperature environment, will have to be made rugged to
prevent mechanical damage due to severe vibrations and high temperatures.

3.2. Engine Area Network

The engine area network, one of the most crucial modules of the distributed engine
control systems, will supply power to all smart nodes as well as enable data transmission
between all smart nodes and FADEC. The data transmission will be based on a
communication protocol, which will define the rules for packaging the data, error
detection, correction and node clock synchronization. The engine area network will be
based on a common standard which will include the communication protocol guidelines,
power supply requirements, compatible physical layers and certification guideline. The
communication media of the engine area network can be either an electric cable, fiber
optic cable or based on wireless communication. Twisted pair or coaxial cable will allow
serial data transmission and will require shielding to minimize electromagnetic
interference and data distortion. Multi-mode optical fiber cables will provide high data bandwidth with high reliability with low attenuation and interference than an electric wire harness. Secured, low power wireless communication is still at its infancy stage and needs to be further developed before it is used for safety-critical applications. Communication of real-time systems can be classified either as time-triggered communication or event-triggered communication, depending upon the triggering mechanism. If the execution of the task, which may include data transmission or data processing, is based on the progression of the internal real-time clock of each node, then it is classified as time-triggered communication. The methodology for design of distributed real-time systems based on time-triggered communication mechanism, static scheduling and globally synchronized time-base is presented in [27]. As shown in [27], precise synchronization of all node internal clocks is very crucial for a successful operation of time-triggered systems. Each node is scheduled to transmit data periodically at fixed time interval, i.e. after passage of a predetermined clock ticks. Ref. [28] presents an in-depth review of the design principles and methodologies for design of distributed real-time safety-critical systems based on time-triggered communication protocol. If the triggering mechanism is based on an external event and not on the internal real-time clock, then it is known as event-triggered system. The triggering mechanism of event-triggered systems occurs as soon as the design parameter exceeds a pre-determined threshold. A comparison between event-triggered and time-triggered communication protocols for distributed control systems is discussed in [29]. For safety-critical DCS, there is a clear preference for time-triggered protocols over the event-triggered protocols.
Time-triggered protocols offer high level of reliability with fault-tolerance. A time-triggered system supports replica determinism, supports temporal composability and allows error detection. However, these systems are less flexible to design changes as the message priorities have to be reassigned, have a minimum response time equal to the designed time slot period and also have higher bandwidth utilization. Event-triggered protocols have higher flexibility and support easy addition/deletion of nodes. They also are resource efficient as the bandwidth is used only when necessary. An ideal communication protocol should allow for both time-triggered and event-triggered message transmission. Broadcasting is a communication method of transmitting the data packet to all recipients simultaneously. Broadcast communication networks can be designed based on a token-passing principle, master-slave principle, carrier sense multiple access/collision detection (CSMA/CD) or time division multiple access mechanism (TDMA). ARCNET, fiber distributed data interface (FDDI) are some of the protocols based on token-passing principle, in which the node possessing the token is allowed to transmit before passing it to the next node. The use of this protocol has declined due to their highly unreliable operation. Profibus, local interconnect network (LIN) are protocols designed on master-slave principle for low cost, reliable communication. Since the data packet transmission is initiated by the master, collision detection and avoidance mechanisms are not required. If the network architecture does not allow a master-slave configuration, a carrier sense multiple access/collision detection based protocol can be used. According to this protocol, only the node having highest message priority can transmit data. CAN, ARINC 825 are few examples of protocols.
which employ CSMA/CD technique. TTP/C, FlexRay and TTEthernet are few protocols which are designed based on TDMA mechanism. Smart nodes transmit data only within the predetermined time slot, which is determined by the local synchronized clock. This guarantees deterministic transmission latency and makes the network tolerant to faults. Time-Triggered Ethernet (TTEthernet), Avionics Full-Duplex Switched Ethernet (AFDX), ARINC 664 are some of communication protocols which offer both time-triggered and event-triggered data transmission. This allows transmission of time-critical data packets required for deterministic, safety-critical application along with non-critical, low priority data packets for diagnostic purposes. TTP/C, TTEthernet use quartz based internal clocks for time-triggered communication while protocols like ARINC 825 use software based techniques to ensure time-triggered communication behavior. Comparison of several communication protocols on the basis of their fault management strategies, topologies and implementation is discussed in [30], [31] and [32]. The benefits as well as the technological challenges in the design of aircraft control and health management systems based on wireless sensor networks is briefly discussed in [33]. A brief review of several candidate communication protocols is presented below.

3.2.1. Flexray

FlexRay is a communication protocol which was designed to replace the Controller Area Network (CAN). FlexRay allows transmission of both synchronous (time-driven) and asynchronous (event-driven) data transmission with high bandwidth and deterministic behavior. The static segment of FlexRay is similar to TTP and is based on a time division
multiple access scheme. The dynamic segment is based on the Byteflight mini-slotting protocol. A bus guardian is designed to supervise data transmission and can disable the network node in case of node failures or mismatches in the time schedule. FlexRay provides dual redundant data transmission. If redundancy is not required, both the channels can be used to transmit different set of messages, thereby increasing the channel bandwidth. However, this protocol was developed only for automotive application and needs to be further refined for use in aerospace applications. AUTomotive Open System ARchitecture (AUTOSAR) is currently working on developing an International Organization for Standardization (ISO) standard for Flexray.

3.2.2. AFDX

Avionics Full-Duplex Switched Ethernet (AFDX) is communication protocol based on Ethernet and offers highly deterministic communication with high bandwidth. AFDX was developed to replace ARINC 429 currently used in aircraft navigation systems and flight control systems. Deterministic communication is provided by limited bandwidth, dual redundancy and bounded jitter and latency. The maximum bandwidth supported by AFDX is 100 Mbps which is shared by the virtual links (VL). Bandwidth allocation gap (BAG) and AFDX frame is assigned to each VL. The largest AFDX frame size is 1518 bytes which include a maximum data payload of 1471 bytes. For data payloads less than 17 bytes, the minimum frame size is 64 bytes. BAG is the minimum time delay between two consecutive AFDX frames and ranges in powers of 2 from 1 to 128 milliseconds. VL
scheduling mechanism ensures that the maximum jitter is maintained below 500 microseconds.

3.2.3. TTEthernet

TTEthernet is a time-triggered communication protocol developed by TTTech Computertechnik AG. It allows critical control system to coexist with audio/video and standard LAN systems on the same network. The applications of TTEthernet include time-critical, deterministic or safety-relevant systems. TTEthernet can be used for high-speed active control systems, smart control systems, critical audio/video delivery systems, modular control systems. It can be also used for safety critical embedded systems in aerospace and defense, automotive, medical, energy production, and industrial automation. The TTEthernet standard allows easy integration with the ARINC 664 as well as classical Ethernet network. The standard also defines a fault-tolerant self-stabilizing synchronization strategy, in which, the devices synchronize their local clocks to each other using a fault-tolerant technique [34]. The logical layer of TTEthernet supports hard real-time, rate-constrained and unconstrained traffic on a common mixed-criticality network using following three different message categories:

i. Time-triggered messages:

These messages are used for periodic data transmission. The delivery of these messages and time delay is guaranteed. Their temporal precision is as accurate as necessary. Sensor
data message, which is necessary for control system operation, is periodic in nature with a fixed sampling time and will be labeled as time-triggered messages.

ii. Rate-constrained messages:
This message type is used for nodes which have less stringent determinism and real-time requirements. The bandwidth used for these messages is predefined and the time delays sustained by these messages are within a tolerable limit.

iii. Best-effort messages:
Best-effort message type is used for low priority messages and uses the available bandwidth for data transmission. There is no guarantee on the successful transmission of the message and also on the time delays which will be encountered for the message. Non safety-critical messages like the data required for maintenance can be transmitted using this message type. For example, the vibration sensor data, which is not used for control purposes and is used only for maintenance, can be transmitted using this data type.

3.2.4. ARINC Specification 825
Controller Area Network (CAN), a serial communication bus network, was initially developed for application in an automotive industry. Due to the many advantages of CAN, including its high reliability and cost effectiveness, it has found application in other industries, including aerospace industry [35] and [36]. The use of CAN for real-time applications is studied using both simulation tools and experimental test systems in
ARINC 825 was developed by the CAN Technical Working Group of the Airlines Electronic Engineering Committee (AEEC) Aircraft Network Infrastructure and Security Subcommittee and is suitable for both military applications as well as general aviation.

Some of the advantages of ARINC 825 include:

1. Low implementation cost
2. High Modularity
3. Can be easily connected to other aircraft CANs.
4. Good error detection and error signaling capability
5. Support of system level functions
6. Data bandwidth of 1000 kbits/s
7. Predictive behavior of system due to time-triggered bandwidth management
8. Optimum use of bandwidth due to message prioritization.
9. Support of design of gateways between CAN and other networks by providing information on protocol conversion, bandwidth management and fault isolation.

ARINC 825 allows both synchronous and asynchronous data transmission. However, this protocol does not use a global time for node synchronization and utilizes time-triggered bandwidth management, which is a software-based approach for achieving deterministic behavior. ARINC 825 uses message prioritization technique in which each node is designed to have a specific message priority. Whenever two or more nodes attempt to transmit data, the node with highest priority is allowed to transmit the message. This
eliminates data loss due to packet collisions. In addition, it offers low cost, high modularity and predictive behavior of system due to time-triggered bandwidth management.

**Bandwidth Management**

Since CAN is based on a broadcast communication protocol, all nodes participate in the communication and health monitoring, thereby ensuring inherent data consistency between all active nodes. Message prioritization is used to avoid multiple nodes from accessing the data network at the same time. If two or more nodes attempt to access the data network to transmit data at the same time, only the node with highest priority (lowest numeric identifier) is granted access to the data bus and all other nodes have to withdraw and retry the data transmission in next transmission window. This CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) arbitration mechanism avoids data loss due to data packet collisions, but it can increase the data transmission latency for nodes with low message priorities. Predictable data latency is guaranteed for all nodes by the implementation of bandwidth management strategies. This is achieved by designing the node transmission based on major and minor time frames and by controlling the transmission rate of each node connected to the network. Major time frame period is defined as the lowest sampling period and minor time frame period is defined as the highest sampling period of the time-triggered nodes.
3.3. **Data Bus Requirements**

1. The data transmission must be reliable and must include error-detection code embedded within the message.

2. The communication protocol must provide deterministic timing performance including bounded jitter, predictable transmission delay and latency for time-sensitive data such as engine warning data and constant latency with minimum, bounded jitter for control-critical data transmission.

3. The bandwidth utilization should be less than 50%, which will allow addition of new sensors/actuators and also allow future reconfigurations in the architecture.

4. The communication protocol should have a mechanism to detect clock offset and to synchronize internal node clocks.

5. The fail-operational feature of the communication protocol should detect and isolate faulty nodes.

6. The data frame should include source identifier and a time stamp to indicate the time of data transmission.

7. The communication protocol of the EAN should be compatible with the data busses used for other aircraft systems such as flight control systems, navigation systems, etc.

8. The EAN should provide the desired performance under different system workloads.

9. The communication protocol should be based on an open standard and should not require expensive licensing fees.
10. Commercial off-the-shelf (COTS) software and hardware packages should be readily available for the OEM supplies to design and evaluate the performance of their components.

11. The EAN should have failure rate of less than $10^{-9}$

12. The EAN must meet the requirements for aviation data bus given in the Advisory Circular 20-156 issued by Federal Aviation Administration (FAA).

Since these requirements are best met by ARINC 825, the proposed partially distributed control design methodology is discussed based on the ARINC 825 protocol features.

3.4. Communication Constraints

The addition of an EAN will impose additional constraints on engine control system design. Control systems in which data transfer takes place via a real-time communication network are known as networked control systems (NCS). These additional constraints include limited communication bandwidth, network-induced time delay, sampling jitter and packet dropouts [41]. These communication constraints may not only degrade the control system performance but may also destabilize the control system if they are not included in the design process. An overview of control design methodologies for networked control systems to guarantee desired performance under these communication constraints is discussed in [42], [43],[44] [45], [46] and [47]. The different modeling approaches for stability analysis and controller design of time delay systems is presented in [48] and [49]. In [50], it is shown that the stability and performance of networked control system is also dependent also on node message scheduling. Ref. [51] presents an
approach to identify the timing requirements of real-time NCS based on message scheduling. The effects on these constraints on the stability and performance of distributed engine control systems will be briefly discussed below.

3.4.1. Constraint on Channel Bandwidth

The capacity of the communication network to carry a finite amount of information per unit amount of time is known as channel bandwidth. The network must have sufficient bandwidth and latency to enable closed loop control and must also be robust to accommodate the safety and critical functions. Also, the bandwidth of the physical layer should be sufficient enough to support the required number of smart nodes and should allow for future addition of smart nodes. Bandwidth utilization is the ratio of bandwidth utilized for data transmission to the total available bandwidth and will depend upon the number of sensor nodes, size of data packet and node sampling rate.

3.4.2. Network-induced Time Delays

Network-induced time delays include transmission time delay, propagation time delay, time required for sensor data acquisition, signal conditioning and control law processing. The transmission time will be dependent upon the channel bandwidth and size of data packet. The propagation time will be determined based on the physical medium of the data network and is in the range of $2 \times 10^8$ m/sec for copper wires and $3 \times 10^8$ m/sec for fiber optic media. Since physical separation between the network nodes and total length of data network will be small, the finite propagation time will be a fraction of the total
network-induced time delay and can be safely discarded. The signal conditioning and
data processing time delay will be introduced at the smart sensor location. The total time
delay between the transmission of first sensor data packet and reception of last sensor
data packet is the sensor-controller time delay. The time required for the controller to
process the data and calculate the control input value is the controller time delay. There
will be also a time delay between controller and actuator, which is classified as
controller-actuator time delay. Time delays also arise due to the physical processes such
as transportation delay for fuel flow between fuel control valve and fuel nozzle or time
delay due to fuel-air mixing. Although the individual time delays may be very small, the
combined effect of all of these delays can significantly affect the control system
performance and result in performance degradation or loss in control system stability.

3.4.3. Network-induced Data Loss
Congestion in data networks, transient node failures or transmission errors can result in
unsuccessful data transmission. Most of the communication protocols use an error-
detecting mechanism through a cyclic redundancy check (CRC) code. The CRC code is
embedded with the message frame and used to detect data loss due to fault data network
and to guarantee data integrity. If the message frame fails the CRC test, the receiver can
either request a retransmission attempt or use an estimator to predict the lost sensor data
value. If both of these options are not feasible or for worst-case analysis, the controller
can assume that there is no change in sensor data value and use the previous sensor value
for control law calculation. Similarly, the actuator can assume that there is no change in
control input value and continue to hold its position until it receives a new control input value.

### 3.5. Sensor Requirements

Gas turbine engine sensors are used for overall engine control, condition monitoring and for providing warning indication. The control system sensors measure gas path parameters such as temperatures, pressures, shaft speeds and shaft torque. Fuel valve, VBV and VSV positions are measured using linear variable differential transformer (LVDT). A fuel flow measurement sensor is also used to measure the mass flow rate of jet fuel. Condition monitoring and diagnostic sensors include sensors for measuring vibrations, oil quantity and oil debris detectors. The FADEC processes the data from these diagnostic and control sensors to provide warning indications to the pilot.

Resistance temperature detectors (RTD) or thermocouples are used for gas path temperature measurements at $T_{2.5}$, $T_3$, $T_{4.5}$ stations. Non-contact sensor like fuel cooled radiation pyrometer can be used to directly measure the turbine blade temperature. A dual pressure and temperature probe designed to prevent ice and debris is used to measure inlet pressure and total air temperature. MEMS pressure transducers or piezoelectric pressure sensors are used for static or dynamic pressure measurements. Rotational speed is measured using speed sensors which operate on the principle of electromagnetic induction. Piezoelectric accelerometers for used for vibration detection and measurement. Sensors are selected based on the required sensitivity, resolution, measurement range and sensor bandwidth. Ideal sensors should be light weight, have high reliability, low power
requirements and should have low procurement cost. Since few of these sensors are
placed near the gas path, rugged sensor packaging schemes should be employed for
protecting the sensor electronics from high temperatures and extreme vibrations.
The advanced engine health monitoring and active control techniques call for installation
of additional sensors. New sensors are being developed for measuring blade tip clearance,
fuel properties, burning pattern factor, exhaust gas composition and fuel quality and
carbon emissions. Temperature at engine intake and compressor stations is approximately
750 °C while at combustor station can go up to 1700 °C. The current silicon based
sensors can sustain only up to 125 °C. Research is being conducted to manufacture
sensors using Silicon-on-Insulator (SOI) for operations up to 300 °C, SiC for
temperatures up to 500 °C and sensors based on ceramic material SiCN are expected to
be operational up to 1700°C. It is widely expected that these new sensors will be
developed using advanced materials and will be available for on-engine installation in
next 10-15 years. The distributed engine control system architecture should be designed
to allow easy future installation of these sensors. The sensor and actuator requirements
for future turbine engine control systems is presented in [52]. The effects of high
temperature environment and atmospheric radiation on design of future sensors and
control system electronics is discussed in [53]. Bit-rate requirements analysis for engine
control systems based on both baseline centralized and distributed architectures was
presented in [54] for a generic communication protocol. As shown in table 3.1, the
minimum required bandwidth for current legacy sensors and actuators is approximately
52,074 bits including the ARINC 825 communication protocol overhead.
Table 3.1: Minimum required bit-rate for engine control system sensors

<table>
<thead>
<tr>
<th>Number</th>
<th>Parameter</th>
<th>Update Rate (Hz)</th>
<th>Message Payload Length (bits)</th>
<th>Maximum Frame Size (bits)</th>
<th>Bit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ambient total pressure</td>
<td>25</td>
<td>16</td>
<td>99</td>
<td>2475</td>
</tr>
<tr>
<td>2</td>
<td>Fan inlet total temperature</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>3</td>
<td>LP spool speed</td>
<td>20</td>
<td>32</td>
<td>198</td>
<td>5742</td>
</tr>
<tr>
<td>4</td>
<td>Fan discharge static pressure</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>5</td>
<td>HP spool speed</td>
<td>20</td>
<td>32</td>
<td>198</td>
<td>5742</td>
</tr>
<tr>
<td>6</td>
<td>HPC inlet total temperature</td>
<td>20</td>
<td>16</td>
<td>99</td>
<td>1980</td>
</tr>
<tr>
<td>7</td>
<td>HPC inlet total pressure</td>
<td>20</td>
<td>16</td>
<td>99</td>
<td>1980</td>
</tr>
<tr>
<td>8</td>
<td>Compressor discharge static pressure</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>9</td>
<td>Compressor discharge total temperature</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>10</td>
<td>LPT inlet total temperature</td>
<td>20</td>
<td>16</td>
<td>99</td>
<td>1980</td>
</tr>
<tr>
<td>11</td>
<td>Turbine discharge total temperature</td>
<td>20</td>
<td>16</td>
<td>99</td>
<td>1980</td>
</tr>
<tr>
<td>12</td>
<td>Variable bleed valve</td>
<td>20</td>
<td>32</td>
<td>198</td>
<td>3960</td>
</tr>
<tr>
<td>13</td>
<td>Variable stator vanes</td>
<td>20</td>
<td>32</td>
<td>198</td>
<td>3960</td>
</tr>
<tr>
<td>14</td>
<td>Transient bleed control</td>
<td>20</td>
<td>32</td>
<td>198</td>
<td>3960</td>
</tr>
<tr>
<td>15</td>
<td>Fuel metering valve</td>
<td>20</td>
<td>32</td>
<td>198</td>
<td>3960</td>
</tr>
<tr>
<td>16</td>
<td>Power lever angle (PLA)</td>
<td>20</td>
<td>16</td>
<td>99</td>
<td>1980</td>
</tr>
<tr>
<td>17</td>
<td>Mach number</td>
<td>25</td>
<td>16</td>
<td>99</td>
<td>2475</td>
</tr>
<tr>
<td>18</td>
<td>Altitude</td>
<td>25</td>
<td>16</td>
<td>99</td>
<td>2475</td>
</tr>
<tr>
<td>19</td>
<td>Total air temperature</td>
<td>10</td>
<td>16</td>
<td>99</td>
<td>990</td>
</tr>
<tr>
<td>20</td>
<td>Fuel temperature</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>21</td>
<td>Engine oil pressure</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>22</td>
<td>Engine oil temperature</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>23</td>
<td>Case temperature</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>24</td>
<td>Engine oil quantity</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>25</td>
<td>Ignition switch</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>26</td>
<td>Thrust reverser</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>27</td>
<td>HP Shut-off Valve</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td>28</td>
<td>Take-off/go-around switch</td>
<td>5</td>
<td>16</td>
<td>99</td>
<td>495</td>
</tr>
<tr>
<td></td>
<td>Total (bits/sec)</td>
<td></td>
<td></td>
<td></td>
<td>52,074</td>
</tr>
</tbody>
</table>
3.6. **Distributed Engine Control Architectures**

The smart nodes of the distributed engine control system will be required to have additional electronics for signal conditioning and for encoding the data based on the communication protocol. Since these smart nodes will be located in an environment of extreme vibration and high temperature, these nodes will require the use of high-temperature semiconductor technologies. Due to the limited availability of high temperature electronics, the transition from a centralized architecture to a fully distributed architecture will be restricted to an incremental transition process via a partially distributed architecture [55]. A distributed engine control system which has both legacy sensor/actuators and smart nodes is known as a partially distributed engine control system while a distributed engine control system with only smart or intelligent nodes is known as a fully distributed engine control system. From the controller design standpoint, there can be three distinct architectures which will affect the control design and stability and performance analysis. However, there can be several architectures derived from these three distinct architectures depending on the control system design objective and physical implementation limitations. A brief overview is presented for each of these architectures along with their benefits and their impact on the bandwidth utilization.

3.6.1. **Partially Distributed Architecture with Data Concentrator**

This architecture uses the legacy analog sensors and actuators but redistributes some of the functionality of the centralized FADEC to the data concentrator. The analog sensors and analog actuators are connected to the data concentrator through analog wire harness.
The data concentrator performs the A/D and D/A signal processing and conditioning and transmit the data to the distributed FADEC module via a digital communication bus. The data concentrator, in order to be located centrally, is placed near the core and thus may suffer detrimental heating effects. However, the new FADEC unit won’t have any location restrictions and can be placed at a location having more favorable operating conditions. This will reduce the thermal management requirements of FADEC and will allow the use of advance, intelligent model-based control methodologies. Figure 3.1 shows the partially distributed FADEC with data concentrator.

Figure 3.1: Dual-redundant partially distributed FADEC with data concentrator
As seen in Figure 3.1, the engine control system based on the distributed architecture has
dual-redundant FADEC channels, with each FADEC channel connected to an
independent sensor suite, single data concentrator, dual-redundant control-critical sensors
and dual-redundant engine area network. Since this architecture is based on a dual-
redundant system, system failure can be detected; however, additional processing may be
required for failure identification. For example, a fault in any one of the sensors or data
concentrator will render the entire FADEC channel to be determined as inactive channel.
However, if both the data concentrators and FADEC units are connected by a shared
engine area network, failure in any one of the sensors or data concentrator will not have
any adverse effect on the FADEC channel. As shown in Figure 3.2, each FADEC unit
will have access to both the sensor suites and can determine the data consistency at the
controller level. The use of triple redundant engine area network will also increase the
reliability of the entire control system.

This architecture can be also modified to include two or more data concentrators, each
data concentrator connected to adjacent sensors. Each data concentrator will be
responsible for sensor data acquisition, signal conditioning and data encoding of the
connected sensor suite. This modified architecture can also have a mix of smart sensors
and legacy sensors. The smart sensors and the data concentrators will be connected to the
engine area network.
3.6.2. Fully Distributed Architecture with Smart Nodes

If all of sensors and actuators of a turbine engine control system are smart nodes, then the architecture is classified as a fully distributed architecture. The smart nodes will perform signal conditioning and data encoding based on the communication protocol. Although the signal conditioning and fault management functions of the FADEC will be distributed to the smart nodes, the controller will be still based on a centralized scheme. The FADEC will be responsible for data processing and control law calculation. Figure 3.3 shows such an architecture with smart nodes.

Figure 3.2: Triple-redundant partially distributed FADEC with data concentrator
3.6.3. Fully Distributed Architecture with Intelligent Nodes and Decentralized Controller

In this architecture, the controller will be decentralized into local controllers with FADEC acting as a supervisory controller. The local loop closures will be performed by intelligent nodes. These intelligent nodes will be smart nodes with additional computational capabilities to perform local loop closures, fault detection and isolation. The use of intelligent nodes will allow implementation of active control systems such as active stability management, active stall/surge control, active emission control, etc.
The local control loop calculations can be performed either through intelligent sensors or intelligent actuators. Intelligent actuators will contribute to a robust subsystem control performance by making the subsystem robust to sensor failures and will be connected to the required sensors by local analog connections or if required, through low speed, low cost digital data network. The active control systems shown in Figure 3.4 are active combustion instability control (AIC), active surge control (ASG) and active clearance control (ACC). Tip clearance (TC) sensor and actuator is required for ACC while the burner pattern and emission (BPE) sensor is used for loop closure of AIC.

![Dual redundant fully distributed FADEC with active control sub-systems](image)

Figure 3.4: Dual redundant fully distributed FADEC with active control sub-systems
It is important to note the difference between smart and intelligent sensor/actuator nodes. A smart node is a sensor or actuator which has analog interface, analog-to-digital convertor/digital-to-analog convertor, signal processing capability and bus interface module. An intelligent sensor/actuator is a smart sensor/actuator with an additional capability of performing intelligent functions, which can include control law processing, fault detection, health management, etc. Also, it is important to differentiate between control design based on a centralized and decentralized architecture. For a control system design based on a centralized framework, only one control unit is responsible for controlling the individual sub-systems, while for a control system design based on a decentralized architecture, each sub-system will have its own local control units with one control unit acting as a supervisor.

3.7. Partially Distributed Engine Controller for Turbo-shaft Engine

As discussed in chapter 2, a turbine engine control design is an integration of three types of controllers: set-point controllers, transient controllers and limit controllers. Since turbo-shaft engines are always designed to operate at a constant power turbine speed, design of set-point controller for turbo-shaft engine is less complex than for a turbo-fan engine. Due to the control design simplicity and availability of non-classified public domain engine models [56], [57], [58], [59] and [60], a turbo-shaft engine was selected to demonstrate the distributed engine control system architecture design process. The same engine model will be used in the next chapter to validate the control law design process and for performance analysis. The proposed control design methodology can be extended
to turbo-fan engine control system with some minor modifications. T700 turbo-shaft engine is one of the most widely used turbo-shaft engines and is used for powering both military and commercial helicopters including UH-60 Black Hawk and the AH-64 Apache. It is a two-spool engine consisting of a gas generator section and free power turbine section. Since the power turbine extracts work from gas turbine section, there is a one way coupling between power turbine and gas generator turbine.

The traditional T700 Engine Control System consists of two major components, Hydromechanical Unit (HMU) and Electrical Control Unit (ECU). The control objectives of T700 engine control system are

1. The engine control system must maintain a constant power turbine speed in presence of rotor load disturbance via collective pitch and ambient operating conditions.

2. The engine control system must maintain an adequate stall margin.

3. The turbine inlet temperature ($T_{45}$), power turbine ($N_{p}$) and gas turbine ($N_{g}$) shaft speed should not be exceeded beyond a tolerable limit.

4. The engine control system should ensure that the torque load is effectively shared between two engines.

The T700 turbo-shaft engine control system has three control variables- fuel flow rate, compressor bleed valve and variable guide vanes. The compressor bleed valve and variable guide vanes are both open loop controlled based on a schedule of gas generator turbine speed $N_{g}$ and compressor inlet temperature $T_{2}$. ECU provides an electrical trim
signal to HMU for controlling the fuel flow rate. A typical T700 engine control system requires following sensor data -

\begin{align*}
N_{g1} &= \text{Gas generator speed sensor} \\
N_{g2} &= \text{Redundant gas generator speed sensor} \\
N_{p1} &= \text{Power turbine speed sensor} \\
N_{p2} &= \text{Redundant power turbine speed sensor} \\
N_{p\ O/S} &= \text{Power turbine over-speed sensor} \\
N_r &= \text{Rotor speed sensor} \\
Q_{s1} &= \text{Power turbine shaft torque engine 1} \\
Q_{s2} &= \text{Power turbine shaft torque engine 2} \\
T_1 &= \text{Inlet temperature} \\
T_2 &= \text{Compressor inlet temperature} \\
T_{45} &= \text{Power turbine inlet temperature} \\
P_1 &= \text{Inlet pressure to engine} \\
P_{s3} &= \text{Compressor static discharge pressure} \\
LDS &= \text{Load Demand Spindle position} \\
PAS &= \text{Power Available Spindle position} \\
CID &= \text{Collective INC/DEC} \\
BV_f &= \text{Bleed valve position feedback} \\
VG_f &= \text{Variable Geometry position feedback} \\
WF_f &= \text{Fuel Flow Rate feedback} \\
T_{oil} &= \text{Oil Temperature Detector}
\end{align*}
\[ P_{\text{oil}} = \text{Oil Pressure Transmitter} \]
\[ D_{\text{chip}} = \text{Chip Detector} \]

The functions and I/O connections of HMU and ECU are briefly discussed below.

![Figure 3.5: T700 Hydromechanical Unit [61]](image)

**Hydromechanical Unit (HMU)**

The main function of HMU is to govern the fuel flow based on the power control lever (PCL) position, collective position and fuel control input from the ECU. As shown in Figure 3.5, the HMU also houses the actuator which controls the variable geometry consisting of inlet guide vanes and variable stator vanes as a function of gas generator speed and engine inlet air temperature. The compressor bleed valve is also controlled by the HMU and is a function of acceleration and deceleration schedules as well as gas generator speed and inlet air temperature. The compressor inlet temperature \( T_2 \) sensor,
gas turbine speed \( (N_g) \), compressor discharge pressure \( (P_{s3}) \) sensor, Power Available Spindle (PAS) and Load Demand Spindle (LDS) are also connected to the HMU. LDS sensor data is used for collective pitch compensation. HMU limits the gas turbine speed \( (N_g) \) by using the \( T_2 \) and \( N_g \) sensor data.

Figure 3.6 T-700 Electrical Control Unit [61]

**Electrical Control Unit (ECU)**

The ECU reduces the pilot work load by maintaining a constant power turbine speed. The ECU also acts as a limit controller limiting the power turbine inlet pressure \( (T_{45}) \) and power turbine speed \( (N_p) \). Load sharing between the two engines is also performed by the
The ECU is connected to a suite of seven identical temperature sensors which measure power turbine inlet temperature. The ECU is also connected to the torque sensor ($Q_t$) and over-speed sensor ($N_p O/S$) and power turbine speed sensor ($N_p$), which are all housed in the power turbine section. The collective INCR/DECR switches are also connected to the ECU and are used to set the reference for power turbine shaft speed ($N_p$). The ECU transmits the $T_{45}$ and $N_p$ data to the cockpit and $Q_t$ data to other engine ECU. Figure 3.6 shows the electrical connections of the T-700 ECU.

Figure 3.7: T-700 engine control system [61]
Figure 3.7 shows T-700 engine control system. Accessory Section Module (AGB) contains a number of diagnostic sensors like oil temperature detector, oil pressure transmitter, chip detector and fuel filter bypass which transmit diagnostic/health data to the cockpit.

Design of a partially distributed turbo-shaft engine control system with multiple data concentrators is presented below. In this architecture, the functions of HMU and ECU are distributed to three nodes- fuel metering node, variable geometry (VG) control node and bleed valve (BV) control node. Airframe interface (AI) node is used to exchange data with the cockpit and FADEC node for performing the control law calculations. The description of each node is as given below.

Fuel Metering (FM) Node

The main function of fuel metering (FM) node is to maintain a constant power turbine speed and prevent power turbine speed and power turbine inlet pressure from exceeding a predetermined threshold. The redundant gas turbine speed ($N_{g2}$) sensor, redundant power turbine speed sensor, fuel flow feedback ($w_f$) and all other diagnostic sensors like $T_{oil}$, $P_{oil}$, etc are connected to FM node. The FM node also contains the electronics required for A/D, D/A signal conditioning and communication node interface module.
Variable Geometry (VG) Node

The variable geometry (VG) control node is connected to the variable geometry actuators consisting of inlet guide vanes and variable stator vanes and also to the gas turbine speed sensor \( N_g \), compressor static discharge pressure sensor \( P_{s3} \), compressor inlet temperature \( T_2 \) and variable geometry feedback sensor. The main function of this node is to control the variable vanes based on either the schedule governed by \( N_g \) and \( T_2 \) or by the FADEC if the control system is designed based on multiple-input, multiple-output design methodology.

Bleed Valve (BV) Node

The main function of the bleed valve (BV) control node is to control the compressor bleed valve actuator based either on the signal received from the FADEC or the \( N_g \) and \( T_2 \) sensor data received from the VG node. This node is connected to the \( N_p, Q_{sl}, N_p\ O/S, T_{45} \) and \( BV_f \) sensors.

FADEC

This node will house all the circuitry required for data processing and control law calculation. The FADEC unit will not be connected to any sensors or actuators and will receive the data only through the communication network. Hence, this node can be located on the airframe with less stringent cooling requirements.
Airframe Interface (AI) Node

This node will serve as an interface between all other nodes on the engine area network and the cockpit. Data from collective INC/DEC, collective pitch, throttle input and the Torque from the second engine control system ($Q_{s2}$) will be transmitted to all other nodes through the AI node. The $N_p$, $N_g$, $T_{45}$, $Q_{s1}$ and diagnostic sensor data will be transmitted to the copilots display unit (PDU) and the central display unit (CDU) through the AI Node.

Figure 3.8: Proposed partially distributed turbo-shaft engine control system
Figure 3.8 shows the partially distributed engine control architecture with 5 nodes. For simplicity, only one FADEC channel is shown with dual redundant $N_g$ and $N_p$ sensors. The sensor connections are made such that each node has access to the required sensor suite and also to maintain an even distribution of signal processing and condition electronics between all 5 nodes.

3.8. **Estimation of Bandwidth Utilization**

The network throughput is the average rate of successful delivery of data messages over the data network. The channel efficiency or bandwidth utilization is the ratio of the achieved throughput to the maximum available data bandwidth. This ratio, expressed as a percentage, is affected by the communication protocol overhead and determines the number of retransmission attempts or future allowable increase in the number of nodes. Determination of the bandwidth utilization for the proposed partially distributed turbo-shaft engine control system based on ARINC 825 communication protocol is briefly discussed below. Based on literature survey, it is assumed that the dual-redundant engine control system consists of 11 engine sensors with update rate of 100 Hz and two actuators $w_f$ and $V_g$. Control critical sensors $N_g$ and $N_p$ are redundant per FADEC channel, sensors for measuring $Q_s$, $P_{s3}$, $T_{45}$, $T_2$, $BV_{pos}$, $T_1$, $P_1$ are redundant per engine. It is assumed that each actuator has an inbuilt feedback sensor. Non-control critical sensors include sensors used for diagnostic purposes, for example, oil level sensor, fuel level sensor, etc. and have an update rate of 5 Hz. It is also assumed that each sensor data payload is 2 bytes.
The $T_{45}$ sensor has 7 probes and it will transmit both the average value and individual probe temperature. Assuming the maximum available ARINC 825 bandwidth of 1Mbits/sec, the network transmission delay for each data packet can be calculated as shown in Table 3.2. Table 3.2 also shows the overall data frame size which includes data payload and the frame overhead.

Table 3.2: ARINC 825 data frame size and network transmission delay

<table>
<thead>
<tr>
<th>Data Payload (Bytes)</th>
<th>Data Frame Size (bits)</th>
<th>Network Transmission Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>89</td>
<td>0.089</td>
</tr>
<tr>
<td>2</td>
<td>99</td>
<td>0.099</td>
</tr>
<tr>
<td>3</td>
<td>109</td>
<td>0.109</td>
</tr>
<tr>
<td>4</td>
<td>119</td>
<td>0.119</td>
</tr>
<tr>
<td>5</td>
<td>128</td>
<td>0.128</td>
</tr>
<tr>
<td>6</td>
<td>139</td>
<td>0.139</td>
</tr>
<tr>
<td>7</td>
<td>149</td>
<td>0.149</td>
</tr>
<tr>
<td>8</td>
<td>158</td>
<td>0.158</td>
</tr>
</tbody>
</table>

\[
BW \ Utilization \ (\%) = \left( \sum \frac{message \ length \times number \ of \ messages}{transmission \ interval \times data \ rate} \right) \times 100
\]

Bandwidth utilization can be calculated based on either the minor time frame or major time frame as the transmission interval. Since the diagnostic sensor values are transmitted only once in the major time frame duration, they should be included in the minor time frame duration to obtain conservative bandwidth utilization.

Table 3.3 shows the bandwidth utilization when each sensor transmits the data as individual data packet and Table 3.4 shows bandwidth utilization when each node of the proposed partially distributed engine control system encodes and transmits the data of
multiple sensors through a single data packet. Since the second scenario requires less number of data packets and uses less data overhead, the bandwidth utilization also reduces. The bandwidth utilization for both the scenarios is less than 50%. The bandwidth utilization calculated based on minor time frame is more conservative because it assumes diagnostic sensors data transmission during the minor time frame period.

Table 3.3: Bandwidth utilization for individually transmitted data packets

<table>
<thead>
<tr>
<th>Data Transmission Interval (ms)</th>
<th>Data Messages</th>
<th>Total Data Size (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Payload (Bytes)</td>
<td>Frame Size (bits)</td>
</tr>
<tr>
<td>Control Sensors</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>Diagnostic Sensors</td>
<td>200</td>
<td>2</td>
</tr>
<tr>
<td>Actuator Feedback Sensors</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>Control Input</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Total Required Bandwidth 3,583 52,850
Total Available Bandwidth 10,000 200,000
Bandwidth Utilization 35.83 % 26.45 %

Table 3.4: Bandwidth utilization for proposed turbo-shaft engine control architecture

<table>
<thead>
<tr>
<th>Data Transmission Interval (ms)</th>
<th>Data Messages</th>
<th>Total Data Size (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Payload (Bytes)</td>
<td>Frame Size (bits)</td>
</tr>
<tr>
<td>Control Sensors</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>Diagnostic Sensors</td>
<td>200</td>
<td>2</td>
</tr>
<tr>
<td>Actuator Feedback Sensors</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>Control Input</td>
<td>10</td>
<td>6</td>
</tr>
</tbody>
</table>

Total Required Bandwidth 2,316 27,510
Total Available Bandwidth 10,000 200,000
Bandwidth Utilization 23.16 % 13.755 %
Hence, from the above tables, it is seen that the proposed partially distributed control system architecture results in reduced bandwidth utilization.

3.9. **Estimation of Network-induced Time Delay**

Transmission of continuous time signal over a serial communication network consists of data sampling, data encoding, data transmission and data decoding. The total network induced time delay consists of network access delay, network transmission delay and propagation delay. Network access delay is the random wait time for the smart node after data encoding but before it can access the shared network. Network transmission delay is the time required for transmission of the data packet through the physical data network. This time delay is a function of the data packet size and data network transmission rate and is independent of physical separation between the transmitter and receiver. Network transmission delay can be either constant or random depending on the communication protocol, network congestion and channel quality. The time required for the digital signal to travel through the physical medium is called as propagation delay. As the electrical signals in a copper wire or fiber optic cable travel at approximately 2/3 the speed of light, the propagation delay is approximately equal to 5 nanosecond per meter of cable length. The networked induced time delay can be further subdivided into sensor-to-controller delay and controller-to-actuator delay. The serial nature of the data network will allow for transmission of only one sensor data packet at a time, thereby introducing a time delay between transmission of the first sensor value and the last sensor value required for control input calculation.
ARINC 825 has bandwidth of 1 Mbits/s and the frame size can range between 89 bits to 158 bits depending on the data payload. Hence, the sensor-controller delay for one sensor data packet is between 89 $\mu s$ and 150 $\mu s$. Since the controller is event-driven, it will wait for all sensor data packets before performing the control law processing. Hence, assuming 15 control-critical sensors and maximum data payload, the total sensor-controller time delay is $\tau_{SC} = 2.37 \text{ ms}$. Since there are three actuators, the controller-actuator delay is $\tau_{CA} = 0.474 \text{ ms}$. The total transmission delay, is given as $\tau = \tau_{SC} + \tau_{CA} = 2.884 \text{ ms}$.

### 3.10. Network Faults

Data message loss or packet dropouts occurs due to packet collisions or node failures. Although the use of time triggered protocol or bandwidth management strategy based on message prioritization ensures that packet collisions do not occur, the network is still subjected to node failures and data corruption. If the CRC or acknowledgement bit present in the message overhead detects data corruption, then the node can be granted a retransmission attempt. If the active node experiences a transient failure and is not able to transmit the data, the error detection, isolation and containment mechanism of the protocol prevents the faulty node from transmitting the data. This prevents the faulty node from disrupting the data transmission between active nodes and helps to maintain the network integrity. Bit error, bit-stuffing error, CRC error, format error, and acknowledgement error can be detected by the ARINC 825 error management mechanism. Once the error is detected, the node which first detects the errors transmits an
error frame. All other active nodes discard the corrupted data message after receiving the error frame and the faulty node is granted a retransmission attempt. Each node has a receive error counter (REC) and transmit error counter (TEC) for tracking the received and transmitted corrupted messages. ARINC 825 error containment and bus-off management strategy assigns three types of operation modes to each node: error active mode, error passive model and bus off mode. The node is allowed to transmit in both error active and error passive mode, but the passive error flags are used in error passive modes to indicate the node errors to other active nodes. The node is logically disconnected from the data bus in the bus-off mode. The node is entered into the bus off mode when the transmit error counter (TEC) value exceeds 255 and is returned to the error active mode only when the node detects 128 occurrences of 11 consecutive recessive bits, i.e. successful monitoring of at least 128 data packets. This allows the data loss due to packet dropouts to be modeled as a time delay in multiples of sampling time.

If the node exceeds the maximum number of allowed re-attachments, the failure is marked as a permanent node failure and will require the use of fault-tolerant control design methods to reduce the performance degradation or loss in control. Temporary power supply loss, connector defect or a dominant time-out protection will create an internal open circuit within the node leading to partial loss of communication. Erratic data network behavior will increase the bus error rate (BER) resulting in an increase in message corruptions and number of retransmission attempts. Since these failures are difficult to detect and repair, the controller should be designed to reduce the performance degradation caused by these failures.
CHAPTER 4

DISTRIBUTED ENGINE CONTROL DESIGN

4.1. Gas Turbine Engine Control System

The gas turbine engine dynamics arise due to rotating inertias, gas-flow behavior in the compressor and turbine, heat transfer between gas and metal, thermal behavior of the engine and component losses. Control system components like sensors, actuators and engine area network also add to the engine dynamics resulting in a highly complex, non-linear behavior. The main control inputs variables are combustor fuel flow rate, variable bleed valves (VBV), variable stator vanes (VSV) and thrust augmentor for military engines. The measured output variables are component temperatures and pressures, shaft speeds and shaft torque for a turbo-shaft engine. The controlled output or the performance measures are variables such as thrust, stall margins or engine component efficiencies. The engine is anticipated to operate in a varied environment and external factors such as aircraft speed, operating altitude, inlet temperatures and inlet pressure affect the engine dynamics. The engine control system should consider all these external factors along with engine health deterioration to give a consistent desired performance while maintaining all the mechanical limitations of the engine. The engine mechanical
limits include the shaft speed limits, turbine and compressor pressures and temperatures limits. The engine control system should also guarantee a safe operation and prevent the compressor and turbines from entering the stall and surge conditions. The control objective for a turbo-fan engine is provide the pilot desired thrust and for a turbo-shaft engine is to maintain a constant power turbine shaft speed. The set-point and limit controllers are integrated using a min-max control architecture. Mechanical wear due to normal use, component damage, engine-to-engine manufacturing variations and random, transient component failures can cause fluctuations in engine response and degrade the engine control system performance. Engine health parameters like component efficiencies, flow rates, etc. are used to represent both slow engine deterioration and sudden transient faults. These health parameters along with state variables such as shaft speeds, component temperatures and pressures are used to generate non-linear dynamic engine model. The non-linear engine model is linearized based on the flight condition defined by the operating speed, atmospheric conditions, engine health state and steady state shaft speeds or engine pressure ratio (determined by the thrust request or power lever angle). If VBV and VSV are scheduled based on engine parameters, then the engine control system is designed as a single-input control system with only fuel flow rate as the control input. The engine performance can be improved by designing the control system as a multiple-input multiple-output control system with VBV and VSV as the additional control inputs. Generally, a gain scheduled linear controller based on either frequency domain or time domain control design methodologies is used to achieve the desired performance over the entire operating envelope. The limit, transient and gain-scheduled
set-point controllers are integrated using a min-max architecture which guarantees desired performance within the operating environment. The turbo-fan non-linear model is linearized based on the operating flight condition (altitude and Mach number) while the turbo-shaft engine is linearized based on the gas generator shaft speed (expressed as a percentage of design speed). The generated linear engine models are used for design of both gain scheduled set-point and gain scheduled limit controllers.

The control objectives of the turbine engine control system can be summarized as follows:

1. The turbo-shaft engine control system must maintain a constant power turbine speed in presence of rotor load disturbance via collective pitch and ambient operating conditions.
2. The turbo-fan engine control system must provide the desired thrust level as indicated by the pilot via the thrust lever.
3. The engine control system must maintain an adequate stall margin and limit the maximum and minimum static compressor discharge pressure $P_{s3}$.
4. The turbine inlet temperature ($T_{45}$), shaft 1 speed ($N_f$ or $N_p$) and shaft 2 speed ($N_c$ or $N_g$) should not be exceeded beyond a tolerable limit.
5. In addition, the turbo-shaft engine control system should ensure also that the torque load is effectively shared between the on-board engines for multi-engine application.
4.2. Networked Control Systems

As discussed in the earlier chapter, the smart sensors, smart actuators and the engine controller will be connected via a common, serial communication network. The smart sensors will measure engine states, convert it into a network compatible digital data and transmit the digital data packets (messages) in a periodic and serial fashion to the controller. The controller will wait until it receives all the sensor data to start its control law processing and then transmit the appropriate control action to the actuators via the same digital communication network. The actuators receive the control action, perform the required signal processing and control the engine appropriately. The use of an engine area network will contribute to the engine dynamics through additional constraints like network-induced time delays, random data packet loss and limited bit-rate. The network-induced time delays and data packet loss will degrade the engine control system performance and may even destabilize the control system. The limited bit-rate constraint will affect the sampling rate of the control system sensors. Since the data transmission is based on a serial fashion, software or hardware based data transmission mechanisms guarantee periodic data transmission through the use of synchronized internal clocks. Due to manufacturing differences, all internal node clocks will eventually lose their synchronization. This clock drift, also known as network jitter, will cause the difference between sampling periods of all network nodes. Although the communication protocol may have an in-built mechanism to detect and correct the loss in clock synchronization, the worst-case analysis should include loss in clock synchronization and the designed controller should be made robust to loss in synchronization.
Distributed turbine engine control system can be modeled as hybrid control systems with continuous-time engine dynamics controlled by a discrete-time controller. Some of the several developed hybrid control system design methodologies are discussed below.

1. Sampled data systems with continuous controller design

In this approach, a continuous-time controller is designed for a continuous-time plant model, which is then discretized and implemented using a sampler and hold devices.

2. Deterministic augmented discrete-time approach

In this method, the continuous-time plant model is first discretized based on a sampling rate to obtain a discrete-time dynamic model. The sensor and controller are assumed to be time-driven, the actuator is event-driven and the time delays can be modeled as either constant or time-varying delays. The infinite dimensional time delayed system is converted into a finite dimensional discrete-time model by augmenting the original model with the previous input and output states. A discrete-time controller is designed for the augmented sampled-data plant model based on either the time delay control design theory or any of the current discrete-time control design methodologies.

3. Hybrid system approach

If the plant model cannot be conveniently described by continuous models and differential equations, a discrete event system (DES) models can be developed to control the hybrid system. These discrete event dynamical systems are modeled using finite automata and the controller is designed based on game theory or dynamic programming.
The stability analysis and controller verification techniques for such models are significantly different from the continuous-time or discrete-time analysis methodologies.

4. Sampling time scheduling approach

In this approach, the time delay is assumed to be constant, known and less than one sampling period. The stability of the control system is ensured by selecting a long enough sampling period such that the network-induced delays will not affect its performance. However, this approach cannot handle data loss due to packet dropouts and random network jitter.

Since time-delay systems have state spaces of infinite dimensions, it is necessary to convert it to a finite dimensional state space model for gain scheduling as well as for integrating the set-point and limit controllers using min-max methodology. Also, the designed control system must be robust to network jitter, data loss and random time delays more than sampling period. Hence, only the first two control design approaches are more feasible for distributed engine control system design. To ensure predictable control system behavior, both network and engine dynamics are preferred to be modeled using deterministic modeling and control design processes. Hence, the deterministic augmented discrete-time approach was selected for this application. A brief review of the different communication constraints is presented first followed by the stability and performance analysis and control design for the proposed distributed engine control system.
4.3. Communication Constraints

4.3.1. Bit-rate Constraints

The continuous-time engine model is converted into an approximate discrete-time model using discretization based on a designed sampling rate. The minimum sampling rate of the sensors will be bounded by the Nyquist–Shannon sampling theorem, which states that perfect reconstruction of an analog signal is possible when the sampling frequency is greater than twice the maximum frequency of the sampled continuous signal. The control system sampling rate cannot be less than the sampling rate of the sensors and hence the maximum control system sampling rate will be bounded by the sampling rate of the sensor with the lowest sampling rate.

Let $f_{s1}, f_{s2}, f_{s3} \ldots f_{sp}$ be the sampling rates of sensors 1, 2, ..., p and let $f_{sc}$ be the control system sampling rate. Hence,

$$f_{sc} > \min\{f_{s1}, f_{s2}, f_{s3}, \ldots f_{sp}\}$$

(4.1)

The maximum desired sample time ($h$) of the discrete-time control system is given by

$$h < \frac{1}{f_{sc}}$$

(4.2)

The performance of the discrete-time networked control system is dependent on the sampling time. As the sampling time decreases, the control system performance increases at the expense of increase in bandwidth utilization. The limited bit-rate constraints determine the minimum sampling time. If the sampling time decreases, the time delay between sensor data acquisition and control action implementation increases, which can...
severely degrade the performance or even destabilize the control system. The sampling
time for the distributed turbine engine is assumed to be 10 ms. As shown in chapter 3,
bandwidth utilization due to this sampling time was within the allowable limits. The
effect of this sampling time on the maximum allowable consecutive data dropouts will be
discussed later in this chapter.

4.3.2. Network-induced Time Delay

Time delay systems have been studied extensively in last few years. Time delay systems
(TDS) are dynamic systems whose behavior depends not only on the current system state
but also on the former states. Such systems arise in real-time control systems where loop
closures are performance through a data network, processes which have internal time
delays due to mass flow phenomena, biological systems, chemical process plants, etc.
This time delay, which can be either constant or time-varying, can significantly degrade
the performance of the system, or can even destabilize the system. Transmission of
continuous time signal over a network consists of data sampling, data encoding, data
transmission and data decoding. The total network induced time delay network access
delay and network transmission delay. Network access delay is the time taken for the
shared network to accept data. Network transmission delay can be either constant, time
varying or random depending on the network congestion and channel quality.
The design considerations for networked control systems are presented in [62] in which it
is shown that the sampling period to obtain the acceptable performance of networked
control systems is bounded by both lower and upper bounds. When the continuous-time
model is sampled with a constant sampling period, there is a maximum sampling time beyond which the closed loop control system will become unstable. Also, as the sampling period decreases, the number of data packets transmitted over the communication network increases which may result in control system performance degradation. The procedure to calculate the maximally allowable transmission interval (MATI) and the maximally allowable delay (MAD) that guarantee stability of the NCS in the presence of communication constraints is discussed in [63]. The discrete time control design methodology for networked control systems with time-varying sampling intervals, packet dropouts and time-varying bounded delays is discussed in [64] and [65]. Delay dependent stability conditions for time-varying bounded delays are presented in [66] and [67] for discrete-time systems while [68] presents stability conditions for sampled-data systems.

A linear time delay system is as shown below

\[
\dot{x}(t) = Ax(t) + A_d x(t - \tau(t)) \tag{4.3}
\]

\[x(t) = \phi(t), \forall t \in [-\tau(t), 0]\]

where, \(x(t) \in \mathbb{R}^n\) is the state vector and \(\phi(t)\) is the continuous initial condition.

Assumption 1: If the time delay is constant, \(\tau(t) = \tau\)

Assumption 2: Time-varying time delay, \(\tau(t)\) is a continuous function such that \(0 \leq \tau(t) \leq \tau^{max}\)

Assumption 3: Time-varying time delay, \(\tau(t)\) is a differentiable function such that
\[ 0 \leq \tau(t) \leq \tau^{max} \text{ and } \dot{\tau}(t) \leq \rho < 1 \]

Assumption 4: Time-varying time delay, \( \tau(t) \) is a differentiable function such that
\[ \tau^{min} \leq \tau(t) \leq \tau^{max} \text{ and } \rho^{min} \leq \dot{\tau}(t) \leq \rho^{max} \]

Time delay stability conditions can be briefly divided into two types-

1. Delay-independent stability conditions

Stability conditions which do not depend on the time delay information are known as delay independent stability conditions. Delay independent stability holds for all positive and finite value of delays. Hence, these conditions impart robustness to the system with respect to time delay.

**Theorem 1:**

The time-delay system with constant time delay is asymptotically stable if there exists matrices \( P > 0 \) and \( Q > 0 \) such that [69]

\[
\begin{bmatrix}
A^TP + PA + Q & PA_d \\
A_d^TP & -Q
\end{bmatrix} < 0
\]  \hspace{1cm} (4.4)

Proof: The above LMI condition is obtained from the following Lyapunov-Krasovskii function

\[ V(t) = x^T(t)Px(t) + \int_{t-\tau}^{t} x^T(s)Qx(s) ds \]

**Theorem 2:**

The time-delay system with time-varying delay under assumption 3 is asymptotically stable if there exists matrices \( P > 0 \) and \( Q > 0 \) such that [69]
\[
\begin{bmatrix}
A^T P + PA + Q & PA_d \\
A_d^T P & -(1 - \rho) Q
\end{bmatrix} < 0
\] (4.5)

Proof: The above LMI condition is obtained from the following Lyapunov-Krasovskii function

\[
V(t) = x^T(t)Px(t) + \int_{t-\tau(t)}^{t} x^T(s)Qx(s) \, ds
\]

2. Delay-dependent stability conditions

Delay dependent stability conditions are those which depend on the time delay value. In these conditions, stability is preserved only for some finite value of delay and it is unstable for all other values. Some of the approaches used to analyze the time-dependent stability of time delay systems are bounding techniques, descriptor system model and free-weighting matrix approach. Since all these conditions are based on a linear matrix inequality (LMI) approach, which is a convex optimization technique, only sufficient conditions can be obtained. The conservativeness of these sufficient conditions can be reduced by introducing additional matrix variables at an added expense of increase in computational expense. Delay independent conditions are less conservative than delay independent conditions, especially when the time delay is very small. Hence, delay dependent conditions are widely used for controller design. A delay dependent control design based on descriptor approach is presented below [67].
Theorem 3:

The time delay system is asymptotically stable for all differentiable delays \( \tau(t) \in [\tau^{\min}, \tau^{\max}] \) with \( \rho^{\min} \leq \dot{\tau}(t) \leq \rho^{\max} \) if there exists \( n \times n \) matrices

\[
Q, R_i = 0, 1, 2, S_0, S_{11}, S_{12}, S_{13}, D^k = 0, 1, 2, 3 > 0, W, \mathcal{P}_j, Y_j, Y_{\dot{J}}, T, Z_{1j}, Z_{2j} \text{ for } j = 1, 2
\]

such that \( Q \geq 0, R_i > 0, S_0 > 0, \left[ \begin{array}{cc} S_{11} & S_{12} \\ S_{21} & S_{22} \end{array} \right] > 0 \) and following LMI are feasible

\[
\begin{bmatrix}
\sum_{g=0}^{\infty} \Omega^{j}_{11} + \frac{\dot{\tau}(t)(P^1 - P^2)}{h_2 - h_1} & \Omega^{j}_{12} & R_0 - Y_{11}^T & Z_{1j}^T & (\tau(t) - h_1)Y_{11}^T & (h_2 - \tau(t))Z_{11}^T & \Omega^{j}_{17} & 0 \\
\Omega^{j}_{21} & -Y_{21}^T & Z_{2j}^T & (\tau(t) - h_1)Y_{21}^T & (h_2 - \tau(t))Z_{21}^T & \Omega^{j}_{18} & 0 \\
\phi_4^{(1)} & 0 & 0 & 0 & 0 & R_2 + W - S_{12} \\
\phi_5^{(1)} & 0 & 0 & (\tau(t) - h_1)R_i & 0 & 0 \\
\phi_6^{(1)} & 0 & (h_2 - \tau(t))W_i & 0 & 0 \\
-1 - \dot{\tau}(t)Q + T + T_i^T & -W_3 & -(S_0 + S_1 - Q) \\
-1 - \dot{\tau}(t)Q + T + T_i^T & -(S_1 + R_2) & -W_3 \\
\end{bmatrix}
\]

Where, for \( i, j = 1, 2 \) and \( h_1 = \tau^{\min}, h_3 = \tau^{\max}, h_1 = (\tau^{\min} + \tau^{\max})/2 \)

\[
\Omega^{j}_{11} = A^T + P_{2j} + P_{2j}^TA + S_0 + R_0
\]

\[
\Omega^{j}_{12} = \left[ \frac{\tau(t) - h_1}{h_2 - h_1} P_1 + \frac{h_2 - \tau(t)}{h_2 - h_1} P_2 \right] - P_{21}^T + A^T P_{31}
\]

\[
\Omega^{j}_{22} = -P_{3j} - P_{3j}^T + \sum_{i=0}^{2} (h_{i+1} - h_i)^2 R_i
\]

\[
\Omega^{j}_{17} = Y_{1j}^T - Z_{1j}^T + P_{2j}^TA_1
\]

\[
\Omega^{j}_{27} = Y_{2j}^T - Z_{2j}^T + P_{3j}^TA_1
\]

\[
\phi_3^{(1)} = -(S_0 + R_0 - S_{11} - Q)
\]

\[
\phi_4^{(1)} = -(S_{11} + R_2 - S_{13})
\]

87
\[ \phi_5^{(1)} = -(h_2 - h_1)(\tau(t) - h_1)R_1 \]
\[ \phi_6^{(1)} = -(h_2 - h_1)(h_2 - \tau(t))R_1 \]

The above stability condition is a sufficient condition and is very computationally expensive. It is not feasible to use the above theorem to analyze the stability of all linear models obtained from the non-linear engine model and a less computationally expensive stability conditions needs to be developed.

4.4. Control System Design

Assumption 1: All smart sensors have synchronized internal clocks and are clock driven. All sensors sample data at same global time and transmit the data in multiple, time-stamped data packets.
Assumption 2: The communication protocol can detect network jitters and initiates clock synchronization process when the jitter exceeds a predetermined limit, i.e. the nodes will sample the data with a random jitter bounded by an upper limit of the predetermined jitter limit.

Assumption 3: The controller node is assumed to be event-triggered, wherein, the control law processing is performed as soon as the node receives the data. The control input data is also time-stamped and transmitted to the smart actuator through multiple control packets.

Assumption 4: All smart actuator nodes are clock-driven and triggered after a predetermined clock time. It is also assumed that all actuator nodes transmit a data packet to other actuators and controller acknowledging successful data reception. All of the smart actuators receive these acknowledgement frames before they perform the control action.

As discussed earlier, the total time delay of the distributed control system can be decomposed into three main components: $\tau_{sc}$ which is the time delay between sensors and controller, $\tau_c$ which is the time delay due to the controller and $\tau_{ca}$ which is the time delay between controller and actuators. Based on assumption 1, $\tau_{sc}$ for each individual sensor node will depend on the sensor data transmission scheduling and can be determined by the controller node. Since the controller node is event triggered, it can start sensor data processing as soon it as receives each sensor data packet. However, the control law processing will occur only when it has received all the sensor data packets.
Time required for control law calculation, will depend on the computational resources available for the controller and time delay between controller and actuator will depend on the communication network scheduling and hence the randomness of both \( \tau_c \) and \( \tau_{ca} \) will make the networked control systems unpredictable. Assumption 4 will help to impart deterministic behavior by using clock driven actuators. The global clock time which will trigger the actuators can be static-scheduled or can be dynamically scheduled based on current network load. If the actuator node receives a corrupted control input data packet before the predetermined clock time, it may request the controller to retransmit the control input data packet. However, if the actuator does not receive the control input data packet before the predetermined clock time, it will continue to maintain its current position until the next sampling period. The main advantage of the hold input strategy is that the performance of the control system is dependent only on the total time delay \( \tau_k \) and not on the individual delays of \( \tau_{sc}, \tau_c \) or \( \tau_{ca} \).

### 4.4.1. Output Feedback NCS Design with Constant Time Delay

Consider the following continuous time linear state space model

\[
\dot{x}(t) = \hat{A}x(t) + \hat{B}u(t) \\
y(t) = \hat{C}x(t) \tag{4.8}
\]

\[x(0) = x_0\]

where, \( x(t) \in R^n \), \( y(t) \in R^q \) and \( u(t) \in R^p \) are the states, system output and system inputs respectively and \( x_0 \) denotes the initial condition. \( \hat{A}, \hat{B} \) and \( \hat{C} \) are matrices with appropriate dimensions.
The sampling period is assumed to be $h$ and the time delay, $\tau_k$, includes sensor-controller, processing delay and the controller-actuator time delay. Hence, based on assumption 4, time delay $\tau_k$, can be modelled as a constant time delay, $\tau$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{network_diagram.png}
\caption{Network diagram showing all time delays}
\end{figure}

The discrete control input is given as

$$u(t) = u_k \quad t \in [kh + \tau_k, (kh + 1) + \tau_{k+1}]$$

(4.9)
Hence, accounting for time delay $\tau_k$ and assuming zero hold over a sampling interval of $[kh, (k + 1)h]$, we get

$$u(t) = \begin{cases} 
  u(k - 1), & t \in [kh, kh + \tau_k] \\
  u(k), & t \in [kh + \tau_k, (k + 1)h]
\end{cases} \quad (4.10)$$

The discrete time control system can be written as follows

$$x(k + 1) = Ax(k) + B^0u(k) + B^1u(k - 1) \quad (4.11)$$

where,

$$A = e^{Ah}$$

$$B^0 = \int_0^{h-\tau_k} e^{\hat{A}t} \hat{B} \, dt$$

$$B^1 = \int_{h-\tau_k}^{h} e^{\hat{A}t} \hat{B} \, dt$$

The control input based on a static feedback gain is

$$u(k) = Ky(k) \quad (4.12)$$

Hence, the control input is given as

$$u(k) = KCx(k) \quad (4.13)$$

The closed loop system can be written as

$$x(k + 1) = Ax(k) + B^0KCx(k) + B^1KCx(k - 1) \quad (4.14)$$

The state-space discrete time delay control system model can be converted to discrete-time non-delay control system model by defining a new state variable $\bar{x}(k)$ such that
\[ \bar{x}(k) = \begin{bmatrix} x(k) \\ x(k-1) \end{bmatrix} \quad (4.15) \]

The augmented closed loop system can be rewritten as

\[ \bar{x}(k + 1) = \begin{bmatrix} A & 0 \\ I & 0 \end{bmatrix} + \begin{bmatrix} B^0 & 0 \\ 0 & 0 \end{bmatrix} K \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} + \begin{bmatrix} B^1 & 0 \\ 0 & 0 \end{bmatrix} K \begin{bmatrix} 0 & C \\ 0 & 0 \end{bmatrix} \bar{x}(k) \quad (4.16) \]

Hence,

\[ \bar{x}(k + 1) = \bar{A}\bar{x}(k) + \bar{B}^0 K\bar{C}_1 \bar{x}(k) + \bar{B}^1 K\bar{C}_2 \bar{x}(k) \quad (4.17) \]

where,

\[ \bar{A} = \begin{bmatrix} A & 0 \\ I & 0 \end{bmatrix} \]
\[ \bar{B}^0 = \begin{bmatrix} B^0 & 0 \\ 0 & 0 \end{bmatrix} \]
\[ \bar{B}^1 = \begin{bmatrix} B^1 & 0 \\ 0 & 0 \end{bmatrix} \]
\[ \bar{C}_1 = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \]
\[ \bar{C}_2 = \begin{bmatrix} 0 & C \\ 0 & 0 \end{bmatrix} \]

Lemma 1:

The discrete-time delay system is stable if and only if there exists a positive definite matrix \( P \in \mathcal{R}^{2n \times 2n} \) such that the following linear inequality holds

\[ (\bar{A} + \bar{B}^0 K\bar{C}_1 + \bar{B}^1 K\bar{C}_2)^T P (\bar{A} + \bar{B}^0 K\bar{C}_1 + \bar{B}^1 K\bar{C}_2) - P < 0 \quad (4.19) \]

Remark: The inequality is a necessary and sufficient condition for stability of discrete time system and no other stability condition can be better than this stability condition.
Lemma 2: The following statements are equivalent [70]

a) There exists a positive definite matrix $P$ such that

$$A^TPA - P < 0 \quad (4.20)$$

b) There exists a positive definite matrix $Q$ and a matrix $V$ such that

$$\begin{bmatrix} -Q & V^T A^T \\ AV & Q - V - V^T \end{bmatrix} < 0 \quad (4.21)$$

Theorem 4:

The discrete-time delay system (4.11), (4.12) is stable if there exists positive definite matrices $Q \in \mathcal{R}^{2n \times 2n}$ and matrices $S \in \mathcal{R}^{q \times q}$ $V = \begin{bmatrix} V_{11} & 0 \\ 0 & V_{11} \end{bmatrix}$ with $V_{11} \in \mathcal{R}^{n \times n}$, and $Y \in \mathcal{R}^{p \times q}$ such that both (4.22) and (4.23) holds

$$\begin{bmatrix} -Q & V^T A^T + \bar{c}_1^T Y^T \bar{B}_0^T + \bar{c}_2^T Y^T \bar{B}_1^T \\ AV + \bar{B}_0 Y \bar{c}_1 + \bar{B}_1 Y \bar{c}_2 & Q - V - V^T \end{bmatrix} < 0 \quad (4.22)$$

$$SC = CV_{11} \quad (4.23)$$

The closed loop system is stable with output feedback
\[ K = YS^{-1} \]  

(4.24)

Proof: According to the Lemma 1, the discrete-time delay system (4.11) and (4.12) is stable if and only if there exist a positive definite matrix \( P \) such that

\[
(\bar{A} + \bar{B}^0K\bar{C}_1 + \bar{B}^1K\bar{C}_2)^T P (\bar{A} + \bar{B}^0K\bar{C}_1 + \bar{B}^1K\bar{C}_2) - P < 0
\]

(4.25)

By the Lemma 2, inequality (4.19) is equivalent to the below inequality

\[
\begin{bmatrix}
Q & V^T (\bar{A} + \bar{B}^0K\bar{C}_1 + \bar{B}^1K\bar{C}_2)^T \\
(\bar{A} + \bar{B}^0K\bar{C}_1 + \bar{B}^1K\bar{C}_2)V & Q - V - V^T
\end{bmatrix} < 0
\]

(4.26)

\[
(\bar{A} + \bar{B}^0K\bar{C}_1 + \bar{B}^1K\bar{C}_2)V = \bar{A}V + \bar{B}^0KC[I \ 0]V + \bar{B}^1KC[0 \ I]V
\]

(4.27)

\[
= \bar{A}V + \bar{B}^0KC[1 \ 0]\begin{bmatrix}
V_{11} \\
0
\end{bmatrix} + \bar{B}^1KC[0 \ I]\begin{bmatrix}
V_{11} \\
0
\end{bmatrix}
\]

(4.28)

\[
= \bar{A}V + \bar{B}^0KCV_{11}[I \ 0] + \bar{B}^1KCV_{11}[0 \ I]
\]

(4.29)

Let,

\[ SC = CV_{11} \text{ and } K = YS^{-1} \]

Hence,

\[
(\bar{A} + \bar{B}^0K\bar{C}_1 + \bar{B}^1K\bar{C}_2)V = \bar{A}V + \bar{B}^0YC[I \ 0] + \bar{B}^1YC[0 \ I]
\]

(4.30)

The inequality (4.26) is now equivalent to the below inequality, which proves the theorem

\[ \sum g_{837} \sum g_{845} \]

\[ \sum g_{857} \sum g_{849} \]

\[ \sum g_{868} \sum g_{869} \]

\[ \sum g_{870} \sum g_{871} \]

\[ \sum g_{30} \times \hat{h} \]

\[ \sum g_{31} \times \hat{h} \]

95
\begin{equation}
\begin{bmatrix}
\bar{A}V + \bar{B}^0Y\bar{C}_1 + \bar{B}^1Y\bar{C}_2 \\
V^T\bar{A}^T + \bar{C}_1^T\bar{Y}^T\bar{B}^0 + \bar{C}_2^T\bar{Y}^T\bar{B}^1
\end{bmatrix} < 0
\end{equation}

4.4.2. Stability Under Random, Bounded Jitter

For constant time delay, \( \tau_k \), let the jitter margin be denoted by \( \delta^{max} = \frac{\tau^{max}_k - \tau^{min}_k}{2} \)

Hence,

\[ B^{00} = \int_0^{h-\tau^{max}_k} e^{A t} \hat{B} dt \]

\[ B^{11} = \int_{h-\tau^{min}_k}^h e^{A t} \hat{B} dt \]

\[ B^{\delta m} = \int_{h-\tau^{max}_k}^{h-\tau^{min}_k} e^{A t} \hat{B} dt \]

\[ \bar{B}^{\delta m} = \begin{bmatrix} B^{\delta m} \\ 0 \end{bmatrix} \]

**Theorem 5:**

The closed loop discrete-time system (4.17) is stable for jitter \( \delta \in [0, \delta^{max}] \) if the following condition is satisfied.

\[ \sigma_{max}(\bar{B}^{\delta m}K\bar{C}_1 + \bar{B}^{\delta m}K\bar{C}_2) \]

\[ < -\sigma_{max}(\bar{A} - \bar{B}K\bar{C}) + \left( [\sigma_{max}(\bar{A} - \bar{B}K\bar{C})]^2 + \frac{1}{\sigma_{max}(P)} \right)^{1/2} \]

(4.32)

Where, \( P \) is obtained from following discrete time Lyapunov equation.
\((\bar{A} + \bar{B} \bar{K} \bar{C})^T \bar{P}(\bar{A} + \bar{B} \bar{K} \bar{C}) + I = 0\) \hspace{1cm} (4.33)

Proof: The above theorem is proved using robust stability bounds for time-varying unstructured perturbations [71].

When \(\tau_k = \tau_k^{\text{max}}, B^1 = B^{11} + B^{\delta m}, B^0 = B^{00}\)

\(\ddot{x}(k + 1) = \bar{A} \ddot{x}(k) + \bar{B}^{00} K \bar{C}_1 \ddot{x}(k) + (\bar{B}^{11} K \bar{C}_2 \ddot{x}(k) + \bar{B}^{\delta m} K \bar{C}_2 \ddot{x}(k))\) \hspace{1cm} (4.34)

And when \(\tau_k = \tau_k^{\text{min}}, B^0 = B^{00} + B^{\delta m}, B^1 = B^{11}\)

\(\ddot{x}(k + 1) = \bar{A} \ddot{x}(k) + (\bar{B}^{00} K \bar{C}_1 \ddot{x}(k) + \bar{B}^{\delta m} K \bar{C}_1 \ddot{x}(k)) + \bar{B}^{11} K \bar{C}_2 \ddot{x}(k)\) \hspace{1cm} (4.35)

For \(\tau_k^{\text{min}} \leq \tau_k \leq \tau_k^{\text{max}},\)

\(\ddot{x}(k + 1) = \bar{A} \ddot{x}(k) + (\bar{B}^{00} K \bar{C}_1 \ddot{x}(k) + \bar{B}^{\delta 0} K \bar{C}_1 \ddot{x}(k)) + (\bar{B}^{11} K \bar{C}_2 + \bar{B}^{\delta 1} K \bar{C}_1) \ddot{x}(k)\) \hspace{1cm} (4.36)

Since \(\delta^{\text{max}} \ll h,\) for \(0 \leq \delta \leq \delta^{\text{max}},\) \(\sigma_{\text{max}}(B^{\delta 0}) \leq \sigma_{\text{max}}(B^{\delta m})\) and

\(\sigma_{\text{max}}(B^{\delta 1}) \leq \sigma_{\text{max}}(B^{\delta m})\) \hspace{1cm} (4.37)

Hence the jitter can be expressed as a perturbation in the closed loop nominal matrix as

\(\ddot{x}(k + 1) = (\bar{A} + \bar{B} K \bar{C}) \ddot{x}(k) + (\bar{B}^{\delta m} K \bar{C}_1 + \bar{B}^{\delta m} K \bar{C}_2) \ddot{x}(k)\) \hspace{1cm} (4.38)
4.5. Node Failures

Transient sensor or actuator node failures can result in partial or total communication loss. Communication loss or data loss due to packet dropouts can also occur due to data corruption, transmission errors or due to coding/encoding errors. The effect of communication packet dropouts on the stability and performance of distributed engine control systems is discussed in [72]. It is also shown that the packet dropping margin, which is a measure of stability robustness under packet dropouts, is largely dependent on the closed-loop controller structure. Data communication for a single-input single-output (SISO) networked control system is through only two data packets, one for sensor data and one for control input or actuator data. Loss in either of the two data packets can result in severe performance degradation or loss in stability. Data communication of a single-input multiple-output (SIMO) or a multiple-input multiple-output (MIMO) networked control system is through multiple data packets. Hence, single node failures will result in only partial sensor or actuator data loss, while data concentrator node failure may cause transient failure of multiple sensors leading to severe performance degradation. If the control input data dropout is not compensated, it may result in severe performance degradation of the control system or engine instability.

For model-based turbine engine control system, loss in sensor data can be compensated by using an observer or Kalman filter based on the on-board engine model. Estimation techniques for networked control systems are discussed in [73] while [74] presents a Kalman filter for discrete-time networked control systems with time delay. H-infinity filter for NCS with communication constraints like time delays, packet dropouts and both
delayed noise and non-delayed noise is studied in [75]. As with any fault-tolerant control system, design of an accurate filter for fault detection and isolation of networked control system is very crucial. As shown in Ref. [76], the sensitivity of the conventional observer based residual generator reduces in the presence of communication constraints and is not suitable for networked control systems. A parity space based fault detection scheme is presented in [77] to handle communication constraints while Ref. [78] proposed a new fault detection filter to detect faults present in NCSs with both input delay and measurement delay. A review of fault tolerant control design methodology is presented in [79], [80], [81] and [82]. The following compensation strategy can be implemented when either of the transient sensor node faults, transient actuator node fault, data transmission failure due to packet dropouts or data message loss due to corrupted data is detected.

1. Zero-input strategy in which control input is not applied, i.e., the actuator is moved to either a closed or a null position. If the fault is present in the sensor, the faulty sensor value is not used for control law calculation.

2. Hold-input strategy in which the same control input value or sensor value is held constant until the node recovers from the fault.

3. Reconfigurable strategy in which a new control input value or sensor value is calculated based on a fault-tolerant control design methodology.

Ref. [83] shows that both hold-input and zero-input strategies are equivalent and additional simulations have to be performed to evaluate which of these strategies are better suited for the failed scenario. Design of reconfigurable control systems to handle sensor and actuator faults is widely discussed in literature, for example, [84],
[85] to name a few. Ref. [86] and [87] propose a control design techniques based on old control input values to provide compensation for the lost control input data packets. Reconfigurable control design methodology for distributed engine control systems based on both single packet and multiple packet data transmission policies is proposed below.

4.5.1. Single-Input Single-Output (SISO) Control System

As discussed in earlier chapters, the turbine engine control system has three control inputs, of which, two (VBV and VSV) are scheduled based on pressure and temperature sensor values while the third input (W_f) is used for engine control. The fuel flow control input value is calculated only after the FADEC receives all sensor data and the value obtained using min-max architecture is transmitted to the fuel flow valve. Hence, \( \tau_{sc} = 2.37 \text{ ms} \) and \( \tau_{CA} = 0.474 \text{ ms} \). The computational delay is assumed to be less than 1 ms, i.e. \( \tau_c = 1 \text{ ms} \). The total time delay is \( \tau = 3.844 \text{ (ms)} < h \). If the sensor data or the control input data is not received successfully by the recipient node, a transmission attempt will be made by the transmitter. However, the maximum number of transmission attempts will have to be predetermined in order to maintain the deterministic nature of the communication protocol. If the data message is not transmitted within these maximum number of allowed retransmission attempts, the data message is ignored. The controller has to be robust against such packet dropouts. Both stochastic and deterministic modeling techniques can be used to model data loss due to packet dropouts.
In deterministic data loss modeling, packet dropouts are modeled as a time delay in multiples of the sampling time.

For notational simplicity, let $C_t$ indicate the true or active sensor and $C_f$ indicate the failed sensor due to either transient node failure or communication failure. Similarly, the true or active actuator is denoted by $B_t$ and failed actuator by $B_f$. The following four scenarios are possible for SISO system based on the active or failed nature of the nodes.

A. Both the sensor and actuator nodes are active $(C_t B_t)$

B. Sensor failure and active actuator node $(C_f B_t)$

C. Actuator failure and active sensor node $(C_t B_f)$

D. Both sensor and actuator failure $(C_f B_f)$

Let $\eta_s$ be the successive number of packet dropouts in sensor-controller channel and $\eta_a$ in the controller actuator channel.

Failure scenario $B$ $(C_f B_t)$ can be modeled as a time delay, $\tau_{sc}$ as shown

$$\tau_{sc} = \eta_s \ast h$$

(4.39)

and failure scenario $C$ $(C_t B_f)$ can be modeled as a time delay, $\tau_{ca}$

$$\tau_{ca} = \eta_a \ast h$$

(4.40)

Hence, data dropout in sensor-controller channel will result in a delay in measured system output.

$$y(k) = y(k - \eta_s)$$

(4.41)
The delay in control input due to data dropouts is,

\[ u(t) = u(k - \eta_a) \]  \hspace{1cm} (4.42)

Similarly, for failure scenario D \((G_f B_f)\), the closed loop discrete-time control system is given as,

\[ x(k + 1) = Ax(k) + BKx(k - \eta_s - \eta_a) \]  \hspace{1cm} (4.43)

Let \(\eta_{max}\) be the maximum number of successive packet dropouts that can be tolerated by the control system without a losing its stability.

Hence,

\[ \tau(t)^{max} = \eta_{max} \times h \]  \hspace{1cm} (4.44)

The discrete time control system with packet dropouts will have to be analyzed for stability under time-varying time delays.

The augmented closed loop control systems are given as

\[ \bar{x}(k + 1) = \bar{A}\bar{x}(k) + \bar{B}K\bar{C}_i\bar{x}(k) \]  \hspace{1cm} (4.45)
\[
\tilde{x}(k) = \begin{bmatrix}
x(k) \\
x(k - 1) \\
\vdots \\
x(k - \eta_l) \\
x(k - \eta^{\max})
\end{bmatrix} \quad \text{for } \eta_l = 1,2, \ldots \eta^{\max} \tag{4.46}
\]

where,
\[
\bar{A} = \begin{bmatrix}
A & 0_{n \times n(\eta^{\max}-1)} & 0_{n \times n} \\
I_{(n \times \eta^{\max}) \times (n \times \eta^{\max})} & 0_{n \times n}
\end{bmatrix}
\]
\[
\bar{B} = \begin{bmatrix}
B \\
0_{\eta^{\max} \times 1}
\end{bmatrix}
\]
\[
\tilde{C}_i = \begin{bmatrix}
0_{1 \times n} \\
C & 0_{1 \times (\eta^{\max}-i)n}
\end{bmatrix}, \quad i = 1,2, \ldots \eta^{\max}
\]

Corollary 1:

The discrete-time delay system is stable if there exists a positive definite matrix \( P \in R^{(\eta^{\max}+1)n \times (\eta^{\max}+1)n} \) such that the following linear inequalities hold for any \( i \in \{0,1, \ldots, \eta^{\max}\} \)
\[
(\bar{A} + \bar{B}K\tilde{C}_i)^T P (\bar{A} + \bar{B}K\tilde{C}_i) - P < 0 \tag{4.48}
\]

Based on the above corollary and Theorem 4, a stabilization condition which can stabilize the time-varying control system can be derived.

Corollary 2:

The discrete-time delay system (4.45) is stable if there exists positive definite matrices \( Q \in \mathcal{R}^{(\eta^{\max}+1)n \times (\eta^{\max}+1)n} \) and matrices \( S \in \mathcal{R}^{q \times q} \), \( V = \begin{bmatrix} V_{11} & 0 \\ V_{21} & V_{22} \end{bmatrix} \) with \( V_{11} \in \mathcal{R}^{q \times q} \).
\[ V_{21} \in \mathbb{R}^{((\eta^{\text{max}}+1)n-q) \times q}, \ V_{22} \in \mathbb{R}^{((\eta^{\text{max}}+1)n-q) \times ((\eta^{\text{max}}+1)n-q)} \quad \text{and} \ Y \in \mathbb{R}^{p \times q} \] such that the following linear inequality holds

\[
\begin{bmatrix}
-Q & V^T \hat{A}^T + \hat{C}_i^T Y^T \hat{B}^T \\
\hat{A}V + BY\hat{C}_1 & Q - V - V^T
\end{bmatrix} < 0
\] (4.49)

where,

\[ SC = CV_{11} \] (4.50)

The closed loop system is stable with output feedback

\[ K_\eta = YS^{-1} \] (4.51)

\[ u_\eta(t) = K_\eta y(k) \] (4.52)

Control input value calculated using both control gain \( K_\eta \) and \( K \) are both transmitted to the actuator at each sampling period. If the actuator senses a failure in communication between sensor and controller or between controller and actuator, it switches to the \( u_\eta(t) \).

This guarantees system stability for maximum successive packet dropouts for a time period of \( \eta^{\text{max}} \) times the sampling period and also reduces performance degradation in presence of data dropouts.
4.5.2. Single-Input Multiple-Output (SIMO) Control System

Single-input multiple-output (SIMO) control system is an extension of SISO control system in which multiple sensor measurements are used to calculate the control input value. A partially distributed engine control system with an integration of both set-point and limit controllers can be modeled as SIMO control system. If the control system architecture is based on a data concentrator which encodes all the sensor data as a single data packet, then the failure scenarios will be B-D as discussed above. However, if the sensor data is transmitted as multiple data packets, then in addition to the above discussed failure scenarios, there will be 2 more failure scenarios.

E. Active actuator and failure in at least one sensor \((C_i C_f B_f)\)

F. Actuator failure and failure in at least one sensor \((C_i C_f B_f)\)

Failure scenario \(E\)

The augmented control system is given as

\[
\ddot{x}(k + 1) = \tilde{A}_i \ddot{x}(k) + B^0 \tilde{C}^0 \dot{x}(k) + B^1 K \tilde{C}^1 \ddot{x}(k)
\]

(4.53)

\[
\ddot{x}(k) = \begin{bmatrix} x(k) \\ x(k - 1) \\ \vdots \\ x(k - \eta_i) \\ \vdots \\ x(k - \eta^{max}) \end{bmatrix} \quad \text{for} \ \eta_i = 1, 2, \ldots, \eta^{max}
\]

(4.54)

where,

\[
\tilde{A}_i = \begin{bmatrix} A & 0_{n \times (i-1)n} & BK \tilde{C}_f & 0_{n \times (\eta^{max-i})n} \\ 0_{n \eta^{max} \times n(\eta^{max}+1)} \end{bmatrix} \quad \text{for} \ i = 1, 2, \ldots, \eta^{max}
\]

(4.55)
\[
\tilde{B}^0 = \begin{bmatrix} B^0 \\ 0_{\eta^{\text{max}} \times p} \end{bmatrix}
\]

\[
\tilde{B}^1 = \begin{bmatrix} B^1 \\ 0_{\eta^{\text{max}} \times p} \end{bmatrix}
\]

\[
\tilde{C}^0 = [C_t \quad 0_{q \times \eta^{\text{max}} n}]
\]

\[
\tilde{C}^1 = [0_{q \times n} \quad C_t \quad 0_{q \times (\eta^{\text{max}} - 1) n}]
\]

\(C_t\) and \(C_f\) are the \(C\) matrix rows which represent the true and failed sensors respectively.

Theorem 4 can be used to calculate the reconfigurable control gain based on the (4.53) - (4.55) equations.

**Failure scenario \(F\)**

If the actuator fails first followed by sensor failure, then failure scenario \(F\) is equivalent to failure scenario \(C\). If the sensor node fails first followed by the actuator node, let the number of successive sampling periods between sensor and actuator node failure be denoted by \(\eta^{sa}\).

The augmented control system is given as

\[
\bar{x}(k + 1) = \bar{A}\bar{x}(k) + \bar{B}K\hat{C}_i\bar{x}(k)
\]  

(4.56)
\[
\dot{x}(k) = \begin{bmatrix}
  x(k) \\
  x(k-1) \\
  \vdots \\
  x(k-\eta_i) \\
  \vdots \\
  x(k-\eta_{\text{max}})
\end{bmatrix}
\text{for } \eta_i = 1, 2, \ldots, n_{\text{max}}
\] (4.57)

where,

\[
\tilde{A} = \begin{bmatrix}
  A & 0_{n \times n(\eta_{\text{max}}-1)} & 0_{n \times n} \\
  I_{(n \times \eta_{\text{max}}) \times (n \times \eta_{\text{max}})} & 0_{n \times n}
\end{bmatrix}
\] (4.58)

\[
\tilde{B} = \begin{bmatrix}
  B \\
  0_{\eta_{\text{max}} \times \eta_{\text{max}}}\times p
\end{bmatrix}
\]

\[
\tilde{C}_i = \begin{bmatrix}
  0_{q \times n} & C_t & 0_{q \times n \eta^{sa}} & C_f & 0_{q \times (\eta_{\text{max}}-n^{sa}-i-1) n}
\end{bmatrix}, i = 1, 2, \ldots, n_{\text{max}}
\] (4.59)

\[
C_f = \begin{bmatrix}
  0_{(j-1) \times n} \\
  C_j \\
  0_{(q-j) \times n}
\end{bmatrix}
\]

\[
C_t = C - C_f
\] (4.60)

where, \(C_j\) is the row vector corresponding to the failed sensor \(j\).

A reconfigurable control gain for failure scenario \(F\) can be calculated using theorem 4 and equations (4.56) – (4.60).

### 4.5.3. Multiple-Input Multiple-Output (MIMO) Control System

Control systems with more than one input and output are known as multiple input multiple output (MIMO) control systems. MIMO networked control systems can benefit from the availability of an additional actuator. Several control input values based on reconfigurable control gains can be compressed in one or more data packets and stored in...
the actuator. MIMO control systems will have 3 more failure scenarios in addition to failure scenarios B-F.

G. All active sensors and failure in at least one actuator ($C_t B_f B_t$)

H. Failure in all sensors and at least one actuator failure ($C_f B_f B_t$)

I. Failure in at least one sensor and one actuator ($C_t C_f B_f B_t$)

In addition to the above failure scenarios, timing of the node failure may cause additional failure scenarios. For example, for failure scenario I, sensor failure preceded by actuator failure scenario is different than actuator failure preceded by sensor failure scenario.

**Failure scenario $G$**

The augmented control system is given as

$$
\ddot{x}(k + 1) = \bar{A}_i \bar{x}(k) + \bar{B}_t^0 K \bar{C}_0 \ddot{x}(k) + \bar{B}_t^1 K \bar{C}_1 \ddot{x}(k)
$$

$$
\bar{x}(k) = \begin{bmatrix}
x(k)
x(k-1)
\vdots
x(k - \eta_i)
\vdots
x(k - \eta_{max})
\end{bmatrix}
$$

for $\eta_i = 1,2,...,\eta_{max}$

(4.61)

(4.62)

where,

$$
\bar{A}_i = \begin{bmatrix}
A & 0_{n \times (i-1)n} & B_f K C & 0_{n \times (\eta_{max}-i)n} \\
0_{n \eta_{max} \times n} & 0_{n \eta_{max} \times (\eta_{max}+1)}
\end{bmatrix}
$$

for $i = 1,2,...,\eta_{max}$

(4.63)

$$
\bar{B}_t^0 = \begin{bmatrix}
B_t^0 \\
0_{\eta_{max} \times p}
\end{bmatrix}
$$
\[ \tilde{B}_t^1 = \begin{bmatrix} B_t^0 \\ 0 \end{bmatrix} \]

\[ B_f = [0_{n \times (j-1)} B_j 0_{n \times (p-j)}] \]

where, \( B_j \) is the column vector corresponding to the failed actuator \( j \).

\[ B_t^0 = \begin{bmatrix} B_{1:j-1}^0 & 0_{n \times 1} & B_{j+1:p}^0 \end{bmatrix} \]

\[ B_t^1 = \begin{bmatrix} B_{1:j-1}^1 & 0_{n \times 1} & B_{j+1;p}^1 \end{bmatrix} \]

Where \( B_{i:k}^0 \) and \( B_{i:k}^1 \) denotes column vectors \( l \) to \( k \) of \( B^0 \) and \( B^1 \) respectively.

\[ \tilde{C}^0 = [C \quad 0_{q \times \eta^{\text{max}}_n}] \]

\[ \tilde{C}^1 = [0_{q \times n} \quad C \quad 0_{q \times (\eta^{\text{max}} - 1)n}] \]

A reconfigurable control gain for failure scenario \( G \) can be calculated using theorem 4 and equations (4.61) – (4.63).

**Failure scenario \( H \)**

If all of the sensors fails first followed by an actuator failure, then failure scenario \( F \) is equivalent to failure scenario \( B \). If the actuator node fails first followed by the sensor node, let the number of successive sampling periods between actuator and sensor node failure be denoted by \( \eta^{as} \).

The augmented control system is given as

\[ \tilde{x}(k + 1) = \bar{A}\tilde{x}(k) + \bar{B}_t K \tilde{C}_t^0 \tilde{x}(k) + \bar{B}_f K \tilde{C}_t^1 \tilde{x}(k) \quad (4.64) \]
\[
\bar{x}(k) = \begin{bmatrix} x(k) \\ x(k-1) \\ \vdots \\ x(k-\eta_i) \\ \vdots \\ x(k-\eta_{\text{max}}) \end{bmatrix}
\text{for } \eta_i = 1, 2, \ldots, \eta_{\text{max}}
\] (4.65)

where,

\[
\bar{A} = \begin{bmatrix} A & 0_{n \times n(\eta_{\text{max}}-1)} & 0_{n \times n} \\ I_{(n \times \eta_{\text{max}}) \times (n \times \eta_{\text{max}})} & 0_{n \times n} \end{bmatrix}
\] (4.66)

\[
\bar{C}_i^{0} = [0_{q \times \eta_{\text{max}}^a}, C_{q \times \eta_{\text{max}}-i}n], i = 1, 2, \ldots, \eta_{\text{max}}
\]

\[
\bar{C}_i^{1} = [0_{q \times \eta_{\text{max}}-\eta_{\text{max}}^a-i+1}n, C_{q \times \eta_{\text{max}}-\eta_{\text{max}}-i}n], i = 1, 2, \ldots, \eta_{\text{max}}
\]

\[
\bar{B}_f = \begin{bmatrix} 0_{n \times (j-1)} & B_{j} & 0_{n \times (p-j)} \\ 0_{\eta_{\text{max}} \times p} \end{bmatrix}
\]

where, \(B_j\) is the column vector corresponding to the failed actuator \(j\).

\[
\bar{B}_l = \begin{bmatrix} B_{1:j-1} & 0_{n \times 1} & B_{j+1:p} \\ 0_{\eta_{\text{max}} \times p} \end{bmatrix}
\]

Where \(B_{l:k}\) denotes column vectors \(l\) to \(k\) of \(B\).

A reconfigurable control gain for failure scenario \(H\) can be calculated using theorem 4 and equations (4.64) – (4.66).

**Failure scenario \(I\)**

If one or more actuators fail first followed by failure of one or more sensors, the augmented closed loop system is as shown below.
\[
\ddot{x}(k + 1) = \ddot{A}x(k) + \ddot{B}^0 K \ddot{C}^0 \ddot{x}(k) + \ddot{B}^1 K \ddot{C}^1 \ddot{x}(k) + \ddot{B}_f K \ddot{C}_f \ddot{y}_f(k)
\]

\[
\ddot{x}(k) = \begin{bmatrix} x(k) \\ x(k - 1) \\ \vdots \\ x(k - \eta_i) \\ \vdots \\ x(k - \eta_{max}) \end{bmatrix}
\text{for } \eta_i = 1, 2, \ldots, \eta_{max}
\]

where,

\[
\ddot{A} = \begin{bmatrix} A & 0_{n \times n(\eta_{max} - 1)} \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix}
\]

\[
\ddot{B}_f = \begin{bmatrix} 0_{n \times (j - 1)} & B_j & 0_{n \times (p - j)} \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix}
\]

where, \(B_j\) is the column vector corresponding to the failed actuator \(j\).

\[
\ddot{B}_t = \begin{bmatrix} B_{1:j-1} & 0_{n \times 1} & B_{j+1:p} \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix}
\]

\[
B^0_t = \begin{bmatrix} B^0_{1:j-1} & 0_{n \times 1} & B^0_{j+1:p} \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix}
\]

\[
B^1_t = \begin{bmatrix} B^1_{1:j-1} & 0_{n \times 1} & B^1_{j+1:p} \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix}
\]

Where \(B_{l:k}, B^0_{l:k}\) and \(B^1_{l:k}\) denotes column vectors \(l\) to \(k\) of \(B, B^0\) and \(B^1\) respectively.

\[
C_f = \begin{bmatrix} 0_{(j-1) \times n} \\ C_j \\ 0_{(q-j) \times n} \end{bmatrix}
\]

\[
C_t = C - C_f
\]

where, \(C_j\) is the row vector corresponding to the failed sensor \(j\).

\[
\ddot{C}^0 = \begin{bmatrix} C_t & 0_{q \times \eta_{max} n} \end{bmatrix}
\]

\[
\ddot{C}^1 = \begin{bmatrix} 0_{q \times n} & C_t & 0_{q \times (\eta_{max} - 1) n} \end{bmatrix}
\]
\[ \tilde{C}_i^2 = [0_{q \times n} \quad C_f \quad 0_{q \times (\eta^{max} - i)n}], \quad i = 1, 2, \ldots \eta^{max} \]

\[ \tilde{C}_i^3 = [0_{q \times (\eta^{as} + i + 1)n} \quad C \quad 0_{q \times (\eta^{max} - \eta^{as} - i - 1)n}], \quad i = 1, 2, \ldots \eta^{max} \]

If one or more sensors fail first followed by failure of one or more actuators, the augmented closed loop system is as shown below.

\[ \tilde{x}(k + 1) = \tilde{A}_i \tilde{x}(k) + \tilde{B}_i^0 \tilde{K} \tilde{C}_i^0 \tilde{x}(k) + \tilde{B}_i^1 \tilde{K} \tilde{C}_i^1 \tilde{x}(k) + \tilde{B}_f \tilde{K} \tilde{C}_i^2 \tilde{x}(k) + \tilde{B} \tilde{K} \tilde{C}_i^3 \tilde{x}(k) \]

(4.70)

\[ \tilde{x}(k) = \begin{bmatrix} x(k) \\ x(k-1) \\ \vdots \\ x(k - \eta_i) \\ \vdots \\ x(k - \eta_{max}) \end{bmatrix} \text{ for } \eta_i = 1, 2, \ldots \eta^{max} \]

(4.71)

where,

\[ \tilde{A} = \begin{bmatrix} A & 0_{n \times n(\eta^{max} - 1)} & 0_{n \times n} \\ I_{(n \times \eta^{max}) \times (n \times \eta^{max})} & 0_{n \times n} \end{bmatrix} \]

(4.72)

\[ \tilde{B} = \begin{bmatrix} B \\ 0_{\eta^{max} \times p} \end{bmatrix} \]

\[ \tilde{B}_f = \begin{bmatrix} 0_{n \times (j-1)} & B_j & 0_{n \times (p-j)} \\ 0_{\eta^{max} \times p} \end{bmatrix} \]

where, \( B_j \) is the column vector corresponding to the failed actuator \( j \).

\[ \tilde{B}_t = \begin{bmatrix} B_{1:j-1} & 0_{n \times 1} & B_{j+1:j} \\ 0_{n \times \eta^{max} \times p} \end{bmatrix} \]

\[ \tilde{B}_t^0 = \begin{bmatrix} B_{1:j-1}^0 & 0_{n \times 1} & B_{j+1:j}^0 \end{bmatrix} \]

\[ \tilde{B}_t^1 = \begin{bmatrix} B_{1:j-1}^1 & 0_{n \times 1} & B_{j+1:j}^1 \end{bmatrix} \]
Where $B_{l:k}$, $B^0_{l:k}$ and $B^1_{l:k}$ denotes column vectors $l$ to $k$ of $B$, $B^0$ and $B^1$ respectively.

$$C_f = \begin{bmatrix} 0_{(j-1)\times n} \\ C_j \\ 0_{(q-j)\times n} \end{bmatrix}$$

$$C_t = C - C_f$$

where, $C_j$ is the row vector corresponding to the failed sensor $j$.

$$\bar{C}^0 = [C_t \quad 0_{q\times n}]$$

$$\bar{C}^1 = [0_{q\times n} \quad C_t \quad 0_{q\times \eta^{max-1}n}]$$

$$\bar{C}^2_i = [0_{q\times n} \quad C_t \quad 0_{q\times (\eta^{max-i}n)}, i = 1,2,\ldots \eta^{max}]$$

$$\bar{C}^3_i = [0_{q\times (\eta^{as+i+1}n)} \quad C_f \quad 0_{q\times (\eta^{max-as-i-1}n)}, i = 1,2,\ldots \eta^{max}]$$

A reconfigurable control gain for failure scenario $l$ can be calculated using theorem 4 and equations (4.67) – (4.69) or equations (4.70) – (4.72) depending on the failure sequence.

### 4.6. Simulation:

A non-proprietary T700 linear engine model is used for simulation of the proposed control methodology. The control design objective is to maintain a constant free turbine speed in presence of disturbance. Since, power extraction or engine inlet distortion can result in gas generator speed disturbance, a unit step disturbance in $Ng$ is assumed. For simplicity, it is also assumed that the limit controllers are inactive. The sensor and controller data transmission is scheduled based on time instants and hence each sensor and control input data packet will have its own pre-determined constant time delay.
4.6.1. SISO T700 Engine Simulation

For the below single input model, the T700 engine is operating at 95% Ng and is coupled with Apache airframe [60]. The continuous time engine model is as shown below.

\[
x(t)^T = [N_g \ N_p \ Q_{mr} \ N_{mr}]
\]

\[
u(t) = [w_f]
\]

\[
y(t) = [N_p]
\]

\[
\hat{A} = \begin{bmatrix}
-4.4650 & -0.1540 & 0 & 0 \\
2.6063 & -0.2482 & -31.0270 & 0 \\
0.0550 & 5.2088 & -0.8777 & -5.2001 \\
0 & 0 & 10.5890 & -0.6569
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
27.7136 \\
21.7189 \\
0.4581 \\
0
\end{bmatrix}
\]

\[
\hat{C} = [0 \ 1 \ 0 \ 0]
\]

\[
x_0 = [1 \ 0 \ 0 \ 0]^T
\]

The continuous time engine was discretized with sampling rate of 0.01 second and it was assumed that the upper bound on time delay is 0.004 second. SeDuMi, a MATLAB toolbox for solving semidefinite programming problems was used to solve the linear matrix inequalities (LMI) and to obtain the control gain [88]. The control gain which stabilizes the time-delay discrete model is

\[
K = -0.6139
\]

Figure 4.2 shows the Simulink model which was developed to simulate the performance of the above single-input control system with sensor and actuator first order dynamics.
Figure 4.2: Simulink model for SISO T700 engine

The output response with network-induced delay and jitter but without node failures is shown in Figure 4.3. This behavior is expected as the control gain was designed to stabilize the control system with network-induced time delay. From theorem 5, the SISO model is stable for jitter margin, $\delta^{max} = 50 \mu sec$. As discussed in Ref. [89] and [90] the jitter due to bit stuffing for CAN is bounded and its maximum size at the highest data rate (1 Mbps) is $24 \mu sec$. Since $\delta^{max}$ is greater than the worst case jitter, the proposed control design based on a distributed architecture is stable for all jitter values introduced either due to the communication protocol or loss in clock synchronization. The system output when $\eta^{max} = 10$ is as shown in Figure 4.4. The performance degradation is seen through an increase in both settling time and maximum overshoot. The occurrence of sensor and actuator faults is also shown; value 1 indicating presence of faults. The stability test fails for $K = -0.6139$ and $\eta^{max} > 7$ and hence, the systems becomes
unstable in presence of either sensor or actuator faults but gets stabilized again when nodes have recovered from their failed states.

A new control gain, $K_{new} = -0.0049$ is designed for $\eta_{max} = 7$ using corollary 2 and the control action is calculated for both the old and new control gains. Since one control input data packet can have 4 data values of 2 bytes each, both the control inputs can be transmitted to the actuator without increasing the bandwidth utilization. If the actuator does not receive the control input value at any time instant and self-detects its failure, the control action corresponding to the new control gain is applied. As shown in Figure 4.5, this minimizes the performance degradation.

Figure 4.6 shows the importance of scheduling the data transmission. If the node scheduling is not implemented properly, the controller and actuator node will receive out of sequence data and hence the system becomes unstable.

Figure 4.3: Output response with network dynamics
Figure 4.4: Output response when $\eta^{max} = 10$

Figure 4.5: Output response with $K_{new}$
4.6.2. SIMO T700 Engine Simulation

The same T700 engine model with additional sensor inputs is used to demonstrate the single input, multiple input scenarios. The additional sensors are \( N_p \), \( Q_{mr} \) and \( N_{mr} \) sensors along with \( N_g \) sensor. The Simulink model for SIMO engine control system is shown in Figure 4.7.
Since all the states of the T700 engine are available as feedback measurements, the optimum control law can be calculated using linear quadratic regulator (LQR) control design. The output response of the T700 SIMO control system with LQR control gain, network-induced time delay and jitter but without any network failures is shown in Figure 4.8.

\[ K_{lqr} = [-0.5025 \ -0.8011 \ -0.7354 \ -0.1315] \]

If the measured sensor data from all four sensors is transmitted through a data concentrator by encoding it into a single packet, communication failure will result in loss of all sensor data. Network failures for SIMO control system with single data packet transmission are represented by failure scenarios B-D. As seen in Figure 4.9, the distributed engine control system based on single packet transmission is unstable with LQR control gain in presence of failures. A reconfigurable control gain, \( K_D \), is designed.
for failure scenario D and a reconfigurable control gain, $K_E$, is calculated for failure scenario E assuming failure in sensor 2, i.e., $N_p$ sensor for both the scenarios.

$$K_D = [-0.0188 \quad -0.0515 \quad 0.0118 \quad -0.0162]$$

$$K_E = [-1.2358 \quad -0.8011 \quad 0.0697 \quad 0.0212]$$

Figure 4.9: Output response with single data packet transmission

The output response for the SIMO control system is as shown in Figure 4.10 - Figure 4.14. The four node failure strategies are

1. **Hold-output of faulty sensor with hold-input of faulty actuator (HO, HI)**
   In this strategy, only the faulty sensor data output and the faulty actuator control input is held constant until the node recovers from failure.

2. **Reconfigurable output of faulty sensor with hold-input of faulty actuator (RO,HI)**
   In this strategy, a set of control gains are calculated based on failure scenario D for single or multiple failures of each sensor nodes. For example, whenever the
data packet from $N_p$ sensor node experiences a communication failure or if $N_p$ sensor node itself experiences a transient failure, the controller calculates the control input using gain $K_D$ for active sensors and gain $K_{lqr}$ for failed sensors. Control input failure compensation is provided using hold-input strategy.

3. Hold-output of faulty sensor with reconfigurable input of faulty actuator (HO, RI)

This strategy is an extension of the first node failure compensation scheme in which along with the control input based on $K_{lqr}$ gain, a control input based on $K_E$ gain is encoded within the same data packet. If the actuator node detects loss in control input data packet, it uses the control input value based on $K_E$ gain stored in its memory.

4. Reconfigurable output of faulty sensor with reconfigurable input of faulty actuator (RO, RI)

In this strategy, reconfigurable control gains are used for providing compensation for both sensor and actuator failures.

The performance degradation due to the occurrence of node failures, measured by an increase in overshoot, rise time and settling time, can be validated by comparing Figure 4.10 - Figure 4.13 with Figure 4.8. The worst or maximum performance degradation is obtained for the first scheme based on a hold-output and hold-input strategy. Use of a reconfigurable control gain for compensating sensor node failures reduces performance degradation; however, the best failure compensation or minimum performance degradation is obtained by using either scheme 3 or 4 in which, a reconfigurable control
gain is used to provide actuator failure compensation. Figure 4.15 demonstrates performance comparison between $K_{lqr}$, $K_D$ and $K_E$. The best time-domain performance is obtained with $K_{lqr}$ which is less robust to node failures. Although $K_D$ and $K_E$ are more robust to sensor and actuator node failures respectively, they give a sub-optimum performance which is evident from an increase in settling time and overshoot. Hence, an optimum performance is obtained by implementing a control design strategy based on gain switching in which the reconfigurable control gain is used only during the presence of node failures.

Figure 4.10: $N_g$ sensor output response
Figure 4.11: $N_p$ sensor output response

Figure 4.12: $Q_{mr}$ sensor output response
Figure 4.13: $N_{mr}$ sensor output response

Figure 4.14: $w_f$ control input
4.6.3. MIMO T700 Engine Simulation

The continuous engine model of a multiple input, multiple output (MIMO) T700 engine is as shown below [58].

\[
x(t)^T = [N_g \ N_p \ Q_{mr}N_{mr}Q_{tr}N_{tr}T_{4.5}V_g]
\]

\[
u(t)^T = [w_f \ V_{gc}]
\]

\[
y(t)^T = [N_g \ N_p \ T_{4.5}]
\]

\[h=0.01sec \ \tau_k = 0.004 \ sec\]
The continuous time engine was discretized with sampling rate of 0.01 second and it was assumed that the upper bound on time delay is 0.004 second. The control gain calculated using to stabilize the time-delayed engine model is

\[
\begin{bmatrix}
-3.64 & 0 & 0 & 0 & 0 & 0 & 0 & 16.39 \\
7.84 & -1.09 & -1.212 & 0 & -0.15 & 0 & 0 & -34.3 \\
56.7 & 259 & -10 & -260 & 0 & 0 & 0 & -24.8 \\
0 & 0 & 0.1812 & -0.45 & 0 & 0 & 0 & 0 \\
0 & 2516 & 0 & 0 & 0 & -2516 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.465 & -0.371 & 0 & 0 \\
-5.9 & 0 & 0 & 0 & 0 & 0 & -2.2 & 24.7 \\
0 & 0 & 0 & 0 & 0 & 0 & -10 & 0
\end{bmatrix}
\]

\[
\hat{A} =
\begin{bmatrix}
1.01 & 0 \\
2.14 & 0 \\
5.35 & 0 \\
0 & 0 \\
0 & 0 \\
3.74 & 0 \\
0 & 10
\end{bmatrix}
\hat{B} =
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\]

\[
x_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T
\]

A Simulink model developed for the above given linear engine model of T700 turbo-shaft engine is shown in Figure 4.16. It is assumed that if a data loss occurs due to node failure, the node will require at least 1 second to recover from the failure. As seen in the Figure 4.17, the control gain \( K \), stabilizes the control system. It was also observed that the implementation of time-triggered actuators made the performance of the control system to be dependent only on the pre-determined \( \tau_k \) and not on the individual sensor-
controller time delay. This allows the user to schedule the sensor data transmission based on the control law calculation requirements and not on the control system performance.

Figure 4.16: Simulink model for MIMO T700 engine

Figure 4.17: Output response with gain $K$
The Figure 4.18 shows the sensor outputs in presence of both sensor and actuator faults. Sensor or actuator fault value 1 indicates presence of fault while 0 indicates successful data transmission. In presence of fault, the data packet is dropped and the controller will assume that there was no change in the sensed variable, i.e., it will use the hold-output scheme. Figure 4.19 shows the output response when there is an increase in sensor faults. The degradation in performance is due to an increase in settling time and overshoot in $T_{4.5}$. Figure 4.20 shows the output response with an increase in actuator failures. If the loss in performance is not acceptable, the active actuator can be used to provide compensation for degradation by using a reconfigurable control gain.

![Output response with transmission faults](image.png)
Figure 4.19: Output response with increase in sensor faults

Figure 4.21 shows the output response for a reconfigurable control where instead of hold-input strategy, the control input of active actuator switches to a new control input value. This new control input value is transmitted at each control instant and stored in the actuator memory. When there is failure in any one of the actuators, the failed actuator transmit an acknowledgement error which reports the failure to the controller and active actuator and the active actuator switches to the new control input value based on the reconfigurable control gain. Figure 4.22 shows the degradation in performance when the node recovery time increases to 2 seconds from 1 second.
Figure 4.20: Output response with increase in actuator faults

Figure 4.21: Output response with a reconfigurable control gain
Figure 4.22: Output response with node recovery time of 2 seconds

The SISO, SIMO and MIMO T700 engine control system simulations validate the benefits of the proposed control methodology. With a minor increase in bandwidth utilization, an active reconfigurable controller can be used to compensate for actuator node failures or communication failures between controller and actuator.
CHAPTER 5

CONCLUSIONS

Gas turbine engines are one of the most efficient internal combustion engines offering very high power to weight ratio, high operational reliability and low emissions. They are widely used in military and commercial aircrafts, helicopters, and power generators and to power vehicles on land and sea. Although the current turbine engines are very reliable and have high performance, efforts are made to further improve the performance while reducing the emissions, life cost and to improve the fuel efficiency. Active control systems like active combustion control, active clearance control, etc. require smart sensors and smart actuators with high bandwidth. This has driven the need to redesign the engine control system architecture. The future aircraft engine control systems will be based on a distributed architecture, wherein, all legacy sensors and actuators will be replaced by smart sensors and smart actuators connected via a communication data bus. The distributed engine control system will be subjected to various network faults like time delays and data loss, which may introduce unpredictability in the deterministic safety-critical control system design and may degrade the control system performance. Since, the new advanced control techniques will be designed to operate the engine near
its operational limits, performance degradation due to networked-induced time delay and data loss can severely limit the engine life and increase the life-cycle cost.

This dissertation presents a brief overview of the current turbine engine control system architecture and discusses the several candidate communication protocols. Requirements for the selection of the engine area network are presented. Three different control system architectures are proposed and discussed on the merit of their ease of implementation and their effect on the control system design objective. A T700 turbo-shaft engine is used to demonstrate the design methodology and to estimate the network-induced time delays, data loss due to packet dropouts and network jitter based on ARINC 825 protocol. A necessary and sufficient stability condition is presented for augmented discrete time systems with constant network-induced time delay. A control design methodology is presented to design a controller robust to a predetermined time delay. A sufficient stability condition is presented to ensure stability against bounded network induced jitter. A controller is designed based on the proposed stability condition and it is shown that the proposed controller is robust to both random, bounded time delay. Several failure scenarios are discussed for SISO, SIMO and MIMO control designs. State space models are presented for each of the failure scenarios to aid in the reconfigurable control design. A T700 linear engine model available in public literature is used to verify the proposed methodology. It is observed that the sensor data packet scheduling does not affect the engine control system performance when the time delays are random but bounded. This will allow the use of static communication network message scheduling for the
distributed engine control systems. However, the control system performance severely degrades if the controller or actuator receives out-of-order data packets. The proposed control design methodology is tolerant to random and bounded network-induced time delays, random and bounded jitter, intermittent data loss due to communication failure and transient node failures. It is also shown that the performance of a reconfigurable control design is better than both the performance of hold-input and zero-input failure compensation strategies in presence of network and/or node failures.

The future research recommendations include development of gain scheduled controllers for networked control systems with random, bounded time delay and bounded data dropouts. It is also recommended that the min-max architecture for the turbine engine control systems be replaced by model predictive control design based on networked control system model. It is also of interest to study the integration of flight and propulsion control systems under distributed control architecture. The future turbine engine control system with active control sub-systems like active turbine tip clearance, active stall margin control, etc. based on the distributed communication architecture can be modeled as a decentralized networked control systems. It is highly recommended to study the effect of time delays and data loss on the performance of these decentralized networked control systems.
BIBLIOGRAPHY


[10] F.-L. Lian, J. Moyne, and D. Tilbury, “Modelling and optimal controller design of


“825-2 general standardization of CAN (Controller Area Network) bus protocol for airborne use,” ARINC 825-2, 2011.


