CRANFIELD UNIVERSITY

MOHAMMAD RADAEI

Methodology for Avionics Integration Optimisation

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
PhD

PhD in Aerospace
Academic Year: 2020 – 2021

Supervisor: Dr Huamin Jia
Associate Supervisor: Dr Craig Lawson
October 2020

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Aerospace Engineering

PhD in Aerospace

Academic Year 2020 - 2021

Mohammad Radaei

Methodology for Avionics Integration Optimisation

Supervisor: Dr Huamin Jia
Associate Supervisor: Dr Craig Lawson
October 2020

This thesis is submitted in partial fulfilment of the requirements for the degree of PhD
(NB. This section can be removed if the award of the degree is based solely on examination of the thesis)

ABSTRACT

Every state-of-art aircraft has a complex distributed systems of avionics Line Replaceable Units/Modules (LRUs/LRMs), networked by several data buses. These LRUs are becoming more complex because of the increasing number of new avionics functions need to be integrated in an avionics LRU. The evolution of avionics data buses and architectures have moved from distributed analogue and federated architecture to digital Integrated Modular Avionics (IMA). IMA architecture allows suppliers to develop their own LRUs/LRMs capable of specific features that can then be offered to Original Equipment Manufacturers (OEMs) as Commercial-Off-The-Shelf (COTS) products. In the meantime, the aerospace industry has been investigating new solutions to develop smaller, lighter and more capable avionics LRUs to be integrated into avionics architecture.

Moreover, the complexity of the overall avionics architecture and its impact on cable length, weight, power consumption, reliability and maintainability of avionics systems encouraged manufacturers to incorporate efficient avionics architectures in their aircraft design process. However, manual design cannot concurrently fulfil the complexity and interconnectivity of system requirements and optimality. Thus, developing computer-aided design (CAD), Model Based System Engineering (MBSE) tools and mathematical modelling for optimisation of IMA architecture has become an active research area in avionics systems integration.

In this thesis, a general method and tool are developed for optimisation of avionics architecture and improving its operational capability. The tool has three main parts including a database of avionics LRUs, mathematical modelling of the architectures and optimisation algorithms. The developed avionics database includes avionics LRUs with their technical specifications and operational capabilities for each avionics function. A MCDM method, SAW, is used to quantify and rank each avionics LRU's operational capability. Based on the existing avionics LRUs in the database and aircraft level avionics requirements two avionics architectures are proposed i.e. AFCS architecture (SSA) and avionics architecture (LSA). The proposed avionics architectures are then modelled using mathematical programming. Further, the allocation of avionics LRUs to avionics architecture and mapping the avionics LRUs to their installation locations are defined as an assignment problem in Integer Programming (IP) format. The defined avionics architecture optimisation problem is to optimise avionics architecture in terms of mass, volume, power consumption, MTBF and operational capability. The problems are solved as both single-objective and multi-objective optimisation using the branch-and-bound algorithm, weighted sum method and Particle Swarm Optimisation (PSO) algorithm. Finally, the tool provides a semi-automatic optimisation of avionics architecture. This helps avionics system architects to investigate and evaluate various architectures in the early stage of design from an LRU perspective. It can also be used to upgrade a legacy avionics architecture.

Keywords:

Avionics Architecture, Optimisation, LRU, PSO, Integer Programming, MCDM, SAW, Database

ACKNOWLEDGEMENTS

I would like to thank my primary supervisor Dr Huamin Jia who provided me this opportunity to study at Cranfield University and got me involved in two EU-funded projects (FUCAM and GUASS) from which I gained a broad experience in systems engineering, avionics systems integration and design as well as aircraft systems designs. He supported me both scientifically and financially to make this research happen. I have benefited from his supervision in many aspects including the method and attitude of scientific research as well as hard-working. This would certainly be with me for the rest of my scientific career.

I would also like to thank my secondary supervisor Dr Craig Lawson for his great advice and recommendations to push me for being creative and productive. I further extend my gratitude to Allan Seabridge from whom I learned a lot in avionics data networking, hardware integration and testing course at Cranfield University. Moreover, I do thanks Dr James Whidborne and Dr Kim Blackburn for their great advice during my PhD progress sessions. I would also express my high level of appreciation to every any individuals from Cranfield University staff who inspire students throughout their journeys to a high level graduations.

Last but not least, I would like to thank my family and friends for their endless love and support.

To My Late Father

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimisation
ADC	Air Data Computer
ADCN	Aircraft Data Communication Network
ADS-B	Automatic Dependent Surveillance- Broadcast
AFCS	Automatic Flight Control System
AFDX	Avionics Full Duplex Switched Ethernet
AHRS	Attitude and Heading Reference System
AIOSS	Avionics Integration Optimisation Software System
ATM	Air Traffic Management
BP	Binary Programming
CAD	Computer Aided Design
CAN	Controller Area Network
CNS	Communication Navigation Surveillance
COTS	Commercial Off The Shelf
CPM	Core Processing Module
DIMA	Distributed Integrated Modular Avionics
ECU	Electronic Control Unit
EVS	Enhanced Vision System
FDR	Flight Data Recorder
FMS	Flight Management System
FUCAM	Future Cabin Design for Asian Market
GA	Genetic Algorithm
GAMS	General Algebraic Modelling System
GPS	Global Position System
HUD	Head Up Display
IAC	Integrated Avionics Cabinet
IMA	Integrated Modular Avionics
IP	Integer Programming
LRM	Line Replaceable Module
LRU	Line Replaceable Unit
LSA	Large Scale Architecture
MBSE	Model Based System Engineering
MCDM	Multi Criteria Decision Making
MFD	Multi-Function Display
MIP	Mixed Integer Programming
MTBF	Mean Time Between Failure
OC	Operational Capability

OCA	Operational Capability Assessment
OEM	Original Equipment Manufacturer
PFD	Primary Flight Display
PSO	Particle Swarm Optimisation
RA	Radar (Radio) Altimeter
RNP	Required Navigation Performance
RNAV	Area Navigation
RTA	Required Time of Arrival
RTOS	Real-Time Operating System
SA	Simulated Algorithm
SATCOM	Satellite Communication
SAW	Simple Additive Weighting
SWaP	Size, Weight and Power
SoC	System on a Chip
SoM	System on a Module
SSA	Small Scale Architecture
TCAS	Traffic Collision Avoidance System
TMDA	Time Division Multiplex Access
TRL	Technology Readiness Level
TTP	Time Triggered Protocol
WSM	Weighted Sum Method
WXR	Weather Radar
XML	Extensible Mark-up Language

1 Introduction

1.1 Avionics System Architecting and Architecture

Avionics system is one of the main aircraft systems that has interactions with many other aircraft systems and/or subsystems, and the performance and safety of which strongly depend on the technologies used in avionics systems. The functions known as avionics systems include navigation, communication, auto-flight control system (AFCS), Displays and controls, flight management system as well as situational awareness and surveillance. Meanwhile, the advancement of electronics and particularly digital computing brought about a rapid growth of the number of functions need to be integrated [1]. This demand for new and extended multi-functionality raised the system architecture and integration complexity in many cases [2]. The traditional trial-and-error system architecting is no longer efficient. This made system architect engineers to think up new ways and/or tools to handle this complexity.

In general, systems architecting is to create and build systems. It tries to find trade-off, balance, and compromise among the customers' needs and existing resources and technologies as well as technical and/or operational requirements [3], [4], [5]. Avionics system architecting, in particular, is an important and challenging task that can help engineers to visualise concepts by enabling requirements to be mapped, decomposing functions and determining functional links to physical mapping of the functions into an aircraft structure. In other words, the system architecture is a means of visualising concepts, in the early design stage, to discuss things and agree upon interfaces, functional integration and allocation, the usage of COTS components and standards independently of physical implementation [6].

System integration, on the other hands, is the design process in which decisions are made to integrate sub-systems and functions into a total system architecture irrespective of how the system will be fabricated. Finally, avionics architecture is referred to a general arrangement of systems, sub-systems, equipment that together perform a set of functions defined as avionics architecture considering top-level requirements, functional allocation, logical connections and physical interfaces. In the meantime, modern avionics architectures are highly interconnected computing modules. Investigating for new avionics system architecture would then lead to several feasible architectures in many cases. Therefore, it is necessary to assess the various possible architectures at aircraft level by means of quantitative criteria. This then proves the necessity of developing tools based on mathematical programing to help decision-makers in the preliminary design stage. In this research, a decision-analysis tool for identification of different architecture trade-offs in terms of some physical criteria including mass, power consumption, volume as well as reliability (MTBF) and operational capability has been developed.

1.2 Evolution of Avionics Architecture

1.2.1 Federated Avionics Architecture

Federated architecture include all kinds of avionics systems developed from the early design of avionics systems in 1930 until the emergence of IMA concept. In federated architecture each avionics function has its own dedicated Line Replaceable Unit (LRU) with a separate power supply and wiring. In other words, the federated architecture comprises a number of functions that are interconnected, yet are independent functionally. Consequently, in this architecture, due to having a huge number of LRUs the overall avionics architecture becomes quite heavy and expensive [7]. However, the federated architecture was accepted as a low risk approach in terms of reliability. Figure 1-1 illustrates an example of federated architecture. Each system has its own devices. Some aircraft of this system architecture include Airbus 310 and subsequent models, Boeing 757/767, 747-400, 737-300/400/500, and Avro RJ [8].

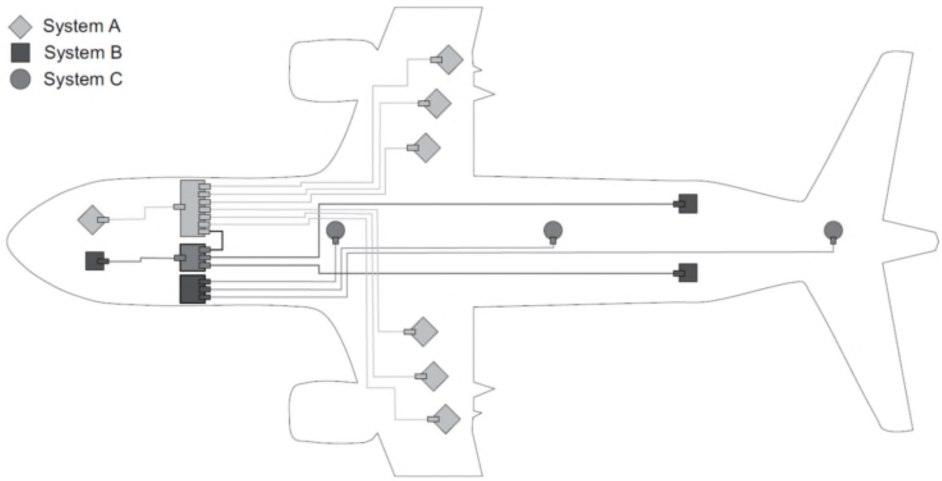


Figure 1-1 Example of a Federated Avionics Architecture

In the course of time, the number of avionics functions as well as the need for interactions and crosstalk between different systems and/or sub-systems have increased rapidly. This made the previously independent avionics system dependent. Therefore, in 1980, air-framers were looking for new solutions to reduce size, weight, and power consumption (SWaP) constraints while keeping the system integration complexity manageable. This then led to the introduction of Integrated Modular Avionics (IMA) architecture.

1.2.2 Integrated Modular Avionics (IMA) Architecture

The main idea of IMA architecture is to standardise the avionics hardware and software as well as resource sharing. The resources need to execute and share a number of aircraft functions which are provided on a standardised hardware, called modules. One module is shared between several functions by using a strict and safety critical partitioning approach. Modules are different types in that they provide computing resources, I/O resources or both. Avionics functions devices in this concepts are replaced by software implementations on IMA modules. In other words, each module comprises a Core Processing Module (CPM) that can host various avionics applications. The main technology used is a special safety critical Real-time Operating System (RTOS) on modules. The Backplane data buses are used for intra-connection communications of the applications. In addition, the IMA modules and devices communication is also enabled by a standardised high-bandwidth data bus called Aircraft Data Communication Network (ADCN) [9], [10].

The IMA architecture has advantages and disadvantages. The key benefits are better usage and optimisation of hardware resources and increasing flexibility, which results in a smaller hardware architecture, lower power consumption, decreasing weight and improving the ability to add new software without the need to update the hardware. The key disadvantage, however, is increasing system integration complexity in that the optimal allocation of software applications to computing nodes while balancing multiple competing resource demands is a very challenging task. Figure 1-2 depicts an example of IMA avionics architecture where aircraft systems share two common IMA modules.

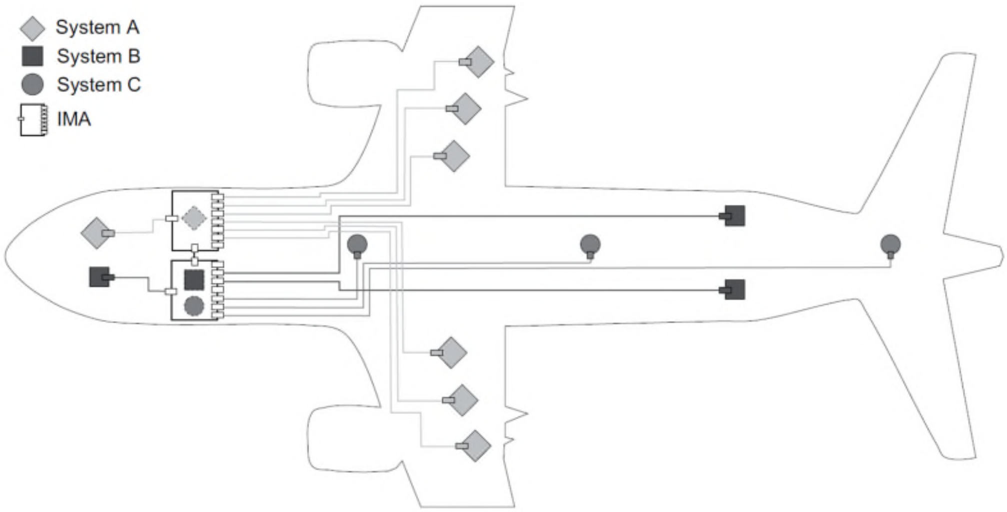


Figure 1-2 Example of an IMA Avionics Architecture

Both Airbus and Boeing have taken this approach in their latest products. In fact, the first aircraft that IMA avionics system has been installed was Boeing 777. The IMA system on Boeing 777 is called Airplane Information and Management System (AIMS) which is incorporated in two redundant cabinets that are equipped with customisable Core Processing Module (CPM) and Input/output Modules (IOM). The cabinets communicate via four ARINC A629 buses. Meanwhile, Airbus introduced IMA with the A380. The IMA in A380 is comprised of Core Processing Input and Output Modules (CPIOM) and Input and Output Modules

(IOM). The modules are connected via two redundant Avionics Full Duplex Switched Ethernet (AFDX) networks. IMA modules in A380 and B777 are installed within avionics bay. This is known as a centralized approach in installation of modules. The centralized installation requires long cables, which significantly impact the overall weight of the avionics system as well as latencies and response time over the data buses. Thus, the decentralized concept of Distributed Integrated Modular Avionics (DIMA) has been developed.

1.2.3 Distributed Integrated Modular Avionics (DIMA) Architecture

DIMA architecture is to place IMA modules and/or devices in several locations all over the aircraft structure in order to reduce cable lengths and response times. Therefore, new types of devices were also needed to enable local I/O handing and control. This is to collect sensor and actuator data locally by I/O handing devices like Remote Data Concentrator (RDC) and forward the data with ADCN to centralized modules [11]. However, remote devices do have requirements in terms of vibration, heat, EMI resistance as well as network reliability and latencies. Figure 1-3 shows an example of DIMA avionics architecture.

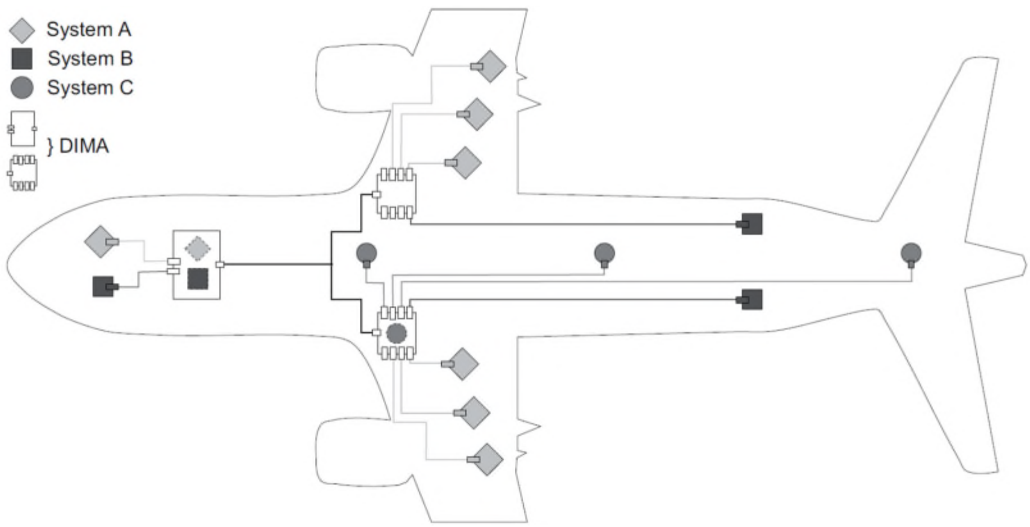


Figure 1-3 Example of a DIMA Avionics Architecture

The first aircraft that used DIMA-like avionics architecture was Boeing 787. The avionics systems of B787 consist of centralized computing modules and distributed RDCs. The computing modules called General Processing Modules (GMP), are installed in dual redundant cabinets forming the Common Core

System (CCS). In addition, RDCs, which are installed locally, are gateways from digital and analogue peripheral (sensor and actuator) signals to a dual redundant AFDX network that interconnects all DIMA components.

The implementation of DIMA architecture by Airbus is found in A350XWB. The A350 avionics systems comprise two CPIOM type hosting applications and Common Remote Data Concentrators (CRDC), which are installed throughout the aircraft. CRDC provides local common digital and analogue interfaces and route their data to a dual redundant AFDX network. Both Airbus and Boeing managed to reduce weight of avionics systems and increase the number of functions hosted on modules by DIMA-like avionics architecture.

1.3 Evaluation Criteria of Avionics Architecture

1.3.1 Avionics Architecture Evaluation Criteria

The quantitative evaluation of architecture is recently driven by the development of modern avionics systems. Due to increasing system complexity and the demand for shorter development cycles, new system design methods are required. The parametric and quantitative architecture evaluation is an approach to help system architects to choose the best possible solution among feasible architectures based on certain suggested criteria. The evaluation criteria can then be used by decision-makers for architecture trade-offs. The designers can also use parametric evaluation to pre-assessment and identification of preferred architecture, which shall be studied in more details in next steps. In addition, the parametric evaluation criteria can be exploited as a basis for automatic architecture optimisation.

The easiest way to derive the evaluation criteria is to do requirement engineering i.e. to analyze customer needs, economic, technical and legal requirements. The evaluation of an architecture is an iterative process during a system design. Firstly, it has to be fulfilled to qualify the system solution like safety, operational capability and reliability. These are referred as “Shall” criteria. Secondly, the architecture should represent a good performance in order to surpass its competitive aircraft in the market by having lower weight and power consumption for instance. These are referred as “Should” criteria. A number of related evaluation criteria for avionics system architecture which have been used in different literatures is presented in table 1 [12]. The table denotes the objectives as shall or should, and gives it sense in terms of maximizing or minimizing. In this research, in particular, operational capability has been introduced.

Table 1 Classification of Evaluation Criteria

Criterion	Shall	Should	Objective
Weight		×	Minimizing
Power consumption		×	Minimizing
Volume		×	Minimizing
Reliability	×		Maximizing
Operational Capability	×		Maximizing
Safety	×		Maximizing
Maintainability		×	Maximizing
Availability	×		Maximizing
Cost		×	Minimizing

1.3.2 Definition of Evaluation Criteria

The **Mass** of the avionics systems is the sum of all avionics hardware components, wiring and installation materials. In other words, the mass of hardware is the accumulated mass of all devices, ADCN mass and peripheral wiring mass. ADCN mass is the mass of network cable and switches, which is effected by ADCN technology, redundancy level and the distribution of IMA modules. The peripheral mass is the mass of all cables between IMA modules and sensors/actuators. In reality, however, the mass of a system often include an 'installation factor' which includes items which are difficult to account for in the equipment and wiring. Such items include connectors and shells, equipment mounting tray or attachments, harness clips which are decided by the installations designers depending on the installation standards for the project. In other words, designers will be given a target mass which their system shall not exceed. This flows down from the aircraft target mass and performance criteria. Therefore, it is one of the most important criteria for evaluation of aircraft systems, as it has a significant impact on aircraft fuel consumption as well as aircraft performance.

$$m_{EvalSys} = m_{wiring} + m_{modules}$$

The **Volume** of avionics system architecture is the space which is needed to install all modules within aircraft structure. It plays a significant role for the technical feasibility of the architecture. Particularly, it is a critical function for small high performance military jets where the volume to install equipment is severely limited. The volume of avionics modules (LRUs/LRMs) is defined by the depth (d_n), the width (w_n) and the height (h_n) of the cuboid.

$$V_n = d_n . h_n . w_n$$

The **Power Consumption** is the amount of electrical power required to operate avionics system. This power needed is for IMA modules and their connected sensors and actuators as well as power for cooling the modules. If the avionics system consumes a higher amount of power, then less power is available for other systems. In other words, more power needs to be produced, which requires a higher fuel consumption. Thus, minimizing power consumption is a major objective for aircraft system.

The **reliability** of avionics system is the probability of IMA system to operate in a certain period of time without fault. The reliability required for a number of functions in aircraft systems is defined by their safety requirements. The improvement of the reliability lowers the time for scheduled and/or unscheduled maintenance. On the other hands, function's reliability depends on IMA devices and network reliability i.e. the Mean Time between Failure (MTBF) for those components.

$$R(t) = e^{\frac{t}{MTBF}}$$

The **operational capability** of avionics system on aircraft level is defined as the performance and capabilities of avionics functions. It is a combination of hardware and software features which impact aircraft performance. In this research, each avionics LRU is evaluated against a set of criteria to calculate the operational capability of avionics system. The criteria for each avionics subsystem are different as they do various tasks. In other words, the operational capability of the avionics system tells us how good a system can do its tasks. Multi-Criteria Decision Making (MCDM) technique is used to calculate the operational capability of each LRU which will be discussed in details in chapter 3.

The **safety** of the IMA architecture is to describe that any malfunction in the property of architecture does not harm or destroy equipment or structure of the aircraft. Generally, a malfunction, safety critical event, is classified into levels from minor to catastrophic regarding its impact on other systems. Safety is a “shall” object i.e. the chosen system architecture has to ensure that each function meets its safety requirements. The contribution of IMA system in safety of the system has to do with hardware reliability, installation locations, and function allocation.

The **maintainability** of the avionics system defines the efforts for checking, repairing and upgrading the avionics systems. In other words, it is related to time to access to the installation location, mean time to repair (MTTR) and/or removal and also time to loading and testing. For example, the installation location can determine the easiness of repair. Also, LRU’s MTBF determines the frequency of maintenance requirements.

The **availability** of avionics system is the time a function is operational. It depends both on MTBF and MTTR. The availability of IMA system architecture can be measured by using its allocated components. The measurement is defined as follows:

$$a = \frac{MTBF}{MTBF + MTTR}$$

The **cost** for avionics system architecture can be defined in many cases including Ship-set Cost (SSC), Initial Provision Cost (IPC), Operational Interruption Cost (OIC), Manufacturing Cost and operation cost. The definition of all in details is in literature [11].

1.4 Avionics Communication Data Buses and Protocols

1.4.1 Communication

Avionics communication networks have experienced a significant development since the evolution of avionics. In the very beginning, avionics systems had a few separate navigation, communication, radar and cockpit displays connected via dedicated wiring. As time went by and aircraft became more advanced, the number of avionics LRUs increased and systems became more complex. By the advancement of digital technology, equipment was designed to communicate to each other, and the use of data bus became necessary in aerospace industry. The emergence of avionics data bus contributed to wiring reduction and ease of maintenance.

The communication in IMA architecture is between IMA modules and other separate LRUs that are not integrated. The communication between IMA modules is known as Aircraft Data and Communication Network (ADCN), which is a single high bandwidth data bus like ARINC 629, ARINC 664 (AFDX) in civil aircraft and MIL-STD-1553 for military ones. In what follows, an overview of the most common technologies used for the ADCN and/or fieldbus is summarized.

1.4.2 Avionics Data buses

ARINC 429. ARINC 429 is the first and the most commonly used digital avionics bus. A429 is a one sender and up to 20 receiver connection. Signals are transmitted by using a serial twisted wires that can connect peers up to 90 meters. The data is transmitted in 32 bit words and are continuously transmitted. The bandwidths available are 12.5 kb/s for low speed and 100 kb/s in high speed mode. To build an avionics system architecture using A429 leads to a huge number of data bus links since every transmitting LRU has an output channel and every receiving LRU has one input channel from which it receives data. The consequence is a complex physical network, however, its data protocol is very simple [13].

ARINC 629. ARINC 629 is a multiple source, multiple sink data bus with a bandwidth of 2 Mb/s and was developed specifically for Boeing 777. It uses a 20 bit word format and implements Time Division Multiple Access (TDMA), i.e. each sender has a Transmit Interval (TI). It also supports signals to be transmitted and received by up to 128 terminals. In addition, it supports triple and quadruple redundancies. The only aircraft used this protocol i.e. B777 did implement dual redundancy in most of aircraft system, triple redundancy in flight control, and quadruple redundancy in Engine' Electric Controllers (EECs) and AIMS cabinets [14].

ARINC 664/ AFDX. ARINC 664 known as Avionics Full Duplex Switched Ethernet (AFDX) is a high bandwidth data bus based on Ethernet technology, which is currently used in aircraft like A380, A350 and B787. AFDX has two wire bus, one transmit and one receive channel. AFDX comprises end-systems, switches and virtual links (VL). End-systems (avionics LRUs) are connected to switches and forward the AFDX messages on multiple output ports as well as other switches and end-systems based on addresses and routing tables. The AFDX uses a particular packed data structure by using Ethernet, however, the AFDX packets do not contain the address of a receiver rather a so-called Virtual Links. A VL specifies one sender and an arbitrary number of receivers. All switches in the AFDX network are statically configured to forward a packet with a certain VL identifier to the a priori defined receivers always on the same route [15].

CAN-Controller Area Network. The CAN bus is a multi-master, message broadcast that provides a maximum signaling rate up to 1 Mb/s which was originally developed for automotive industry. For aerospace industry, CANaerospace and ARINC 825 are defined. The CANaerospace is used as a backbone network for flight state sensors and navigation systems. A825 is an extension of CAN bus for aircraft. The communication protocol of CAN bus is organized in messages of eight bytes payload. Each message is identified by an ID of either 11 bit or 29 bit. CAN is a robust bus as it uses a differential signal that makes it more resistance to noise. Therefore, it is a great choice for embedded applications in hazardous environments or areas with a lot of electromagnetic interference [16].

TTP- Time Triggered Protocol. TTP is an open computer network and a real-time protocol which is designed as a time-triggered fieldbus for vehicles and industrial applications. In aerospace industry, TTP has been implemented in power generation, environmental and flight control systems as well as FADEC and flight computers. In principle, the transmission of TTP messages are in one-to-n manner. The mechanism for controlling bus access is based on TDMA. TTP can also provide time-synchronization, error detection, and redundancy management. TTP can be implemented for electrical two-wire and optical data transmission. Its static schedule provides very low deterministic latencies in microseconds range [17].

1.5 Research Aims

This PhD is based on FUCAM (Future Cabin Design for Asian Market) project aiming to develop a 2025+ conceptual cabin design devoted to the Asian market, for short and medium range aircraft. FUCAM was to investigate current, emerging, and completely new technologies based on users' requirements and market customizations. In other words, FUCAM was to incorporate the most promising technologies for the integration of this cabin concept into the aircraft through technology assessment in order to raise its maturity level to TRL3. The focus in FUCAM was on technologies related to aircraft cabin electric/electrical systems, however, this PhD project has been extended to all avionics systems aboard the aircraft.

In recent years, the idea of how to design integrated modular avionics architecture to achieve best performance and optimality has become one of the major research interest in the aerospace industry including aircraft avionics networks, satellite avionics and unmanned aircraft systems. Avionics architectures include digital processing modules and communication buses which support many avionic applications such as flight control, flight management, fuel management, stability, guidance, passenger entertainments, etc. Furthermore, their complexity is constantly growing which means more functions need to be integrated. Consequently, avionics architectures have become a crucial component of an aircraft. They have to guarantee a variety of important requirements including robustness, safety, flexibility, maintainability, and optimality.

To satisfy these requirements, traditionally manual design is no longer desirable. It wastes both time and money. Therefore, researchers and designers have been investigating and developing a mathematical and Model-based tools which employ optimisation techniques to achieve an optimal avionics system architecture. While the development of model based system engineering tools and mathematical modelling have opened new solutions to achieve some specific optimal criteria, an integration of various optimisation routines and a holistic algorithm aid on aircraft-level avionics systems integration architecture remain challenging.

Similar concept has also been studied from other industries like space (launcher) and automotive, in particular, automotive electric/electronic architectures. Due to ever-increasing number of functions to be integrated in Electronic Control Unit (ECU), well-known automotive manufacturers like Volvo, Siemens, and BMW have been doing quite a lot researches to optimise the intra-vehicle electronic architecture using multi-objective optimisation methods.

As mentioned above, avionics architecture has been optimised in terms of mass, cost, safety and reliability with some level of automation in different integration levels. However, what is missing is to improve operational capabilities of avionics systems, i.e. the aircraft would be able to fly with specific RNP accuracy for instance or would be able to meet upcoming CNS/ATM requirements like 4D trajectory optimisation and RTA while optimising weight at the same time. Furthermore, the avionics equipment integration can also increase situational awareness and help increase landing system with a precision approach capability. This cannot be achieved unless considering the detailed specifications of avionics LRUs/LRMs. Therefore, a holistic and multi-objective optimisation of avionics system on aircraft-level regarding supplier complexities as well as the operational capabilities of the equipment is still a gap in this area.

The aim of this PhD project is to develop an avionics integration optimisation software system (AIOSS) based on mathematical programming to help decision makers and/or avionics system architects to investigate various architectures at the early stage of design in terms of certain suggested criteria including SWaP-C and operational capability. The tool has three main parts including a database, avionics architecture modelling and optimisation techniques. Based on aircraft level avionics requirements and architecture description, the tool (DADO) system is to optimise avionics integration and output the alternative optimal avionics integration architectures, and the benefits of each alternative. Figure 1-4 represents the overall structure and processes in this PhD project.

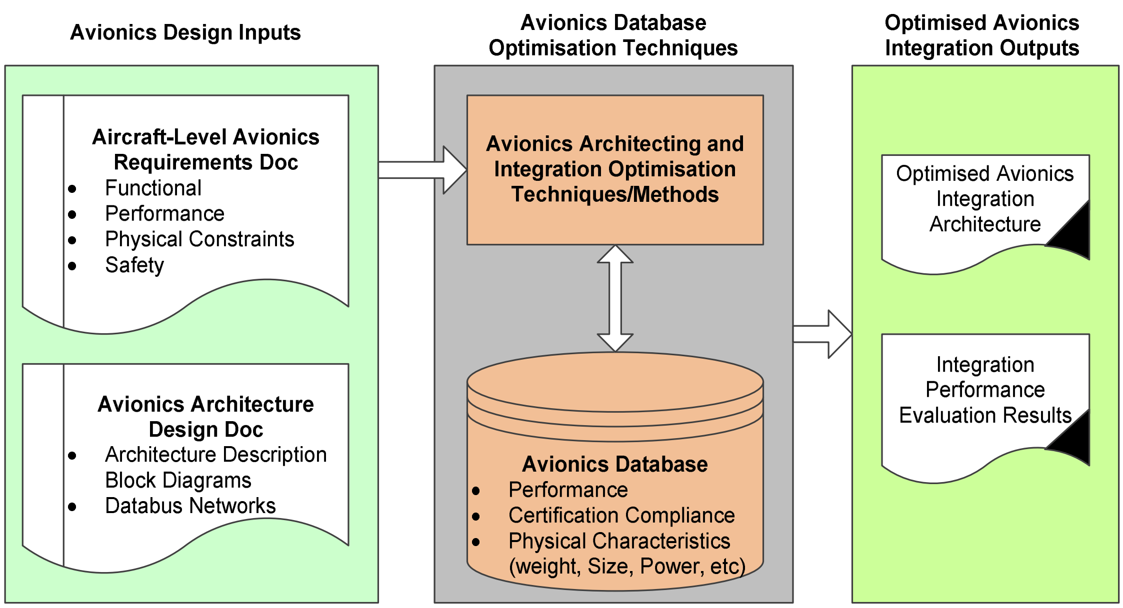


Figure 1-4 DADO Project Structure

It should be noted that the proposed methodology in this thesis has not been validated on the real industrial scenario and it has been remained at algorithm/simulation level. However, the methodology could potentially be gone through the process of industrial implementation where it would be used in the early design stage as a decision-analysis tool and/or for upgrading a legacy architecture. In both cases, having a reference architecture would facilitate things to get validation for the proposed methodology. Furthermore, the avionics architecture can also be evaluated from a number of other criteria as defined in this chapter. In this research, however, mass, volume, power consumption, MTBF and operational capability of each avionics LRU are studied. Other criteria like cost, complexity, reliability and safety are beyond the scope of this research and are proposed for future work.

Since it is advantageous to utilize COTS components to decrease the cost of on-board equipment suite's lifecycle, the functional and operational capabilities of the proposed LRUs/COTS are also evaluated for further optimisation of the avionics architecture. Also, almost all the avionics requirements captured for this research are from LRU/COTS components perspective at aircraft level which are mainly related to CNS/ATM requirements. In other words, the requirements studied are mostly from regulatory documents.

A more detailed requirements like requirements on avionics flight reliability and safety model in the form of Functional Failure Analysis (FFA) and/or Functional Hazard Analysis (FHA) at aircraft level as well as interface and network requirements are proposed for future work.

1.6 Research Objectives

The objective of this PhD is to develop a method and tool for Avionics integration optimisation architecture in terms of weight, power consumption, volume, reliability and operational capability. In other words, this project is to provide an open-system avionics architecture for ease of aircraft upgrades as well as preliminary design space exploration regarding to future CNS/ATM and operational requirements using COTS products. Particular research objectives have been identified and outlined as follows:

- To do functional decomposition of avionics systems on aircraft-level to provide a generalized datum architecture and identify functional links.

- To draw system architecture of the proposed datum functional architecture by using avionics data buses. The physical architecture is divided into two parts including small scale and large scale. The small scale is the Automatic Flight Control System (AFCS) architecture and the large scale is the whole avionics architecture.
- To develop an avionics database of solutions for each avionic function. A database in excel has been developed to record avionics LRUs/LRMs from various vender products with their technical specifications including mass, volume, power consumption, MTBF, and operational capability.
- To assess the operational capability of each LRU separately using Multi Criteria Decision Making (MCDM) method. The method used in this research is Simple Additive Weighting (SAW).
- To capture aircraft-level avionics requirements to improve the operational capability of the proposed architectures.
- To model avionics integration architecture. The proposed architectures are modelled using mathematical programming to assign the best avionics LRUs/LRMs to the architecture while satisfying some design constraints. The model include decision variables, constraints and cost functions.
- To seek optimised architecture by implementation of optimization algorithms including exact method, branch-and-bound algorithm, as well as heuristic method, PSO and WSM.

1.7 Thesis Outlies

Chapter 2 reviews avionics integration and architecture optimisation methods. It tries to discuss each method in details and the pros and cons of each is summarised and compared.

Chapter 3 describes the methodology developed for the optimisation of avionics integration and architecture. The method developed is realised based on industrial processes and Cranfield University' internal projects in Aerospace Vehicle Design (AVD) group. It is comprised of functional and physical architecture, development of avionics database, and operational capability assessment of each LRU.

Chapter 4 introduces the mathematical foundations of architecture modelling used in this thesis. The modelling technique used is based on linear programming which is a widely used method to translate real-world problems into mathematical expression. The avionics integration and architecture is defined as a general assignment problem. This chapter also introduces exact and heuristics/meta-heuristics methods being used to solve these types of problems.

Chapter 5 solves the architecture optimisation problems. The different cost functions defined in chapter 4 will be solved using commercial solver, i.e. GAMS and MATLAB for exact and heuristic methods. Both single and multi-objective optimisation are analysed and the results are discussed in details.

Chapter 6 summarises the entire thesis and provide final conclusions, contributions and discussions as well as recommendations for future work.

2 Literature Review

2.1 Introduction

Currently, the avionics system architectures have reached a level of complexity that designing new architectures manually takes a lot of time. Also, the ever-increasing number of avionics functions and emerging technologies complicate manual design even more. Moreover, from industry point of view, it is crucial to exploit the full potential concept of IMA and/or DIMA architecture which seems impossible by designing architecture by hand. Therefore, in recent years, a trend to create computer-aided design and model-based design tools is growing [18]. The main goal of these tools is to provide an automatic and/or semi-automatic avionics architecture optimisation tool. The input to these tools is a model that represents the system and allowing formal analysis, verification, evaluation as well as simulation. On the other hands, avionics integration (functional and physical) is also an important task which tries to make the best use of software and hardware resources. In other words, avionics integration is to utilise resources, management processes to identify the sharing of resource capabilities, the potentiality of reuse of resources in order to improve equipment utilisation, operating efficiency and availability. In what follows the very basic principle of avionics integration is defined.

2.2 Principle of Avionics Integration

The term integration is widely used in aerospace industry including integrated team, integrated products and integrated solutions which invoke a variety of definitions. It therefore seems that the meaning of integration becomes ambiguous in many cases. The dictionary definition of integrated is “made up of parts”. Integration thus is the process of bringing these parts together to make a whole. Consequently, to define integration, the parts that are going to be combined and merged need to be defined. In aircraft system design, the integration can happen in different levels like component, function, system, process and information level [19]. Integration happens because engineers want to investigate solutions that are more efficient in operation and in their use of equipment. To do so, designers often incorporate many functions into one hardware component and/or Line Replaceable Item (LRI) [6].

Therefore, the first step in avionics system integration is functional integration which brings together functions. In other words, functional integration is the level of integration that defines how the functions are partitioned and how they interrelate. In functional integration the distinctions between previously separate boundaries is the principle safety issue. On the other hands, physical integration is the level of integration that defines how the functions are implemented by hardware and software components. It is therefore the task of deciding how the system is to be implemented in real world in terms of its geographical location, electrical isolation, logical view of its environment and hardware modularity.

It should be noted that, if a system contains ‘n’ functions, the physical integration of that system may not yield a system containing ‘n’ physical elements. In other words, there may not be a one to one mapping of functional to physical elements [20]. In the process of physical integration, the system functional partitioning may become unclear. There is no formal link between physical and functional integration, however, there is a trend, and that is, as the level of functional integration increases, the necessary co-operation between functions becomes more complex and demanding. This is true about Integrated Modular avionics. While IMA architecture reduces constraints like size, weight, and power consumption, the level of system integration has increased. To overcome the complexity of system integration and optimise avionics architecture, researchers are striving to provide tools and/or methods to ease the process and exploit the full potential of the integrated architecture. To do so, the first step is to model the avionics architecture.

2.3 The State-of-the-art Avionics Architecture Modelling

A couple of modelling approaches are being used in avionics architectures ranging from static system modelling over safety modelling to dynamic timing analysis. This section reviews models used in the context of IMA and/or DIMA architectures, and their strength and weakness are addressed. It should be noted that, some general system modelling approaches like UML [21] or SysML [22] can also be used to model some aspects of IMA architectures, however, UML profiles and their lack in strictness, for instance, makes it difficult to verify, evaluate and optimise IMA architecture automatically. Thus, they are not studied in details here.

2.3.1 Things need to be modelled

The main goal in avionics system architecture modelling is to enhance the planning and the design process as well as improving the operational capability of the desired architectures. Thus, it is critical to fully understand all aspects of the avionics system architectures including the driving requirements, IMA architecture components and the process of functions allocations to hardware, and hardware mapping of the components into aircraft structure. Figure (2-1) represents an overview of the IMA/DIMA architecture elements from software mapping to hardware mapping and their associated constraints and qualitative/quantitative measures.

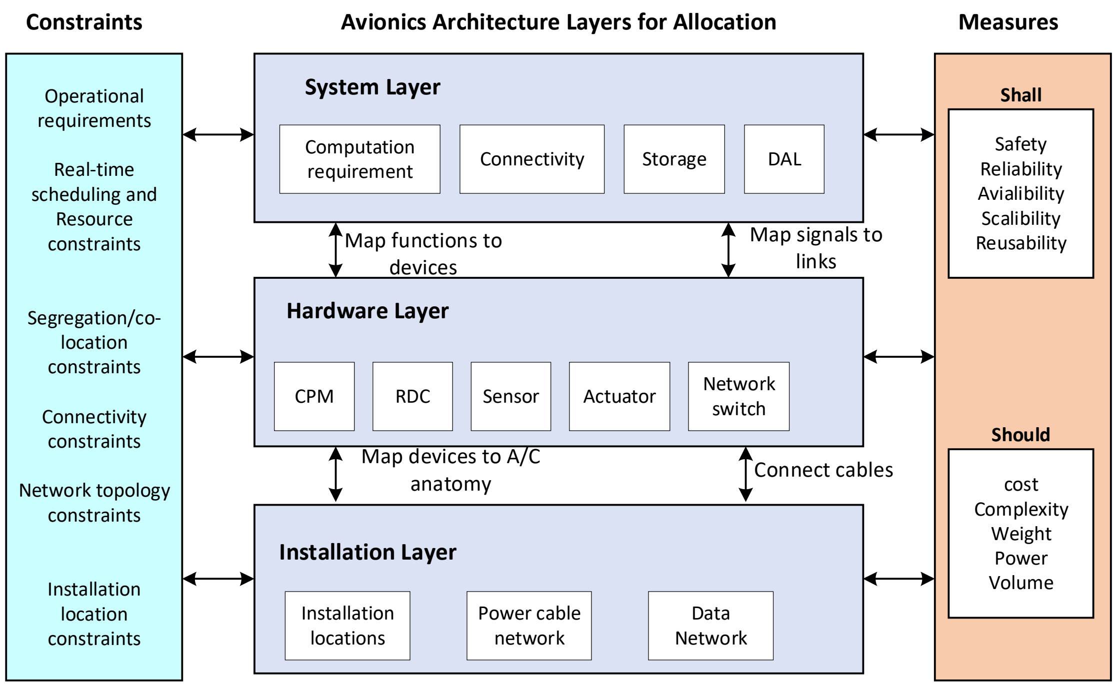


Figure 2-1 IMA/DIMA System Architecture Elements and Design Layers

The **system layer** is to describe the functional breakdown and/or decomposition i.e. each subsystem has its dedicated technical specifications like resources (computation and storage), connectivity and capacity as well as other properties associated to IMA/DIMA architectures such as Design Assurance Level (DAL). The aim of this layer is to identify the system tasks, logic and connections as well as required sensors/actuators and interfaces for functional subsystems. For instance, a high lift system task/function is the extension and retraction of flaps and slats triggered by a flap lever in the cockpit. This may include a sensor acquisition, a monitoring and actuator modules. These modules are enabled by a set of hardware components in hardware layer including CPMs, RDCs and sensors.

The **hardware layer** is all about hardware devices and their configurations as well as network topology that support the functions in the system layer. The typical devices in IMA/DIMA architecture are CPMs, RDCs, I/O modules, sensors, actuators, switches and cables.

The **installation layer** is to allocate hardware to aircraft anatomy. It is indicated by installation locations and power network. The devices can be placed at different installation locations like avionics bay, cockpit, centre and tail of the aircraft. These locations have their own constraints including mass, volume, cooling capability, number of slots and power supply. These criteria must be taken into account for the optimisation of architecture while placing the devices in their installation locations. It can affect both the cable length and safety requirements.

All the three mentioned layers must be carefully modelled and designed by taking into account the required measuring criteria to achieve an optimised avionics architecture. In what follows some modelling techniques and/or tools are reviews.

2.3.2 Architecture Analysis and Design Language (AADL)

AADL is one of the most popular tool in avionics domain [23]. It is a modelling language to describe software and hardware of critical and embedded systems. The hardware in the AADL is modelled as a composition of processors, memory, buses and devices. Software is comprised of processes, threads, subprograms, and data. The system is comprised of an arbitrary set of software and hardware components. Several systems can be implemented. Further, components can be linked together by either signal relations or be child relations (a hierarchal dependencies), for instance, assigning a process to a processor. AADL is standardize by SAE, including an XML data format and graphical representation [24]. It is freely available within the TOPCASED project [25]. Since AADL is a universal modelling language, it can be used for software and hardware system architecture modelling. It also allows to specify different models of the same system. The freedom in creating AADL models does not allow automatic evaluation and simulation of the models. Thus, AADL extensions and domain-specific models in relation of other modelling or simulation languages have been proposed to express IMA software and hardware interaction.

2.3.3 AADL-based Modelling of IMA

The different layers shown in figure (2-1) have been modelled by AADL. Fraboul et al propose an IMA domain model consisting of four layers including application, architecture, mapping and execution [26]. The application model include ARINC 653 partitions and ports as well as their connections. The architecture layer comprises CPM and buses. The application layer is bound to the architecture in the mapping layer. Finally, it is viable to automatically derive discrete even simulations by combining layers with execution layer, which holds run-time information on hardware and software.

Delange et al also proposed an AADL scheme to model ARINC 653 partitioned systems [27]. The scheme defines how A653 partitions, processes, communication ports, health monitoring as well as A653 hardware is modelled. It defines the AADL classes to use and set of attributes. The proposed rigid modelling provides a derivation of executable dynamic schedule models. The approach is used for model-based verification of system and A653 OS layers. Further, a simulation approach of configured IMA system down to hardware is given by Lafaye [28]. In this paper, it is presented how to transform AADL models in SystemC models. SystemC models are executable and used for hardware software interaction simulations. A unique one-to-one mapping between AADL classes to SystemC modules is given. Moreover, with the rigid modelling rules, transaction level SystemC models (TLM) can automatically be derived. Also, a demonstration shows how to verify processor and memory load limits on an IMA device loaded with system software over time.

2.3.4 Mathematical-based Modelling of IMA

Foerster proposed π-calculus as a textual modelling language for avionics and IMA systems specification and verification [29]. π-calculus is a formal mathematically motivated process specification language. The modelling elements are processes and their communication. The Execution of π-calculus allows formal verification of the logical correctness of the modelled systems. The proposed model covers functional and communication specifications as well as hardware and network topology. The automatic derivation of specification is enabled by specifying software-hardware bindings. The simulation of this implementation is used to verify complex software/hardware systems.

In addition to IMA devices, avionics networking i.e. ADCN modelling has also been modelled in some literatures. AFDX end-to-end delays are analysed by Charara where he compares network calculus, the queuing network modelling approach, and model checking [30]. Network calculus is a mathematical model expressing each network node by queuing capability and queue size. For given message sizes and arrival rates this can be propagated through the network automatically. The queueing approach builds up networks from basic building blocks like links, buffers, and multiplexers. Underlying behaviour and configurable properties of each block allow a simulation of network traffic to obtain delay information. Model checking uses timed automata to express network nodes. Comparing the three approaches the precision goes up from network calculus to model checking as the calculation effort does.

Lauer also proposed a modelling and simulation approach to verify latency and freshness requirements in AFDX networks based on the tagged signals model [31]. The Loaded IMA systems are modelled as processes and signals transmitted over timed channels. IMA devices with applications are expressed as processes. AFDX links are timed channels. The transforming tagged signal models are expressed in mathematical optimisation problems in which end-to-end latency and freshness parameters can be calculated for different network setups. The proposed approach is applied on a FMS composed of a switched AFDX network, RDCs, and the cockpit panels.

2.3.5 General Assignment and Mapping Problem

The assignment problem answer the question of how to assign m (jobs, students) items to n other items (machines, tasks). Since there are many possible ways to assignment, it is then defined as an optimisation problem to achieve the best suited assignment for the problem [32]. The mathematical model is defined as follows. Let $x_{ij} = 1$ if task i is assigned to person j and 0 otherwise. Let C_{ij} the cost of assigning i to$j$. Then, the objective function is to minimize the total cost of the allocation which is to

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}x_{ij} \tag{2-1}$$

Each task goes exactly to one person and each person gets only one job. These are given by the following constraints

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall\, i \tag{2-2}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall\, j \tag{2-3}$$

The decision variable is defined as a binary integer programming (BIP), i.e. $x_{ij} \in \{0,1\}$.

One of the applications of assignment and/or mapping problem is the allocation of software applications to processors in distributed computing modules [33]. The objective in these problems is that a set of tasks are assigned to processors for the shortest time execution. The assignment can be either static or dynamic. The processes are restricted by resources like calculation time, memory or bandwidth. The network can also be considered. In other words, the communication topology plays a significant role when optimising the overall execution time. The assignment problem classes differ in multiplicity of assignment, the number and type of resources, the number of assignment layers, constraints and the objective functions. This problem is well-known to be NP-hard in most cases, i.e. it cannot be solved by a deterministic algorithm in certain amount of time.

In other words, the optimal solutions can be found by an exhaustive search, yet there are n^m ways to assign m tasks to n processors for instance, the search is often impossible. Therefore, optimal solution algorithms only exist for small problems. This combinatorial optimisation problem can be solved by exact and heuristics methods which results in optimal and suboptimal solutions respectively. The best solution algorithm depends upon the problem properties and the problem size. The exact methods usually work for small instances, yet for the large instances heuristics are created. Two examples of distributing computing systems defined as static assignment problem and solved by heuristics are given below.

Lo created a three step heuristics method for assigning tasks to processors to optimise overall execution time and communication cost. First, tasks are assigned in an artificial two processor network, which is optimal but not complete. Thus, it is repeated iteratively with updated processor and network capacities. Finally, the unassigned tasks are assigned by trial to the first capable processor. They managed to get a great objective values in short runtimes for group of 34 tasks and six processors [34]. Kafil and Ahmad used the A^* algorithm to solve the same problem. The algorithm is a non-deterministic global search algorithm using a search tree to calculate the costs of visited parts. The method further was extended by reducing the number of nodes, which happened by an initial guess. The results for a 20 tasks and 4 nodes have been calculated in very short runtimes [35].

The distributed systems assignment problems are basically similar to the issues in IMA and DIMA architectures design. For example, assigning the avionics functions to IMA devices is an assignment problem. Also, software mapping is a static assignment problem. Moreover, IMA task assignment depends strongly on resources like power, memory, and I/O types. The safety requirements are also very important. Similar examples of these types of problem can be found in operations research. In what follows the IMA related works are overviewed.

2.4 An Overview of IMA Architecture Optimisation

The aim in avionics architecture optimisation is to support the integrated modular avionics design process with automated and/or semi-automated architecture generation. To achieve this goal, the IMA design problem must be expressed formally and mathematically. A mathematical algorithm is then applied to the model to do the exhaustive design space exploration. The input and output of the model is IMA architecture models. This enables designers and/or decision makers to investigate various possible architectures as well as the optimal one.

The automated design and sizing of IMA architecture is a relatively young field of research. However, the idea of how to make the best use of resources, resource allocation problem, in computer science and general assignment problems in operations research are very old. In addition, the automated design and optimisation is also carried out in other industries and disciplines like space, automotive, economics, and infrastructure planning. In what follows an overview of literatures on general distributed computing system optimisation as well as IMA architecture optimisation is given.

Over the past decade, the implementation of IMA architecture has been widely used by aircraft manufacturers. IMA architecture optimisation emerged with the creation of the first IMA systems. On early years, the priority was on validation rather than optimality. The main issues in IMA architectures are on processor scheduling and finding safe and reliable mappings of functions and signals in networks. Many approaches previously have been studied for model-based design and verification of IMA architectures. Most of them are related to one hardware devices or a single avionics system. Only a few approaches exist that handle the whole software and hardware architectures, multi-devices and multi-applications, as well as spatial distribution.

Uwe and Reinhard developed a static model for the automatic design of IMA architecture i.e. the allocation of avionics functions to IMA devices and placing the devices in aircraft structure [36]. The model provides some automation in IMA design process by applying optimisation algorithm and focuses on safety evaluation. The model expresses logical, physical and structural aspect of avionics systems. It includes propagation failure layers, function instantiation, hardware, hardware types, and geometry. The failure propagation model expresses system logic in terms of functions and connections. The failure propagation is dealt with in the three stages including 'ok', 'passive', and 'out of control'. Signals and components are 'passive' if an error is detectable and 'out of control' if not.

The hardware types defines all the components with their technical specifications like failure probability and cost. These components are then allocated to geometry model. The function instantiation expresses the operational system which comprise multiple instances of the functions and connections from the failure propagation model. The redundancy level is also defined in function instance model which represents the functions of failure propagation model. The complete model provide the automatic derivation of fault trees and the computation of reliabilities with some simplification like excluding failures canceling each other [37]. The optimisation method used is a conventional genetic algorithm, and is applied to flight control system architecture. The method and the algorithm do work well for small and medium architecture like flap control system, however, for large scale architecture e.g. the whole flight control system with a high redundancy level, it does need an improvement to produce optimal results.

Dougherty and his colleagues in Vanderbilt University in collaboration with Lockheed Martin developed a tool called ScatterD to optimise embedded flight avionics systems [38]. The domain studied is mission computer and flight control systems. The deployment problem is expressed as a multi-resources bin-packing problem. In other words, the computer aided design tool deploys software applications to hardware while satisfying a number of complex constraints including processing time and real-time scheduling like processor time, memory size, and bandwidth. The optimisation algorithm implemented is a hybrid heuristics, Genetic algorithm (GA) and Particle Swarm Optimisation (PSO). The tool manages to reduce the required processors and network bandwidth consumption. The installation location of hardware and the operational capability of the system architecture are not studied.

In another literature, Manolis and his colleagues from Georgia institute of technology in collaboration with Boeing for B787 project developed a general framework called component-based system assembly (CoBaSA) that implement constrained component assembly technique [39]. The main objective of the tool is to create an environment for construction of a large industrial systems by integrating components, particularly, COTS components. CoBaSA software includes an expressive language for component interfaces, properties, and system-level and component-level constraints. Further, it uses a pseudo-Boolean solver to solve the constraints using SAT-based method. The tool enables automatic solving of system assembly problem directly form system requirement. The tool mainly contributes to a greater reliability, lower cost development, shorter development cycles as well as less testing and validation in system design and integration.

CoBaSA is further used for the integration of IMA avionics architecture components for Boeing 787 Dreamliner. The assembly problem of IMA architecture involves mapping of avionics function to LRM, mapping of LRM to cabinets, mapping of RDCs to switches as well as sensors and actuators. Moreover, system architects have to consider some constraints including WCET, I/O timing, memory, latency, network jitter and so on. The implementation of CobaSA has hugely reduced the assembly time for Boeing compared to their current methods.

Shi and Zhang also developed a tool for avionics integration optimisation using mathematical programming [40]. The tool is created in three steps called system organization, system integration, and requirement analysis. First, the system organization is based on linear programming and is to select the best vendor products according to the system performance requirements like minimizing and/or maximizing a particular cost function. In this step, suitable devices are selected for the architecture. The properties considered for the devices to be selected are performance, weight, size, power, processor unit and DAL. Then, in system integration phase, designers are able to form the optimal architecture with all the constraints defined by applying optimisation algorithms.

The algorithm used in this work is GA and PSO where user can select the algorithm for their problems. The requirement analysis further verifies the integration design by Boolean logic. The major requirement assessed in the paper is safety. Since the safety requirements are defined as Boolean, a SMT solver is used for requirement verification. The tool is basically developed in Java, and is eclipse-based. In all steps, user needs to manually choose the safety objectives and evaluation algorithm. Spatial distribution of avionics modules in aircraft structure is not studied in this work, and human machine system is not taken into account in the model for safety assessment.

Zhang et al. modelled the DIMA as a CPS integration scheme with physical layer and function layer. The physical layer focuses on mapping of CPM and RDC into a predefined locations including avionics bay, cockpit, centre and tail. The function layer is about the allocation of tasks/functions to CPM, RDC as well as sensors and actuators [41]. The integration constraints defined have to do with the maximum resources available in each location including mass, slot, and cooling capability for physical layer. For functional layer, constrains are related to calculation resources like memory and segregation constraints of functions.

To implement the scheme and apply multi-objective optimisation for DIMA system, a software based on MATLAB was developed. The multi-objective optimisation problem is solved by using an improved lexicographic optimisation technique for minimizing the weight and maximizing the reliability. The model represents that the reliability measure in optimisation is improved. Moreover, the comparison between IMA and DIMA shows that DIMA proves a better performance in reliability.

Horst Salzwedel et al. represent a new methodology for optimising avionics architecture at aircraft level. The proposed method is developed to deal with Bounded and Statistical uncertainties of early design stages related to the components of an architecture [42]. To handle these complex tasks two techniques have been employed. One technique is used to develop an automated design at system level that tackle the uncertainties, and the other one is developed for system level optimisation of avionics architecture and function. The automated design provide an environment for simulation by connecting to a database of architecture components and their connections information. Monte Carlo method is also used to handle components uncertainties. The process comprises modelling of the architectural components and network system, automated mapping of avionics function into architecture, developing an XML database format, and architectural optimisation at aircraft level. The approach managed to minimize cost, weight, and cable length, and maximize availability and generates an XML description of the optimised IMA architecture. The optimisation results in reducing wiring by 68%, cost by over 78% and increasing availability by several orders of magnitude.

Annighofer et al. developed a model based methodology for architectural design of DIMA architecture. A meta-model is defined for DIMA architecture design from system requirements and aircraft anatomy up to functional allocation to IMA hardware devices, networking, and physical installation locations. As aircraft systems require calculation power to execute their logic, and I/O interfaces to connect their peripherals (sensors and actuators), they must meet safety and performance constraints like reliability and signal latencies which are also modelled. A number of cost functions are also defined to evaluate the system architecture. These are mass, SSC, OIC, IPC. This work is of one a few works that are suitable for aircraft modelling. In [43] the IMA platform which is the combination of hardware, the system applications, signals and peripherals are modelled in different layers including software mapping and hardware mapping. The software mapping is the allocation of avionics functions to IMA hardware, and the hardware mapping is the allocation of hardware to installation locations in aircraft structure. The system constraints defined are peripheral, segregation, atomic, latency, devices, installation locations and power constraints.

Moreover, they proposed a novel method for automated device type selection, sizing and mapping of IMA architecture based on binary programming [44]. The tool is developed in Eclipse EMF framework. The optimisation is implemented in MATLAB. The model employs a combinational Multi-objective solver including Pareto front, branch-and-cut algorithm and a genetic algorithm (NSGA-II). Pareto front shows a better performance in optimality. For single objective solving COTS solver like CPLEX and GUROBI are used as well. As a case study, the method is applied to four aircraft systems including Bleed Air System (BAS), Pneumatic System (PS), Ventilation Control System (VCS), and Over Heat Detection System (OHDS). The optimisation problem is solved for mass and OIC as objective functions. The results show a great improvement in mass and OIC compared to manual design.

They managed to extend their model further to signal routing and network topology. An AFDX topology is described by devices and links and presented as an undirected graph [45]. To achieve the optimal topology, the problem is formulated as a Binary Programming (BP) which is a combinatorial optimisation problem. The network topology evaluated by costs like mass contribution while satisfying a number of constraints. The topology mass is calculated from the mass of all switches used and the cable mass of all links. The optimisation problem is solved for mass and OIC as objective functions. The results show an improvement in signal routing and network topology. The method is used to optimise full or sub- parts of avionics architectures for certain objectives such as mass and cost while considering system requirement.

Lee and his colleagues developed a scheduling tool and algorithm for optimisation of AIMS Boeing 777 cabinets [46]. AIMS is a time synchronized distributed computing system which includes several processors and I/O boards. The proposed algorithm calculates the time scheduling for all partitions, tasks and bus messages. The objective is that all partition and task deadlines are held and the capacity of processors and buses is not exceeded. They came up with a two level algorithm creating the processor’s schedules first and then calculating the bus schedule.

Zhang used MILP to find VL trees in an AFDX network [47]. It is shown how to formulate linear constraints and binary variables such that a consistent routing tree from each source node to each destination node is retrieved. The objective is an overall low bandwidth utilization of AFDX links. It is implemented for a topology of eight nodes that up to 1600 VLs can be routed in four hours. However, global optima are not retrieved.

Christophe and his colleagues in Airbus Toulouse developed a decision-analysis tool for the optimisation of fly-by-wire flight control system architecture of Airliners [48]. The tool for fly-by-wire flight control system architecture is to find a combination of actuators, power circuits and flight computers for each control surface. Different actuator technologies are considered including servo-control (S/C), electro hydrostatic actuator (EHA) and electrical-backup hydraulic actuator (EBHA). The objective is to select the system architecture that keep the weight as low as possible while fulfilling safety and technological constraints. They chose branch-and-bound algorithm to solve this discrete optimisation problem for Airbus A340 roll control system architecture. Two scenarios are solved including 3H in which three hydraulic circuits power the flight control actuators and 2H-2E where two hydraulic and two electrical circuits power the flight control actuators. The resulting architecture weighs 3.1 kg less than the reference architecture.

Literatures reviewed above were the most relevant ones related to automated IMA architecture design and optimisation. The scope of each approach, however, is limited to a certain aspect. The baselines for all of them include problem formulation, constraint definitions, and solving techniques like IP, BIP, and MIP. Nevertheless, examples can be found in other industries as well.

2.4.1 Architecture Design and Optimisation in Other Industries

Similar topics from other industries like space and satellite systems are also popular. Since determining the optimal placement of avionics boxes on the spacecraft is a difficult task which is normally performed manually, Jackson and Norgard developed a tool to optimise avionics box placement [49]. This has been defined as a multi-objective optimisation problem for optimising the placement of avionics boxes on the exterior panels of a spacecraft in that multiple cost functions and constraints must be satisfied. The objectives are to minimize the amount of harness wiring and the length of RF cable runs while keeping the thermal loading and mass distribution across panel in an acceptable limits. The input information into the problem are avionics boxes dimension, masses, power dissipation, mounting location of fixed components, connectivity between boxes and interconnection priority weighting. Further, a Simulated Annealing (SA) algorithm is proposed to solve the problem. SA reaches the optimal solution by perturbing the current best solution to explore more of the solution space.

Fabiano and his colleagues also developed a tool for the spacecraft equipment layout since the decision-making of where to place electrical equipment is a difficult task while taking into account many factors like position of mass centre, moment of inertia, heat dissipation, and EMI and integration issues at the same time [50]. This task is usually done by a group of system engineers which takes time and as soon as a feasible solution is found, it becomes the baseline. In other words, all possibilities and layouts are not explored completely and the solution is not necessarily the optimal one. In this paper, a tool based on Excel is developed that employed optimisation techniques which provides the system engineering team and easier way to explore the layout conceptual design space. The algorithm used is M-GEO which is a multi-objective algorithm. Finally, decision making criteria are used to select solutions from the Pareto frontier. Works with similar concepts can be found in literatures [51] [52].

Annighofer also developed a formal mathematical-based model for the European space launcher ARIANE 5 launcher to upgrade its avionics systems as well as future launchers [53] [54]. Since ARIANE 5 launcher was first developed in early 90s i.e. it has a federated avionics architecture, its avionics systems may not be necessarily optimal regarding to the current technological advancement and requirements. The proposed architecture has used the IMA architecture concept. Further, the avionics architecture design is formulized as a binary programming which include function allocation to IMA devices and mapping the device into installation location while satisfying a number of constraints like power, mass, segregation and power. Twenty installation locations are available for IMA devices and 158 locations for peripherals (sensors and actuators). MIP and COTS solvers like CPLEX and GUROBI are used to solve the optimisation problem. The results led to a huge optimisation in mass, power consumption and reduction of cable length.

Similar concept from automotive industry is a commercial tool called PREEVision. It is a model based design of electrical and electronic (E/E) architecture for automotive domain [55]. The main goal of the tool is to provide a component database which supports automatic design. The model consists of functional, component, and installation layer as well as the evaluation of requirements. The components are automobiles' Electronic Control Unit (ECU) and data buses like CAN. The functions are then assigned to ECU and the ECU are assigned to installation locations. Moreover, the function signals that are assigned to wires or buses can be automatically routed. The complete model and tool enables different instantiations of the same architecture.

Hardung in his PhD dissertation developed a framework for the optimisation of the allocation of functions in automotive networks [56]. Multiple objectives were defined including costs and busload for optimisation while a number of constraints like memory consumption, I/O-pins consumption, timer consumption and bandwidth availability are met within acceptable limits. In other words, the hardware is assumed fixed, and the software components are to be allocated to ECUs. A database model in SQL is built to store all relevant information including ECUs' technical specifications like the weight of ECU and wiring harness, cables, cost, and their functional links. The model is then implemented for a central door locking and keyless entry where the details of the architecture is shown which include network topology, resources, costs, weight, busload as well as supplier complexity. The supplier complexity is defined by the minimum number of suppliers involved in the development of the ECU. This is transferred into a set-covering problem. The optimisation algorithm proposed is Ant Colony Optimisation (ACO) which is also compared to Evolutionary Algorithm (EA) and is shown that ACO is faster than EV at the beginning. However, after a while, ACO cannot find better solutions. While EV is slow at the beginning, it is continuously improving in later phases. Due to better performance, at the end the ACO is selected as the final approach.

2.4.2 Avionics Architecture Design Techniques

What is meant by modelling here is a system model that represents a system, in particular, aircraft systems like fuel system and/or avionics system which is made up of hardware and software as well as human interaction. The model can be a model of models known as meta-model [57]. An aircraft model, for instance, is composed of several models which is still a system model although at a higher complexity level. At conceptual level, Avionics system, particularly, defines the functions and the relationship among them. In further detail designs, the functions ' interaction, flow and physical equation can also be added to predict performance and dynamic behavior of a model. One engineering approach to model aircraft systems is called Model Based System Engineering (MBSE) which use a central model to capture requirements, architecture and design to help system architects in different levels of aircraft system design [58].

There are a number of modelling techniques in aircraft system and avionics system design. Each of which is appropriate for a set of problems. The main goal of modelling is to provide a tool for concept generation, automatic architecture optimisation and trade-off studies of generated architectures. Architecture modelling determines the system boundary, its components and subsystems as well as its interfaces. Architecture modelling is also known as system architecting. The main task of system architecting is to provide a balance between requirements and actual products. Different various design techniques are being used to help systems developers to understand, decompose, analyze, and document the problems including Axiomatic Design, Design Structure Matrix (DSM) and Function/Means tree.

Axiomatic and DSM are design matrices that can systematically analyze and document the relations in the design process [59]. The axiomatic method is based on two rules/axioms which use a matrix to visualize, analyze and transform the customer requirements to functional requirements, design parameters and process variables. Figure 2-2 shows the process in axiomatic design. It includes mapping customer requirements into functional requirements (FRs). To satisfy the FRs, design parameters are defined in the physical domain. Finally, process variables (PVs) are defined to satisfy design parameters. This mapping process is shown linearly in a design matrix.

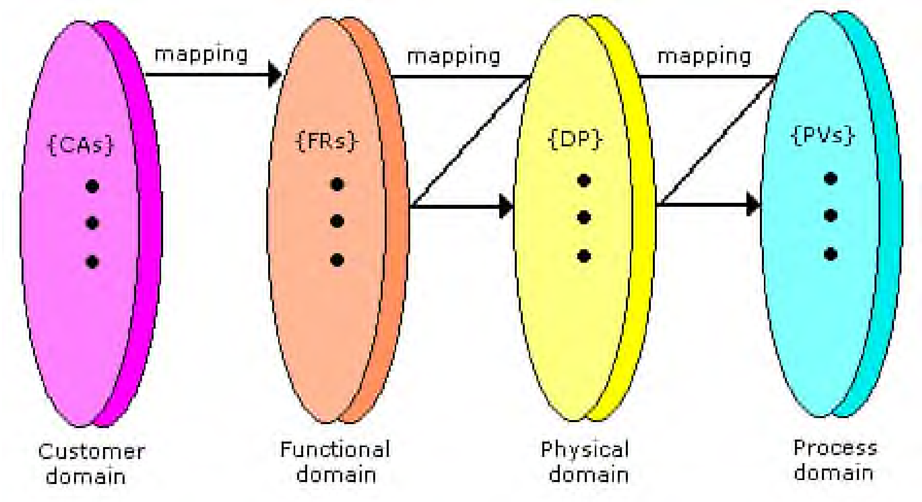


Figure 2-2 The Fundamental Concept of Axiomatic Design

DSM is another tool and technique in system engineering for system decomposition and integration. It provides a visual representation of a complex system as well as design parameters with their interdependencies and flow information which support innovative solutions to decomposition and integration problem. Any changes that may affect the system can also be analyzed by DSM i.e. if a component needs to be changed, all the dependencies and interfaces can be quickly identified. House of Quality (HoQ) is another matrix-based method that combines the analysis of functional decomposition to components dependencies. More matrix-based method can be found in [60].

In large scale problem, matrix-based methods are difficult and time-consuming to handle, but function/means tree is more suitable. This method is also used for functional decomposition, allocation to means i.e. components to fulfil the requirement and concept generation. The function/means tree has a hierarchical structures functions and means on various levels. It can also be used to represent alternative solutions, from which a final candidate can be determined [61]. Since integrated modular avionics architecture has many software applications, and hardware as well as dependencies among their, this allocation-based design technique is more appropriate. Figure 2-3 depicts a generalised inheritance mechanism which is called generic object inheritance enabling a quick reuse and modification of conceptual product and/or solution models at any level in their hierarchical break down structures [62].

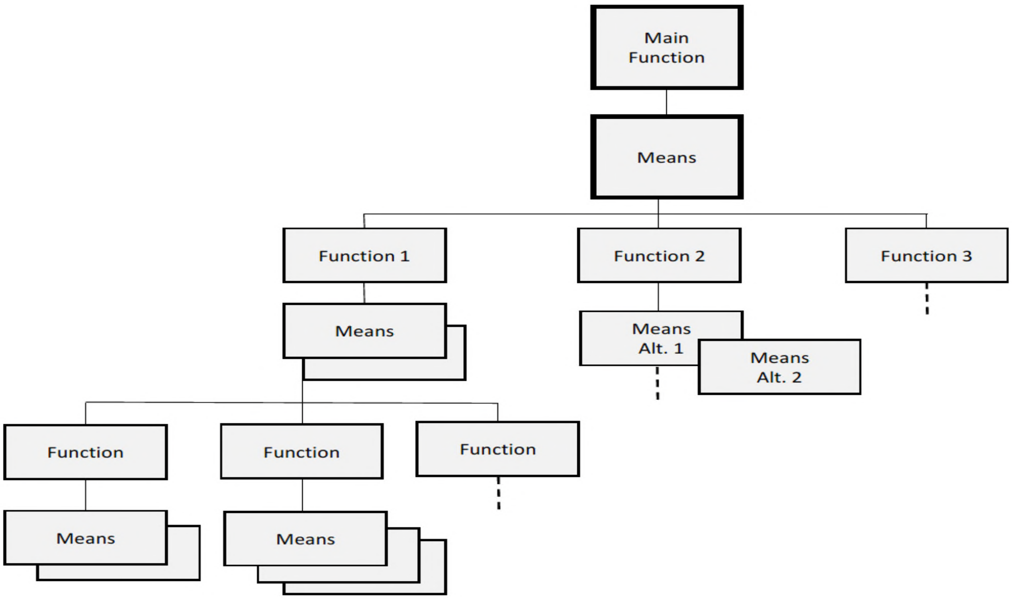


Figure 2-3 Function/Means Tree

In this research, this technique is used for avionics functional breakdown and allocation of avionics LRUs for each functions. Each function, in some cases, a combination of two or three functions is allocated by LRUs (means/solutions) from different venders. This has created a set of data which is used for investigating various architecture as well as trade-off studies. The best allocation is happened based on physical and operational requirements needed. In other words, the avionics functions are assigned to components (LRUs) and components are assigned to installation location.

2.5 Comparison of Modelling Approaches

All the above approaches can be utilized to express IMA architecture in a certain scope. All of them are driven by mathematical programming which can formulate the IMA design problem and speed up the development process automatically and/or semi-automatically through simulation or formal verification. What is common among models is their formulation i.e. the overall definition of the problem mathematically. They includes decision variables, objective functions, and constraints. The decision variables are defined in various forms like Integer Programming (IP), Binary Programming (BP) and Mixed Integer Programming (MIP). Also, the objective functions defined are different, for instance in [37] and [41] the objectives are related to improving reliability and safety of the architecture whereas in [38] the objective is to minimize the hardware (processor) and the required network bandwidth. The main objective functions in [44] and [45] are to minimize weight and costs.

Moreover, models in [63], [36], [64], [43], [39] and [42] are capable of expressing aircraft-level architecture which is within the scope of this research as well. The other issues are related to the separation of software and hardware as well as the level of automation. Software mapping and/or hardware mapping alone cannot express the complete model of an architecture. Automatic design is only addressed in [44] and [36]. [36] Mainly focuses on finding redundancy structures, and lacks resources for installation location. Further, the types and the number of constraints in different models vary. A thorough modelling of IMA architecture which proposes various cost functions and constraints can be found in a patent by Airbus [65] in which constraints are formalized by a set of linear inequalities. However, the most common constraints studied in literatures can be classified as peripheral, segregation, latency, power and installation location constraints.

In conclusion, linear programming which is a widely used technique to express real-world problems into mathematical forms as used in literatures is selected in this research as well. The problem of IMA architecture optimisation is defined as an assignment problem in that it is to assign the best avionics LRUs/LRMs from database to the proposed integrated architectures. A set of costs functions are defined and constraints are expressed in inequalities forms. The decision variables are defined as a binary variables. The contribution of this research into modelling is adding the volume and weight constraints to the architecture as well as introducing the operational capability as a new cost functions.

2.6 Comparison of Optimisation Methods

Linear programming (LP) problems, in general, can be solved by many algorithms. However, in IMA avionics architecture as well as many other aircraft systems architectures, the variables usually defined as either Integer Programming (IP) and/or Binary Programming (BP). IP problems are not easy to solve and are known as NP-hard Discrete Combinatorial Optimisation problems. In other words, they cannot be solved in polynomial time due to the vastness of the solution space [66], [67]. IP problems can be solved by three different algorithms:

- The exact algorithms which guarantee optimal solutions, nevertheless they may need a huge number of iterations. They are branch-and-Cut, branch-and-bound, dynamic programming algorithms, Boolean satisfiability and decomposition.
- The heuristic algorithms which guarantee sub-optimal solutions, but the quality is not guaranteed. While the running time may be polynomial in some cases, they may find a good solution fast. They are like greedy algorithms, local search, metaheuristics (PSO), Tabu search and simulated annealing.
- The approximation algorithms which provide suboptimal solutions in polynomial time and the sub-optimality has a bound. They are like linear programming retaliation and Lagrangian relaxation.

The exact methods are usually suitable for small scale problem, yet for large scale problems heuristics are developed. The assignment problem, in particular, which has been used to define software/hardware allocation, installation location and task assignment is a branch of LP and IP problems. One particular example of using Integer programming in aircraft avionics fleet upgrade optimisation can be found in [68]. Most of the other task assignment problems in IMA architecture e.g. [33], [34], [35], [38], [69], [70] are combined with time constraints for scheduling and network transmission. They all used heuristic methods like Evolutionary Algorithms (EV) including GA, NSGA-II, PSO and ACO for these discrete optimisation problems. There is no literature that has compared these methods in IMA architecture optimisation, however the performance of ACO and PSO are reported to be faster than the GA [56].

The other categories uses COTS solver like SAT-solver, CPLEX, GUROBI and GAMS. In [71], [72], [39], [44] and [45] COTS solver are used and results are compared to heuristic methods. In many cases, the COTS solver results are more accurate than heuristics.

In conclusion, in this thesis, the problem is defined as an assignment problem which is a combinatorial optimisation problem i.e. a very large feasible solutions exist. To solve this problem, the Branch and Bound algorithm which is a widely used algorithm for solving large scale NP-hard combinatorial optimisation problems is selected. Moreover, Particle Swarm Optimisation (PSO) algorithm is also implemented to compare the results using Weighted Sum Method (WSM) as well as solving the multi-objective optimisation problems for finding Pareto-Frontier.

2.7 Future Trends in Avionics Systems Integration

Since the avionics systems significantly contribute to aircraft costs and safety, IMA and DIMA architectural and technological improvements are constantly evolving and developing. The main drivers for avionics systems integration are improvement in costs, weight, communication technologies (data buses) as well as ever-increasing of avionics functions and their system integration complexity. Moreover, the development is triggered by newly emerging technologies like cyber-physical integration, deep learning and cloud computing.

2.7.1 Communication Technologies

Drivers for new communication technologies are ever-increasing demand for bandwidth requirements, improving reliability, faster response time, and saving via cost and weight reduction. The current main research topic is to replace older data bus technologies like CAN, A429, analogue, and discrete lines. The bus must be good enough for highly critical tasks in that it needs a high reliability and fast deterministic response times. Moreover, the required electronics and software must be sufficiently small and cost efficient to be integrated into peripherals. Thus, New Data buses like FlexRay, TTEthernet [73], AFDX are investigated. Besides, transmitting signals through power supply wires (DC-BUS) or wireless could play a significant role in wiring reduction.

FlexRay: is a bus protocol for deterministic time synchronous and asynchronous data transmission with low latency developed in the automotive industry [74]. Its 10 Mb/s bandwidth, robustness, error correction, and redundancy concepts make it interesting for a new data bus alternative for avionics applications. It is currently tested in experimental flight test platforms in general aviation aircraft [75].

Power lines cane be utilized for data transmission, a so called DC-Bus. It is to utilize direct current lines for a DC-Bus network. It is encoded as a high-frequency voltage modulation and superposed with the constant voltage. DC-Bus reduces wiring significantly. Nonetheless, the bandwidth currently is limited to approximately 112 Kbps, and easily affected by interference on the power line [76].

Wireless communication requires no cables for data transmission. Therefore, wireless communication on board of an aircraft is an active research area in avionics domain. Current disadvantages are needed power supply, complex and expensive hardware, as well as the safety and security concerns. Its applications are now limited to monitoring, service, maintenance, and entertainment domain. For example Wireless Avionics Intra-Communication (WAIC) technology installation in cabin systems can significantly reduce the weight [77].

2.7.2 Reconfigurable Plug-&-Fly Avionics

Reconfiguration is the change of the current configuration of IMA/DIMA architectures in terms of function distribution, function availability, and communication channels. It is expected to increase availability which results in reduction of unscheduled maintenance events. Reconfiguration requires spare hardware taking over in case of the original hardware fails. For safety reasons, one or more identical copies of redundant system parts exist, which are either active, in hot-standby or passive. Consequently, a further increase of reliability and availability is predicted by enabling reconfiguration on IMA/DIMA platform.

The standardisation allows less spare resources than redundancy is needed. In other words, in case of failure, the spare resources can dynamically be assigned to software from the failed IMA module to restore a safe configuration. Moreover, reconfiguration could also be utilised for a flight phase dependent adaptation of the avionics system. For instance, more resources for target tracking in case of a fighter involved in air combat [78].

Two main barriers for reconfiguration are technology and certification. Every reconfigured configuration needs to be certified. In particular, automatic and algorithm based reconfigurations are difficult as they require the highest level of certification for decision making algorithms and routines changing the system while running. This is currently infeasible. The only configuration accepted is related to pre-qualified configurations [79]. Eventually, the aim of reconfiguration and configuration research is to have a configuration free avionics system [80], which manages resource allocation and communication, and redundancies dynamically depending on the hardware status and software requirements. Currently, an ongoing research is undertaking in Stuttgart University, Germany known as “Plug and fly” [81]. The aim is to develop an advanced automating systems functions, and getting safety, qualification, integration and configuration which is shown in figure (2-4).

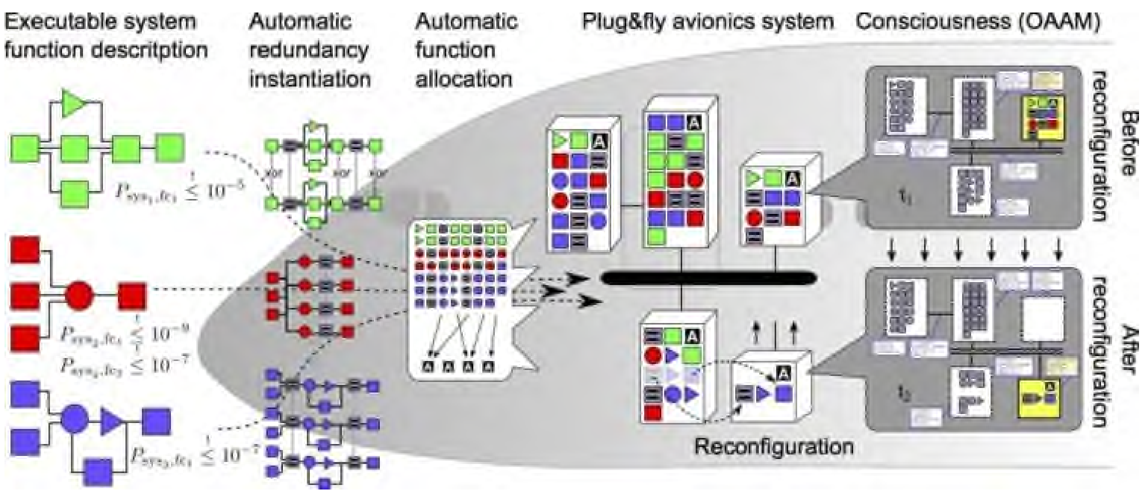


Figure 2-4 Concept of a Plug & Fly Avionics Systems

2.7.3 Intelligent Cyber-physical Avionics Integration

The future of aircraft avionics architecture will heavily depend on the advancement of information and network technologies [82]. The available smart sensors, data concentrator and actuators make a connection between the physical and cyber network. In particular, cyber-physical integration design will provide a huge applications for the physical world including avionics systems health monitoring and prognostics as well as monitoring any other concurrent events [83]. These all need a data-driven approaches for measuring aircraft dynamic behaviour, parameters and the interactions between various components in avionics architecture. Some new developed algorithms and techniques like machine learning and deep learning can help build models and learn the relation of components for predicting the MTTR and the failure rate precisely [84].

2.7.4 Avionics Cloud Computing

The flexibility and sharing configuration of resources in cloud computing technology can be utilised into avionics systems [85]. This can happen via Virtual Machine (VM) management which bring about many advantages. Firstly, avionics resources can be virtualised with arbitrary configuration to enhance the resource utilisation among various subsystems. Secondly, the avionics system can host critical as well as non-critical functions more robustly by virtualisation technology. Consequently, cloud computing is to become a technique to design critical systems with integration of any other non-critical functions. However, the function mapping will become more complicated under avionics cloud since the computing power, memory and storage that are virtualised need to be considered for mapping between Virtual Machines and hardware components [86]. Issues like this must be taken into account while designing avionics cloud in future as it has a huge potential to improve the efficiency of the avionics architectures.

2.7.5 New Hardware Concepts and Technologies

The hardware in IMA/DIMA architecture follows the natural evolution of electronics hardware like the enhancing processor power and memory density.

Moreover, new cabinets, dual-lane modules and smart sensors as well as miniaturization of electronics and sensors reduce the volume, weight and cost of avionics systems. In cabinets housing, power supply, cooling and EMI shielding all happens once and they provide calculation, and I/O resources which cannot operate stand-alone. Dual-lane modules are comprised of identical function in the same housing that can be used for aircraft systems which are duplex. Both lanes provide independent computing and I/O resources. Power supply and ADCN connection exist twice and are separated. Finally, fast inter-lane communication within the dual-lane module enables redundant fast control loops and time critical monitors. Also, smart sensors and actuators combined with a support circuit are used to pre-process sensor signals and carry out local control in order to support an advanced bus protocols. This provides configuration capability that can help to reduce wiring and controller hardware.

2.7.6 Future Avionics Architectural Challenges

The most important avionics architectural challenges ahead is related to the concept of urban air mobility. It is all about an on-demand and automated passenger or cargo-carrying services, typically flown without a pilot. In other words, the integration of UAVs into civilian airspaces. The current technologies may not be suitable to meet these challenges. It does require a collaboration to work through issues with noise impact, cybersecurity and UAS integration with ATM. The key technologies that are need to be developed and/or improved are

- Detect and avoid sensor systems (radar, ADS-B, TCAS)
- Robust high speed data links
- Connected flight management system
- Connected weather information
- Advanced data analytics

Some other solutions for these future architectural challenges are:

Miniaturization of avionics i.e. to develop System on Chips (SoCs) and/or System on Modules (SoMs) in order to enable and provide all the avionics capabilities in limited space, in particular, for small and unmanned aircraft.

Investigating new architecture for Avionics LRUs and avionics networks to provide high bandwidth network and meet latency requirements by means of new communications technologies [87].

3 Avionics Integration Architecture: Methodology

3.1 Introduction

The established framework and method in this thesis is realised based on industrial processes and Cranfield university's internal projects in Aerospace Vehicle Design (AVD) group as well as Avionics Network and architecture lecture notes [88] and other aircraft flight deck and systems documents [89], [90], [91], [92], [93], [94], [95]. Figure 3-1represents an overview of the framework.

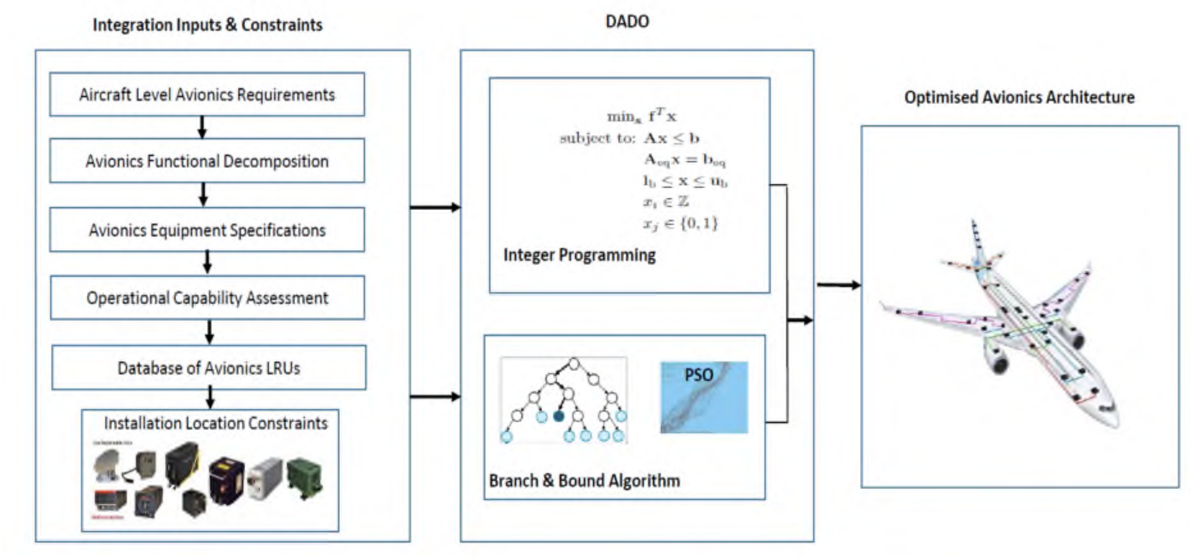


Figure 3-1 Avionics Integration Optimisation Framework

Based on Aircraft Level avionics requirements the avionics system integration and architecting starts from a top level functional decomposition to provide the framework for the avionics systems design and integration. This leads to the equipment specifications with every requirement being satisfied by the equipment performance parameters and/or operational capabilities. In other words, for each avionics function at least three avionics LRUs are investigated from various vendors. The operational capability of each LRU is evaluated separately against a set of criteria using MCDM (Multi Criteria Decision Making) method, SAW (Simple Additive Weighting). Further all the technical specifications of Avionics LRUs as well as their manufacturers are recorded in a database. Finally, the proposed initial system architecture for Automatic Flight Control System (AFCS) architecture and the whole avionics system architecture are modelled using mathematical programming. The problem is defined as an assignment problem i.e. it is to assign the best avionics LRUs from database while satisfying a number of design constraints including mass and volume in installation locations.

The optimisation problem is then solved by applying branch and bound algorithm for single objective cost functions and PSO for multi-objective cost functions. The cost functions defined here in this thesis include minimising the overall weight of the architecture, minimising volume, minimising power consumption, maximizing the reliability and operational capability as well as trade-offs studied between these cost functions. The proposed method is not developed for a specific aircraft type, and it is meant to be a general method that can be used for any aircraft type and/or architecture. However, the proposed avionics architecture is similar to a short to medium haul civil aircraft architecture which is used here as a case study. The proposed architectures have used the concept of IMA architecture although they are not fully integrated.

3.2 Aircraft Level Avionics Requirements

Proper requirements are essential for creating an acceptable avionics system. The designed avionics system cannot perform as expected by customer unless the customer requirements as well as other stakeholders' requirements and related regulations and standards are well documented and understood by designers. The major driving requirements are from safety, mission, cost and certification perspectives. In other words, a variety of technical and functional requirements for on-board avionics equipment can be captured which are comprised of the following generic documents:

- Airworthiness requirements including the requirements related to aircraft equipment and systems as well as international airworthiness standards like CS-25, FAR-25, CAR-525 etc.
- Functional requirements for the on-board aircraft equipment
- Safety requirements
- Reliability requirements

- Avionics network and interface requirements
- Installation and environmental condition requirements

In this research, the focus is on the CNS/ATM functions required to be on-board the aircraft. In other words, the requirements captured are mainly from regulatory documents.

The functions that the avionics systems are expected to fulfil are shown at Figure 3-2 and they are meant to meet all CNS/ATM requirements as well as FANS-1/A. It should be noted that all the FANS and/or CNS/ATM requirements in this thesis are derived from an LRU perspective. The functional breakdown shown can be generally attributed to any civil aircraft.

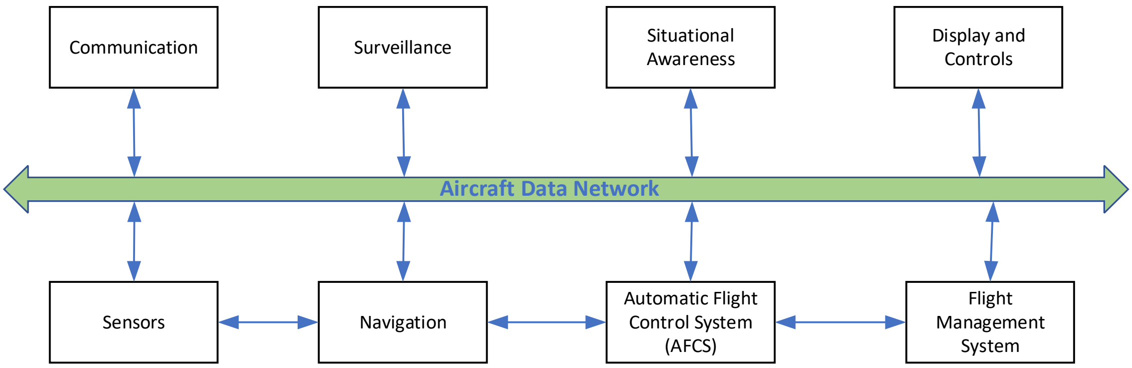


Figure 3-2 Avionics Datum Functional Architecture

The avionics requirements are concerned with the functional capability of avionics equipment on the aircraft. It is only after this document that avionics design team can justifiably determine an equipment list. Then, for each item in equipment list, equipment technical specifications can be prepared. The operational capability required within each function is outlined further in detail in avionics technical specifications and operational capability assessment phase. The following requirements are laid out from the operational requirement of civil aircraft [96], [97], [98], [99]. The specific requirements are related to Communication, Navigation, Surveillance (CNS) and Air Traffic Management (ATM) systems as well as data transmission. The proposed architecture is to meet basic CNS/ATM requirements as well as improving operational capabilities by implementing new technologies like HUD (Head-Up Display) and Enhanced Vision System (EVS) among others.

The Table 2 shows the capabilities that are required for the proposed avionics architectures, and further these capabilities are evaluated from an LRU perspective. In other words, some capabilities are embedded as software within a particular LRU e.g. RNP and RTA are loaded in FMS and/or MCDU. Moreover, the accuracy and performance of some of these capabilities are further classified that determine the operational capability of avionics systems at aircraft level. The definition and classification is taken from a range of aviation international rules and regulations including ICAO Doc 4444: PANS-ATM, ICAO Annex 2, 10 and 14 among others [100], [101], [102], [103]. The avionics systems are classified in three main groups including communication, navigation and surveillance.

The next column in table 2 is related to avionics capabilities. These capabilities are defined from an LRU perspective. However, in some cases, some avionics capabilities are loaded into an LRU as a software. Also, some LRUs are investigated that are capable of doing more than just one avionics function e.g. (T3CAS) which is an advanced integrated surveillance system featuring a Traffic Alert and Collision Avoidance System (TCAS), Terrain Awareness Warning System (TAWS), and Mode S transponder with Automatic Dependent Surveillance-Broadcast (ADS-B).

Table 2 Aircraft Level Avionics Systems Requiremnts

Avionics Systems	Avionics Capabilities	Explanations
Communication	8.33 kHz VHF	VHF transceiver voice channel spacing
	SATCOM	Airborne radio telephone communication via a satellite
	CPDLC	Controller Pilot Data Link Communications (CPDLC) is a means of communication between controller and pilot, using data link for ATC communications
	VDL M2	VHF Data Link mode 2 is a means of communication between aircraft and ground stations
Navigation	GBAS CAT II/III	Ground Based Augmentation System CAT I is a landing system capability known as GLS whose performance is equivalent to ILS CAT I (down to 200 ft). CAT II/III are under development
	SBAS APV CAT I/II	Satellite Based Augmentation System CAT I is a landing system capability whose performance is equivalent to LPV minima (down to 200 ft.)
	RTA (FMS)	Required Time of Arrival enables the pilot to define a waypoint with a specific arrival time plus or minus 30 seconds. If the aircraft cannot meet the 30 second requirement, the flight crew will be notified
	RVSM	Reduced Vertical Separation Minimum for above FL290 from 2,000 feet to 1,000 feet
	Autopilot/Flight Director	Autopilot software, which is integrated with the navigation systems, is capable of providing control of the aircraft throughout each phase of flight
	RNAV 10	RNAV 10, which is designated as RNP 10 in the ICAO's PBN Manual, is an RNAV specification for oceanic and remote continental navigation applications
	RNAV 5 (B-RNAV)	RNAV 5, also referred to as Basic Area Navigation (B-RNAV), has been in use in Europe since 1998 and is mandated for aircraft using higher level airspace. It requires a minimum navigational accuracy of +/- 5nm for 95% of the time and is not approved for use below MSA.
	RNAV 2	RNAV 2 supports navigation in en-route continental airspace in the United States
	RNAV 1 (P-RNAV)	RNAV 1 is the RNAV specification for Precision Area Navigation (P-RNAV). It requires a minimum navigational accuracy of +/- 1nm for 95% of the time.
	RNP 4	RNP 4 is for oceanic and remote continental navigation applications
	RNP 2	RNP 2 is for en route oceanic remote and en route continental navigation applications
	RNP 1	RNP 1 is for arrival and initial, intermediate and missed approach as well as departure navigation applications
	Advanced RNP	Advanced RNP is for navigation in all phases of flight
	RNP APCH	RNP APCH and RNP AR (authorization required) APCH are for navigation applications during the approach phase of flight
	RNP AR APCH	RNP APCH and RNP AR (authorization required) APCH are for navigation applications during the approach phase of flight
	RNP 0.3	RNP 0.3 is for the en-route continental, the arrival, the departure and the approach (excluding final approach) phases of flight and is specific to helicopter operations
Surveillance	ADS-B	A means by which aircraft, aerodrome vehicles and other objects can automatically transmit and/or receive data such as identification, position and additional data, as appropriate, in a broadcast mode via a data link
	Mode S Transponder	Mode S is a Secondary Surveillance Radar (SSR) process that allows selective interrogation of aircraft according to the unique 24-bit address assigned to each aircraft. Recent developments have enhanced the value of Mode S by introducing Mode S EHS (Enhanced Surveillance)
	TCAS/TCAS II/TCAS	Currently, TCAS II version 7.1 is mandated in European airspace.

3.3 Avionics System Architecture

3.3.1 Avionics Functional Decomposition

The functional decomposition is a technique used for describing an avionics system in very general terms. The avionics requirements are divided into a set of “Top Level” functions. The process seems to be straight-forward, however, there is no clear-cut way of doing this. Figure 3-3 shows the functional breakdown adopted as strength point referred to as the datum functional architecture from an LRU perspective. This decomposition evolves from the operational requirements of the aircraft which determines the functions needed.

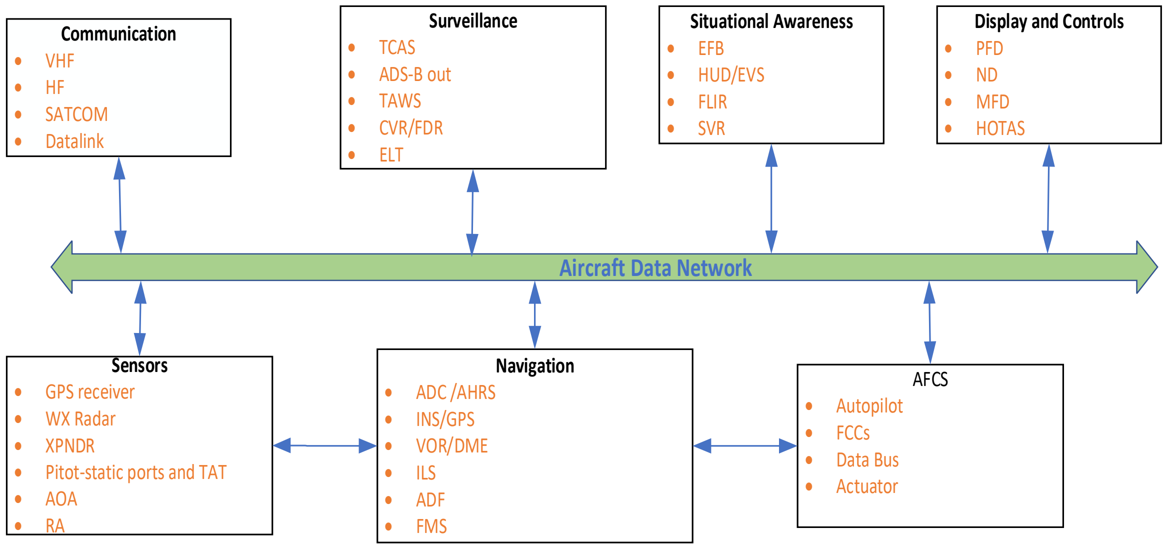


Figure 3-3 Avionics Functional Decomposition from an LRU Perspective

Based on this functional decomposition which is derived from aircraft level avionics requirements, equipment list, in this case LRU, is prepared. For each avionics LRU, at least three different ones are investigated from various venders. The recorded LRUs are different in their physical specifications as well as their operational capabilities. This then arise the problem of choosing the best LRU in order to optimise architecture in some criteria like weight and also improve the operational capability. The initial system architecture based of these equipment list is drawn below.

3.3.2 Automatic Flight Control System (AFCS) Architecture

The avionics system architecting is the determination of the necessary interconnections and functional interrelationships between the components of an avionics system which is a complex task. Typically, this task is shared between the airframe manufacturer and the avionics supplier to ensure that all relevant factors and implications are taken into account. The initial system architecture is adopted directly from the integrated datum functional architecture and functional decomposition. Figure (3-4) illustrates the proposed system architecture for AFCS.

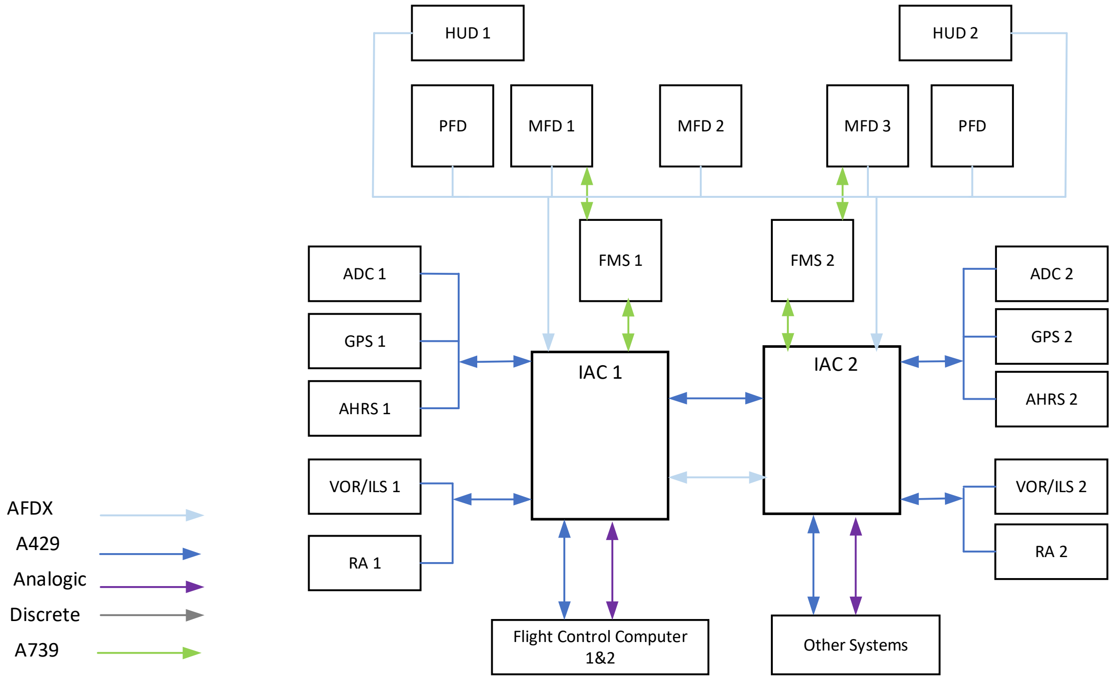


Figure 3-4 Automatic Flight Control System Architecture

The proposed AFCS guarantees three functions including Autopilot (AP), Flight Director (FD) and Yaw Damper (YD). The main components are two Integrated Avionics Cabinets (IACs) which host Automatic Fight Control Application (AFCA) and exchange data with two ADCs, two AHRSs, two FCCs and seven Display units. The navigation sensors are VOR/ILS, GPS receivers and RAs. This architecture in modelling is referred as the small scale architecture i.e. each avionics function and/or LRU will be defined as an avionics Node in a network and/or architecture. Further, the AFDX/A664 data bus is selected as the main data transmission system for the proposed architecture. The idea of Integrated Modular Avionics (IMA) has also been implemented in two Integrated Avionics Cabinets (IACs). The IACs supplies resources for avionics applications including memory, I/O and computations which are shared.

The Integrated Avionics Cabinets hosts the following avionics applications:

- Flight Warning Application (FWA)
- Auto-Flight Control System (AFCS)
- Centralised Maintenance Application (CMA)
- Data Concentration Application (DCA)
- Switch Module Application (for AFDX)

Moreover, each IAC composed of Line Replaceable Modules (LRMs) including CPM (Core Processing Module), different I/O types and Switch Module for AFDX. CPM can host avionics applications and provide the connection to Avionics Data Network (ADN). The various I/O types provide connections for conventional avionics that cannot be directly connected to ADN. One IAC interfaces with other aircraft systems by different means of communications like discrete, analogue and A429. The IMA information is shown in three main systems including two PFDs, three MFDs, one of which is for engine and warning display. This architecture further extended to whole avionics system architecture as below.

3.3.3 Avionics System Architecture (The Large Scale)

Based on functional decomposition and the AFCS architecture, the architecture is extended to whole avionics systems including navigation, communication and indicating and recording systems as well as terrain and avoidance systems. Figure 3-5 illustrates the proposed avionics system architecture.

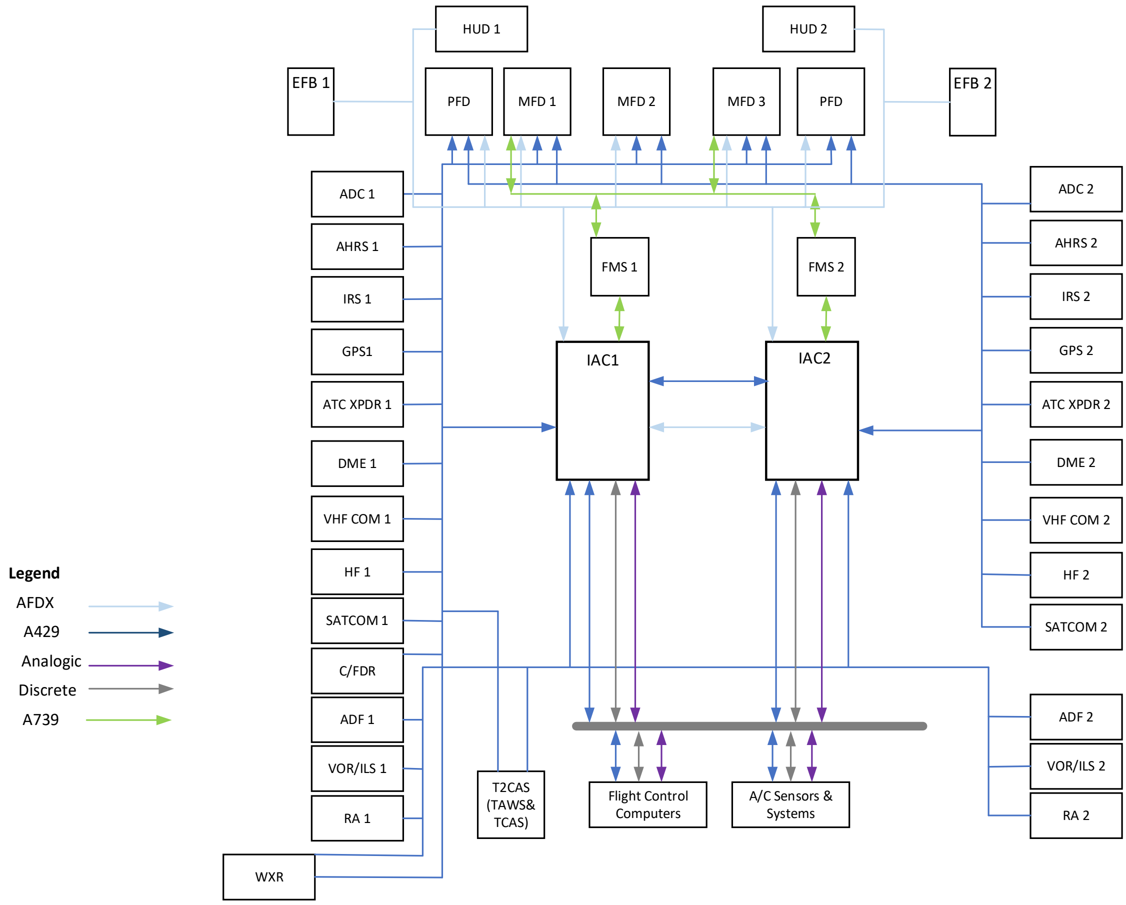


Figure 3-5 Avionics System Architecture

The IMA part is the same as explained in AFCS architecture. Some functional integrations are applied in that two or three functions can be done by an LRU. For instances, TCAS, Transponder Mode S, and ADS-B are integrated in one LRU called T3CAS. Furthermore, Cockpit Voice Recorder (CVR) and Flight Data Recorder (FDR) are also integrated in one LRU. This means that LRUs with these capabilities are found while investigating technologies for avionics functions. This architecture in modelling section is referred as the large scale architecture. The same as the small scale, the avionics functions and/or LRUs will be defined as avionics nodes in a network and/or architecture.

3.4 Avionics Equipment List and technical Specifications

In order to optimize the proposed architectures and trade-offs studies, it is needed to quantify avionics LRUs physical parameters as well as their performances and operational capabilities. Based on the proposed AFCS and avionics system architectures, the avionics equipment needed are as follows. For AFCS architecture, Air Data Computer (ADC), Attitude and Heading Reference System (AHRS), GPS receiver, VOR/ILS receiver, Radio Altimeter (RA), Flight Control Computer (FCC), Head-Up Display (HUD), Primary Flight Display (PFD), Multi-Function Display (MFD), Flight Management System (FMS), and two Integrated Avionics Cabinets (IACs). These eleven avionics LRUs are used for the small scale architecture optimisation.

Furthermore, for the large scale architecture, some other LRUs are also added including Weather Radar (WXR), Traffic/Terrain/Transponder Collision Avoidance System (T3CAS), Electronic Flight Bag (EFB), Cockpit and Flight Data Recorder (C/FDR), SATCOM, and VHF. The LRUs which are 17 are considered as the large scale architecture optimisation problem. The technical specifications and performance description of the avionics LRUs are taken from Jane's Avionics [104] as well as their companies' data sheet. The manufacturers are chosen according to the Flight International civil avionics directory. The physical specifications of avionics LRUs recorded are mass, power consumption, volume, and MTBF which are defined in section 1.3.2. The performance and/or the operational capability of each LRU is evaluated separately to be quantified.

3.5 Avionics LRUs Operational Capability Assessment

3.5.1 Introduction

The operational capability of each avionics LRU is defined as the performance parameters and capabilities that an LRU can perform. Since each avionics LRU does a different function/functions that provides different capability, they need to be evaluated separately against a set of criteria related to their functions. It is very challenging to evaluate new technologies, in general, and avionics LRU, in particular as a number of criteria involved in each assessment. Generally, technology assessment steps are technology identification, selection and evaluation. The important task in technology selection and evaluation is to establish a set of evaluation criteria. Many big companies and departments have their own method to assess technology readiness level and selection including DOE, DOD and NASA [105], [106], [107]. It is also common to develop a tool to do technology assessment, a good example of this can be found in [108] literature from Georgia institute of technology. The main focus in these reports is how to identify the technology readiness level, however in this research the author used existing technologies (LRUs) for integration and further optimisation of avionics architecture. Therefore, technology readiness level (TRL) is not in the scope of this research although other established processes like identification and scoping of new technologies were applied according to these very established framework. Particularly, the selection process is the main part which in this research as it is to select the LRUs with the maximum operational capabilities.

Since in most cases technology evaluation and selection criteria are more than one, a solution is to use Multi-Criteria Decision Making (MCDM) methods [109] to handle this complexity. The methods help decision-makers to decide in the presence of multiple and/or conflicting criteria. What is common among these techniques are a set of technology alternatives, multiple decision criteria, the attitude of the decision makers in favour of one criteria over the other as well as the preference of the technology alternatives. MCDM techniques help decision makers to assess the overall performance of technology alternatives which will be further help the optimisation of design solutions [110].

3.5.2 An Overview of Multi-Criteria Decision Making Methods

The MCDM method is of the most common method is decision making problems. It has two branches including Multi-objective Decision Making (MODM) and Multi-attribute Decision Making (MADM) [111]. Nevertheless, MCDM and MADM are used interchangeably. MODC is used when the decision space is continuous like linear programming problems where there are more than one objective functions to be optimised [112]. On the other hand, MCDM/MADM deals with problems in which the decision space is discrete. Two associated terms that always appear in MCDM problems are “alternatives” and “criteria/attributes”. Alternatives show the various choices of technologies, for instance, to the decision makers. Attributes which are usually referred to as decision criteria show the various dimensions from which the alternatives can be evaluated. Moreover, most the MCDM methods require weights for their criteria as well as decision matrix in matrix format. The decision matrix element a_{ij} represents the performance of the alternative in which it is assessed in terms of the decision criterion.

Based on the type of the data which is used, MCDM methods can be classified as deterministic, stochastic and fuzzy methods. In this research, deterministic methods like TOPSIS, AHP, ELECTRE, and SAW are overviewed. The very common steps in any MCDM techniques are

1. Determining the related criteria and alternatives
2. Establishing a decision matrix by numerical measures to the relative importance of the criteria and to the impacts of the alternatives
3. Calculating the numerical values to settle a ranking for each alternative

The Analytic Hierarchy Process (AHP) break down a complicated MDCM problem into a system of hierarchies [113]. In the AHP, a matrix of a $m \times n$ dimension is structured where m is the number of alternatives and n is the number of criteria. In fact, it is made by using a relative importance of the alternatives compared to each criterion. The vector $(a_{i1}, a_{i2}, \dots, a_{in})$ for each i is the eigenvector of a $n \times n$ reciprocal matrix which determined by pairwise comparison of the impact of the m alternative on the i-th criterion. AHP mainly is used to determine the weighting factor for criteria.

The ELECTRE (Elimination and Choice Translating Reality) method begins with pairwise comparisons of alternatives under each criterion by using physical values noted as $g_i(A_j)$ and $g_i(A_k)$ of the alternatives A_j and A_k respectively [114]. Further, a threshold levels is introduced for the differences$g_i\left(A_j\right) - g_i(A_k)$. The decision maker then states that they have a weak/strict preferences for one of the two or they may not be able to express any preference relations. In addition, the decision maker is asked to assign weights to each criterion to show their relative importance. Eventually, the method produces a system of binary outranking relations among the alternatives which causes to yield a main leading alternatives. In other words, this method eliminate less favourable alternative and gives a clearer view of the alternatives preferences. The method is particularly suitable for problems that have a few criteria with a large number of alternatives.

The Technique for Order Preference by Similarity (TOPSIS) is based on selection of alternatives that have the shortest distance from the ideal solution and the farthest distance from the negative ideal solution in geometric sense [115]. Therefore, the Euclidean distance approach is used in this method to assess the relative closeness of the alternatives to the ideal solution. Consequently, the preference order of the alternatives are derived by a consecutive comparison of these relative distances. This method is appropriate where both negative and positive criteria play a significant role in decision making processes. The other method is Simple Additive Weighting (SAW) which is commonly used in many cases and is based on weighting average. It is described in detail in next section as it is selected for this research. It is to find a weighted sum of the performance on each alternative regarding their associated attributes. It is mostly suitable for making decisions which have a lot of criteria so that the method will ease the process of various decision-makings.

The MCDM techniques mentioned above have been widely used in aircraft design, air transportation systems and air traffic management as today societal, economic, environmental and operational requirements have to be addressed in aerospace and aviation industry [116]. TOPSIS has been used in technology selection in aircraft conceptual and preliminary design phase [117]. It has also been used to evaluate and rank the Taiwan’s five major airlines as well as fleet interoperability and low maintenance cost [118]. The other technique is Simple Additive Weighting (SAW) which has been used to rank aircraft alternatives against six criteria [119], [120]. SAW method uses a 10- point ratio scale to normalise the values of the criteria where the minimum value is given 0 and the maximum value is given 10. That’s why SAW and TOPSIS are very sensitive to normalization and weighting factor.

Analytic Hierarchy Process (AHP) technique is usually used to obtain weighting factor for the criteria in MCDM techniques [121]. The other method is called ELECTRE. One of its main advantages is that it consider uncertainty. However, its outcome is very difficult to be explained in layman’s term [122]. Fuzzy set theory, on the other hands, accept imprecise inputs and can handle insufficient inputs although it is very difficult to develop [123]. Goal Programming (GP) is capable of dealing with large scale problems and can produce infinite alternatives [124]. Nevertheless, for weighting criteria, it needs to be used in combination with other MCDM techniques.

In conclusion, SAW technique is selected in this thesis due to its ability to compensate among criteria and choose the most preferred alternatives which has the highest weighted criteria as well as having a simpler calculation process.

3.5.3 Simple Additive Weighting (SAW) Method

The SAW technique is a simple and widely used method for multi attribute decision problems. The method is based on the weighted average i.e. weighting factors $[w_1, w_2, \dots, w_n]$ are assigned to the criteria by the decision makers. The multi-criteria values with their weighting factors are summed into a single performance metric. SAW then selects the most preferred alternative A^* which has the maximum weighted outcome as it is represented in equation (3-1).

$$A^* = \left\{ A_i | max \sum_{j=1}^{n} w_j x_{ij} \right\}, \qquad \mathrm{i} = 1,2, \dots, \mathrm{m}, \mathrm{j} = 1,2, \dots, \mathrm{n} \tag{3-1}$$

The SAW method process consists three steps:

Step 1: building a Decision matrix for technology alternatives (avionics LRUs) regarding to objectives and the relevant criteria by using a similar approach to Saaty 1-9 scale [125] of pairwise comparison presented in Table 3.

Table 3 Scales for Technology Alternative Comparison

Intensity of Importance	Definition	Explanation
1	Equal importance	Two alternatives equally contribute to the objective
3	Moderate importance	Judgement slightly favoured one alternative over another
5	Strong importance	Judgement strongly favoured one alternative over another

7	Very strong	One alternative is strongly favoured over another
9	Extreme importance	The evidence of favouring one alternative over another is of the highest possible order

Step 2: Calculate the normalized decision matrix for positive criteria:

For positive criteria:

$$r_{ij} = \frac{x_{ij}}{\max\,(x_{ij})} \qquad \textbf{(3-2)}$$

For negative criteria:

$$r_{ij} = \frac{\min\,(x_{ij})}{x_{ij}} \qquad \textbf{(3-3)}$$

Note: in this research, all the criteria defined and considered are positive.

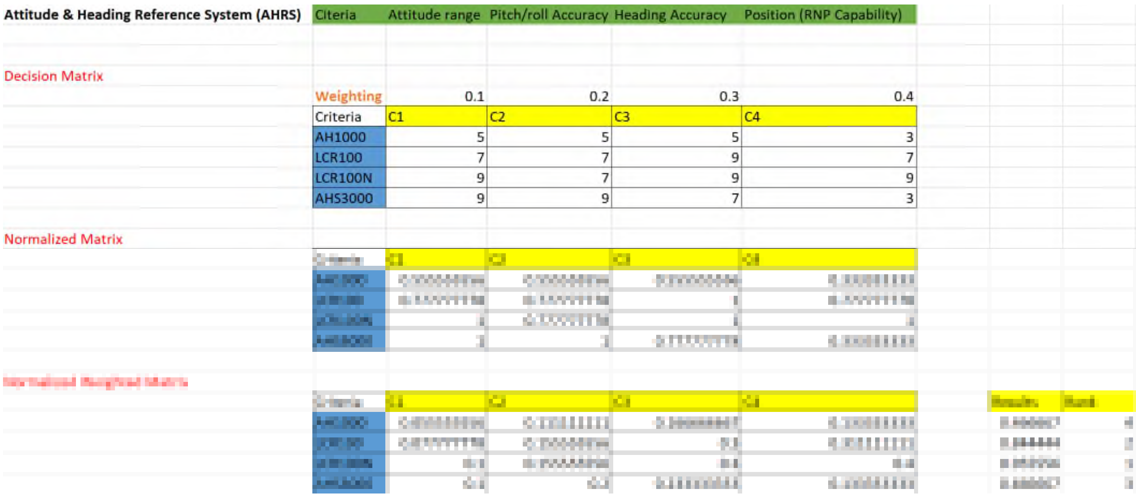
Step 3: calculate the normalized weighted matrix and evaluate each alternative by the following formula:

$$A_i = \sum_{j=1}^{n} w_j x_{ij} \qquad \textbf{(3-4)}$$

Where x_{ij} is the score of the $i$the alternative with respect to the $j$the criteria, and w_j is the weighted criteria.

3.5.4 Operational Capability Assessment

As mentioned before, for each avionics LRU, at least three different ones from various venders with different technical specifications are investigated. In this section, the operational capability of each LRU is evaluated by using SAW method. Since each avionics LRU does a particular function/functions, the criteria that are defined for evaluation are almost different for each of them. Sabatini also developed a set of criteria for avionics LRUs in [126]. However, he did not use any quantitative method to evaluate these criteria. For instance, Attitude and Heading Reference System (AHRS) is assessed against a set of criteria including attitude range, pitch/roll accuracy, heading accuracy and RNP capability. For AHRS, four different LRUs are recorded that can be distinguished by their ID as shown in Figure 3-6.



Attitude & Heading Reference System (AHRS)

Citeria	Attitude range	Pitch/roll Accuracy	Heading Accuracy	Position (RNP Capability)

Decision Matrix

Weighting	0.1	0.2	0.3	0.4
Criteria	C1	C2	C3	C4
AH1000	5	5	5	3
LCR100	7	7	9	7
LCR100N	9	7	9	9
AHS3000	9	9	7	3

Normalized Matrix

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	1	[illegible]
[illegible]	1	[illegible]	1	1
[illegible]	1	1	[illegible]	[illegible]

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Figure 3-6 AHRS Operational Capability Assessment

All the steps mentioned in SAW method are applied to AHRS operational capability assessment including establishing decision matrix, normalized matrix and normalised weighted matrix which is led to a ranked and preferred choices. Table 4 shows all the criteria defined for each avionics LRU and went on the same processes to be evaluated. The selected AHRS guarantees the following performance:

Attitude accuracy: 0.1 degree for straight and level flight, 0.2 degree in dynamic situations

Pitch/roll accuracy: 0.1 degree

Heading accuracy: 0.1 degree

Capability: maintaining RNP envelop in loss of satellite,

Table 4 Operational Capability Criteria Assessment of Avionics LRUs

Avionics LRU	Criteria Assessment				
ADC	PA accuracy	IAS Accuracy	temperature accuracy	Mach number accuracy	RVSM Compliant
VOR/ILS	Deviation Accuracy	Number of Channels	Capability (VOR/ILS/MB/DGPS)		
RA	Height Accuracy	Altitude Range	Attitude Range		
FMS	LNAV/VNAV Capability	RNP/RNAV Capability	RTA Capability	DataLink/CPDLC	
PFD	Display Area	Resolution	Viewing angle	Brightness	
MFD	Display Area	Resolution	Viewing angle	Brightness	
HUD	Resolution	Field of View	Approach Capability	Function Integration (EVS, SVS)	
FCC	CPU	Memory	Interfaces	Application Software	
IAC	CPU	Memory	Interfaces	Application Software	
WXR	Max Detection range	Azimuth Coverage	Elevation Coverage	Capabilities	
T3CAS	Bearing Accuracy	Range Capability	Operating Altitude	# of Functions Integrated	
EFB	Resolution	Viewing angle	CPU & Memory	Functions	
C/FDR	Recording Time	Impact shock	Penetration resistance	Deep See Pressure	
SATCOM	Data Rate	Service Coverage	Functions	Capability (FANS & ACARS)	
VHF	Pre-set channels	Channel change time	VDL Mode2 availability	Channel Spacing (25, 50, 8.33 Capability)	
AHRS	attitude range	pitch/roll accuracy	heading accuracy	RNP capability	
GPS Receiver	Altitude Accuracy	Velocity Accuracy	Position Accuracy	Approach Capability(SBAS/LPV or GBAS/GLS)	

Each assessment is done in a separate excel sheet similar to figure 3-6 for ARHS operational capability assessment. In this way, avionics LRUs operational capabilities are quantified as well as ranked. All the other operational capability assessments can be found in appendix B.

3.6 Avionics Equipment Database

Avionics LRUs for each function/functions with their Technical Specifications and operational capabilities as well as the manufacturers are recorded in an Excel Database. The technical specifications recorded are the ID of each LRU taken from their manufacturer identifications. The technical specifications recorded include physical characteristics of each LRU including mass, volume, and power consumption. The reliability of each LRU is also recorded by the Mean Time between Failure (MTBF) which is used for components reliability assessment. The operational capability of each LRU which is calculated separately in operational capability assessment is also recorded. Figure 3-7 shows the overall structure and some examples of Avionics LRUs technical specifications recorded in the database.

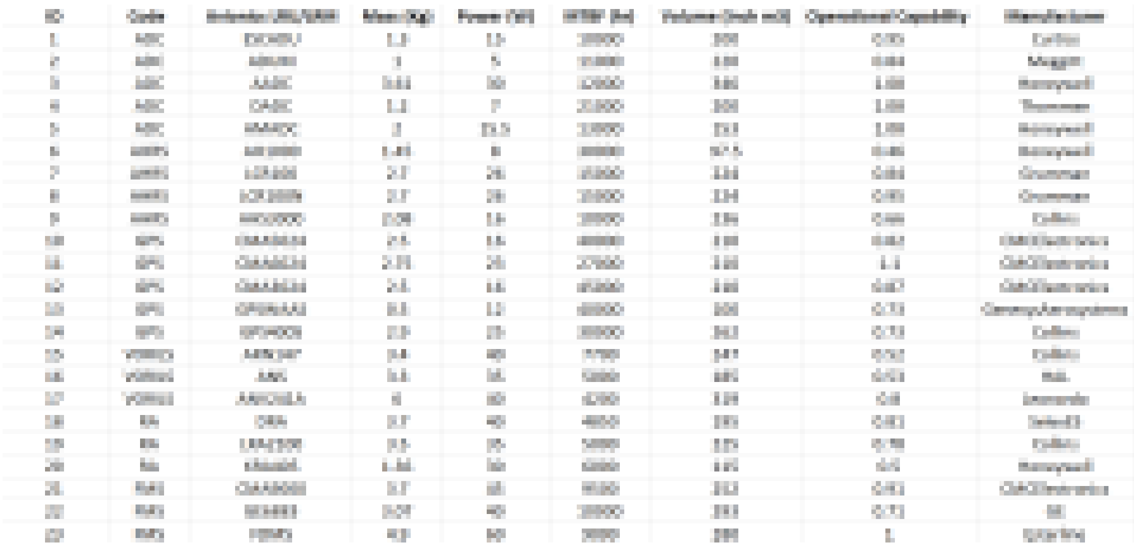


Figure 3-7 Avionics LRUs Database Structure

For the Automatic Flight Control System (the small scale) architecture, 44 avionics LRUs are recorded. For the whole avionics architecture (the large scale), 64 avionics LRUs are recorded. In both cases, the more LRUs can be added and extended. The specifications recorded purposefully as the objectives are to reduce the overall weight, volume, and power consumption of the proposed architectures as well as improving reliability and the operational capability.

3.7 Avionics Integration Optimisation Software Architecture

Based on the proposed methodology and framework for optimisation of IMA architecture a tool and/or software is developed. Figure (3-8) represents a top-level design of Avionics Integration Optimisation Software System (AIOSS). The tool has three main parts including a database of avionics LRUs, avionics integration architecture modelling and optimisation algorithms. The integration constraints and inputs to the software are the avionics LRUs mass, volume, power consumption, MTBF, and the operational capability. The proposed avionics architectures have some design constraints including mass and volume in the installation locations which are also implemented in the software. Furthermore, the modelled architectures are also implemented. For the single-objective optimisation the software is used branch-and-bound algorithm and for multi-objective optimisation, PSO is used although some single-objective cost functions are solved by both methods for comparison.

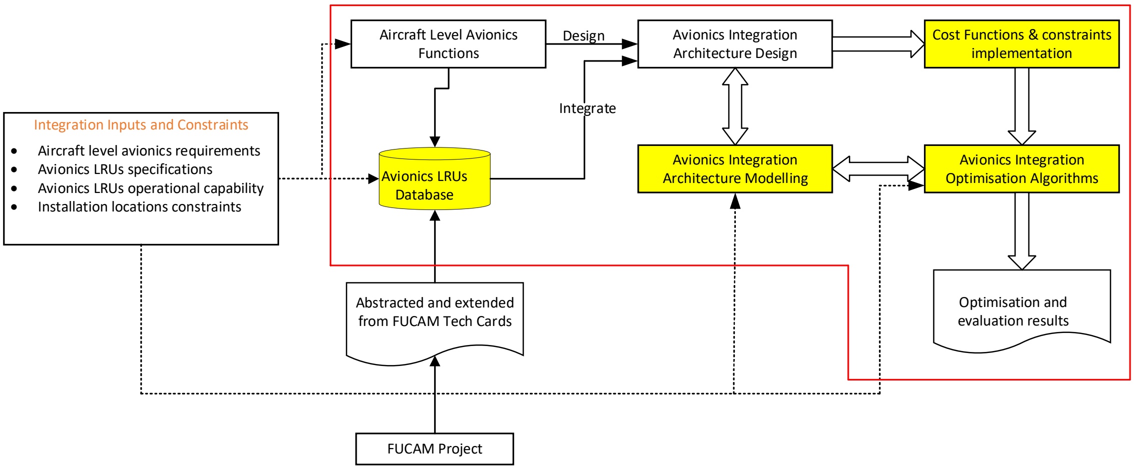


Figure 3-8 Top-level AIOSS Architecture

Finally, the avionics integration optimisation software provides a semi-automatic optimisation and evaluation of avionics architecture by reporting various possible architectures including minimum weight, minimum power consumption, minimum volume, maximum MTBF, and maximum operational capability. It also provides the trade-offs architectures for minimum weight and maximum operational capability as well as minimum weight and maximum MTBF.

4 Avionics Integration Modelling and Optimisation

4.1 Introduction

The mathematical programming and optimisation algorithms as discussed in chapter 2 have been used to model avionics architectures and provide some level of design automation in many cases. The nature of the optimisation problems in these area of studies is understood as combinatorial discrete optimisation. Due to the vastness of space solutions in these types of optimisation problems, the selection of problem formulation and solving algorithms must be carefully happened in order to maintain aircraft level applications. In this thesis, the proposed architectures are modelled and defined as an assignment problem i.e. to assign the best avionics LRUs to the architecture and installation locations while satisfying some design constraints based on some cost functions required.

All the proposed optimisation algorithms are expressed as an integer and/or Binary programming. Integer and/or binary programming algorithms can be solved by COTS solvers as well as MATLAB functions. Moreover, contradicting objectives, for trade-offs studies, require multi-objective optimisation algorithms to calculate global optimal Pareto frontier. Here, Particle Swarm Optimisation (PSO) is proposed for multi-objective optimisation. The binary programming problems subject to single-objective optimisation are solved by “Intlinprog” function of the MATLAB which uses branch and bound algorithm to solve the problems.

4.2 IMA Architecture Layers for Allocation Modelling

As earlier in chapter 2 shown on figure (2-1), IMA architecture has three major layers for allocation problems including function mapping to software, software mapping to hardware, and hardware mapping to installation locations. In each layer, different various constraints need to be kept. These constraints can be classified as follows

Resource constraints: They are related to the computing resources in software mapping to hardware and/or devices like CPU, memory and power.

Physical Constraints: These are mainly related to installation locations when mapping hardware to installation locations including weight, volume, cooling capability and the number of slots.

Segregation/co-location constraints: these are required when two functions and/or hardware cannot be assigned to the same device and/or location due to safety or any other concerns.

Connectivity constraints: This shows the connections between IMA devices as well as virtual links.

Performance Constraints: they describe specifications that a design must satisfy including real-time scheduling and bandwidth.

The type of the allocation problem in this research has to do with hardware mapping to installation location while keeping the installation location constraints within acceptable limits. For each function and/or functions, at least three avionics LRUs are recorded and the optimisation problem is defined to allocate the best possible LRUs to avionics architecture to optimise as well as investigate various architectures from an LRU perspective. The problem is formulated by Integer Linear Programming (ILP) as described below.

4.3 Integer Programming

Integer programming (IP) is the discrete extension of linear programming (LP) where some or all of the variables are integer. It is also referred as Integer Linear Programming (ILP) in which the constraints and objective functions defined are linear. It was first appeared with the development of the cutting-planes algorithm by Ralph Gomory [127]. LP [128] is to minimize and/or maximize a linear cost function $f(x)$

$$f(x) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

Where it is subject to a number of equalities/inequalities constraints

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m$$

$x_1, x_2, \ldots, x_n$ are decision variables and non-negative.

The problem can also be expressed in matrix notation then it becomes

$$f^T\mathrm{x} \tag{4-1}$$

Subject to

$$Ax \leq b,$$

$$A_{eq}x = b_{eq} \tag{4-2}$$

$$x \in \mathbb{R}$$

Where f^T refers to vector transpose and x is the vector of decision variables. A,b,A_{eq}, and b_{eq} are matrices.

LP problems have a huge applications in operations research [129], [130] and control theory [131]. Since Danzig released the Simplex algorithm, many practical problems can be solved efficiently. Moreover, more algorithms like dual simplex algorithm and interior-point methods were developed to solve the problems even better and faster. An Integer Programming (IP) is derived by limiting variables x to be integer i.e. it to minimise

$$f^T\mathrm{x} \tag{4-3}$$

Subject to

$$Ax \leq b,$$

$$A_{eq}x = b_{eq} \tag{4-4}$$

$$x \in \mathbb{Z}$$

A further restriction on integer programming is when the decision variables defined as 0-1 and/or binary programming (BP). It is defined as minimising

$$f^T\mathrm{x} \tag{4-5}$$

Subject to

$$
\begin{aligned}
&Ax \leq b, \\
&A_{eq}x = b_{eq} \qquad \textbf{(4-6)} \\
&x \in \{0,1\}^n
\end{aligned}
$$

Integer and Binary programming have the same complexity and are solved with the same algorithms like branch-and-bound, LP-relaxation, cutting-planes, branch-and-cut, and branch-and-price [132].

4.4 The Formulation of the Optimisation Problem

In this section a Binary programming (BP) representation of each of the optimisation problems is given. In other words, it is clarified how the solution vectors are encoded and how the problems constraints and structures are defined in equalities and/or inequalities.

4.4.1 Avionics LRUs Assignment Problem

The proposed architectures in chapter 3 are modelled here. The AFCS architecture is defined as the small scale architecture and the avionics architecture is defined as the large scale architecture. In both architectures, due to the similarity of the avionics LRUs between Captain's side and the first officer's side, only half of the LRUs are considered. In other words, in the small architecture 11 avionics LRUs are considered including ADC, AHRS, GPS, VOR/ILS, RA, FMS, PFD, MFD, HUD, FCC, and IAC. In the large scale architecture 17 avionics LRUs are consider including 11 LRUs of the small architecture plus WXR, T3CAS, EFB, C/FDR, SATCOM and VHF.

Avionics LRUs which are networked within aircraft structure are defined as avionics nodes. For each node and/or function with the same name of their associated LRU, at least three LRUs with different technical specifications and operational capabilities are recorded in the database. The assignment problem is then defined as the allocation of avionics LRUs form database to each avionics node. Each node must only be assigned with one LRU from the database. The decision variables x_{ij} are defined as binary variables as below

$$x_{ij} = \begin{cases} 1 & if\ equipment\ j\ is\ assigned\ to\ node\ i \\ 0 & otherwise \end{cases} \quad \textbf{(4-7)}$$

Note: “otherwise” means that the equipment j is not assigned to node$i$. In other words, as for each node, there are at least three equipment candidates, only one needs to be assigned and it would be determined by 1, otherwise it would be 0.

Further for optimisation problem, the technical specifications and operational capabilities of each LRU are also defined as follows:

m_{ij}: The mass of each avionics LRU in node i with its associated equipment j

p_{ij}: The power consumption of each avionics LRU in node i with its associated equipment j

v_{ij}: The volume of each avionics LRU in node i with its associated equipment j

$mtbf_{ij}$: The reliability of each avionics LRU in node i with its associated equipment j in terms of MTBF

OC_{ij}: The operational capability of each avionics LRU in node i with its associated equipment j

4.4.2 Avionics LRUs Installation Locations Constraints

The limited mass and volume available in installation location is very critical as explained earlier. These two constraints have been considered in modelling. In other words, for allocation of avionics LRUs in to their installation locations these two limitations must be kept. For mass, in particular, there is also a defined target mass limitation imposed by designers in that the chosen architecture based on selected LRUs must not exist that value. In other words, based on the defined objective functions, the selected LRUs must satisfy these constraints as well.

Mass: the mass of each avionics LRU is recorded in capturing technical specifications. Each installation location has mass constraint which is implemented in architecture modelling for allocation of each LRU into installation location.

Volume: the actual box dimension of each LRU is recorded in capturing technical specifications. Each installation location has volume constraint which is implemented in architecture modelling for allocation of each LRU into installation location.

In this research, three installation locations are taken into account including cockpit, avionics bay and Centre fuselage shown in figure (4-1). Each location has mass and volume constraints and are defined as follows in avionics architecture modelling.

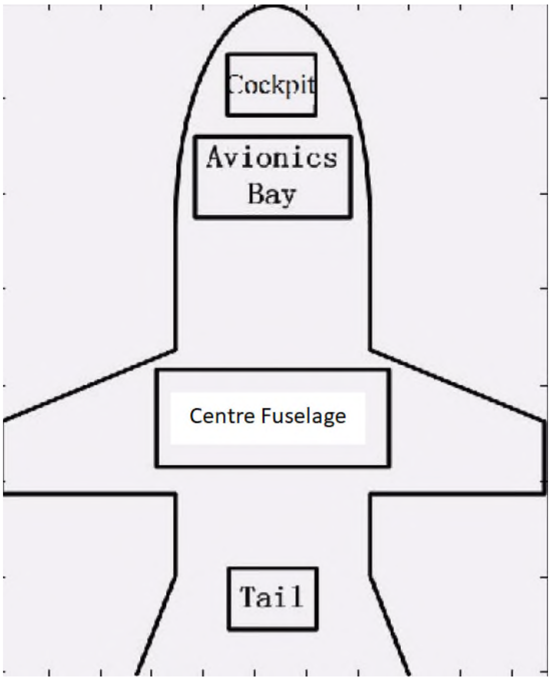


Figure 4-1 Avionics LRUs Installation Locations

M: The overall maximum allowable weight of the chosen architecture

$M_{cockpit}$: The maximum allowable weight of Avionics LRUs in cockpit installation location

M_{bay}: The maximum allowable weight of Avionics LRUs in avionics bay installation location

M_m: The maximum allowable weight of Avionics LRUs in middle installation location

$V_{cockpit}$: The maximum allowable volume of Avionics LRUs in cockpit installation location

V_{bay}: The maximum allowable volume of Avionics LRUs in avionics bay installation location

V_m: The maximum allowable volume of Avionics LRUs in middle installation location

4.4.3 Definition of the Objective Functions and Constraints

The avionics LRUs assignment problem is defined above is to allocate the best avionics LRUs from database to their associated avionics node while satisfying a number of design constraints. The single-objective cost functions defined as below:

1. The minimisation of the weight of the chosen avionics architecture i.e.

$$\min \qquad f_1 = \sum_i \sum_j m_{ij} x_{ij} \tag{4-8}$$

2. The minimisation of the power consumption of the chosen avionics architecture i.e.

$$\min \qquad f_2 = \sum_i \sum_j p_{ij} x_{ij} \tag{4-9}$$

3. The minimisation of the volume of the chosen avionics architecture i.e.

$$\min \qquad f_3 = \sum_i \sum_j v_{ij} x_{ij} \tag{4-10}$$

4. The maximisation of the reliability of the chosen avionics architecture i.e.

$$\max \qquad f_4 = \sum_i \sum_j mtbf_{ij} x_{ij} \tag{4-11}$$

5. The maximisation of the operational capability of the chosen avionics architecture i.e.

$$\max \qquad f_5 = \sum_i \sum_j OC_{ij} x_{ij} \tag{4-12}$$

For the above cost functions and avionics LRUs allocation to avionics node, the following constraints have to be satisfied:

a. Assignment constraints: for each avionics node, only one avionics associated LRU must be assigned. In other words, for each avionics node i, one of the equipment (LRU) from the database has to be assigned to each node i.e.

$$\sum_j x_{ij} = 1 \tag{4-13}$$

b. For a chosen architecture A, the maximum weight must not exceed the overall maximum allowable weight of the chosen architecture defined M i.e.

$$\sum_i \sum_j m_{ij} x_{ij} \leq \mathrm{M} \tag{4-14}$$

c. For a chosen architecture$A$, the maximum allowable weight of Avionics LRUs in cockpit installation location must not exceed the maximum weight limit in cockpit installation location $M_{cockpit}$ i.e.

$$\sum_i \sum_j m_{ij} x_{ij} \leq M_{cockpit} \tag{4-15}$$

d. For a chosen architecture $A,$ the maximum allowable weight of Avionics LRUs in avionics bay installation location must not exceed the maximum weight limit in avionics bay installation location M_{bay} i.e.

$$\sum_i \sum_j m_{ij} x_{ij} \leq M_{bay} \tag{4-16}$$

e. For a chosen architecture $A,$ the maximum allowable weight of Avionics LRUs in middle installation location must not exceed the maximum weight limit in middle installation location M_m i.e.

$$\sum_i \sum_j m_{ij} x_{ij} \leq M_m \tag{4-17}$$

f. For a chosen architecture $A,$ the maximum allowable volume of Avionics LRUs in cockpit installation location must not exceed the maximum volume limit in cockpit installation location $V_{cockpit}$ i.e.

$$\sum_i \sum_j v_{ij} x_{ij} \leq V_{cockpit} \tag{4-18}$$

g. For a chosen architecture $A,$ the maximum allowable volume of Avionics LRUs in avionics bay installation location must not exceed the maximum volume limit in avionics bay installation location V_{bay} i.e

$$\sum_i \sum_j v_{ij} x_{ij} \leq V_{bay} \tag{4-19}$$

h. For a chosen architecture $A,$ the maximum allowable volume of Avionics LRUs in middle installation location must not exceed the maximum volume limit in middle installation location M_m i.e.

$$\sum_i \sum_j v_{ij} x_{ij} \leq V_m \tag{4-20}$$

Note: When assigning each node i with equipment $j,$ their resources and installation locations are assumed to be fixed. Thus, the optimisation problem is trying to find the most suitable LRU/LRM to each avionics node within the proposed architectures.

In short, the assignment problem of avionics LRU to avionics node is to minimise or maximise the single objective cost functions defined (1-5) while satisfying the constraints considered in (a-h). The decision variables are defined according to (4-7).

4.5 The Assignment of Avionics Nodes to Installation Locations

The assignment and/or mapping problem of avionics nodes to installation locations is also defined as Binary Programming (BP). When mapping the avionics nodes to the installation locations, the LRUs' technical specifications like mass, volume and power consumption are already determined and fixed. In other words, in this stage, avionics LRUs for each node are selected and are to be assigned to installation location. Therefore, the binary programming variables is defined as follows

$$y_{il} = \begin{cases} 1 & if\ node\ i\ is\ assigned\ to\ installation\ location\ l \\ 0 & otherwise \end{cases} \tag{4-21}$$

The objective function of the mapping avionics nodes to installation location is to minimise the weight of the architecture i.e.

$$\min \quad f_6 = \sum_i \sum_j \sum_l m_{ij} x_{ijl} \tag{4-22}$$

Where x_{ijl} is defined as below

$$x_{ijl} = \begin{cases} 1 & if\ equipment\ j\ is\ assigned\ to\ node\ i\ to\ location\ l \\ 0 & otherwise \end{cases} \tag{4-23}$$

The assignment of avionics nodes to installation location problem has to satisfy the following constraints:

a. Assignment constraint: each avionics node can only be assigned to one installation location i.e.

$$\sum_l x_{ijl} = 1 \tag{4-24}$$

b. Segregation and/or co-location constraint: this constraint guarantees that two avionics LRU have to be installed separately, not in the same place. As the variables defined as binary the constraint is defined as below i.e.

$$\sum_{j}\sum_{i\in(PFD,MFD,HUD,FMS)} x_{ijl} = 0, \qquad \forall l \in L - \{Cockpit\} \tag{4-25}$$

$$\sum_{j}\sum_{i\in(ADC,AHRS,GPS,RA,VORILS,IAC)} x_{ijl} = 0, \qquad \forall l \in L - \{Avionics\ Bay\} \tag{4-26}$$

$$\sum_{j}\sum_{i\in(ADC,AHRS,GPS,RA,VORILS,)} y_{il} = 5, \qquad \forall l \in \{Avionics\ Bay\} \tag{4-27}$$

$$x_{FCC,j,l} = 0 \qquad l \in L - \{Middle\} \tag{4-28}$$

Equation (4-25) ensures that avionics LRUs like PFD, MFD, HUD, and FMS can only be installed in cockpit whereas equation (4-26) guarantees that avionics LRUs like ADC, AHRS, GPS, RA, VORILS, and IAC can only be installed in avionics bay. Equation (4-27) ensures that ADC, AHRS, GPS, RA, VORILS are installed next to each other and separate from IAC (Integrated Avionics Cabinet). Equation (4-28) states that FCC avionics LRU can only be installed in Middle installation location.

c. The mapping of avionics LRUs to installation locations has to satisfy the maximum allowable weight for each installation locations including cockpit, avionics bay and middle i.e.

$$\sum_{i}\sum_{j} m_{il}x_{ijl} \le M_{cockpit} \tag{4-29}$$

$$\sum_{i}\sum_{j} m_{il}x_{ijl} \le M_{bay} \tag{4-30}$$

$$\sum_{i}\sum_{j} m_{ij}x_{ijl} \le M_{m} \tag{4-31}$$

d. The mapping of avionics LRUs to installation locations has to satisfy the maximum allowable volume for each installation locations including cockpit, avionics bay and middle i.e.

$$\sum_{i}\sum_{j} v_{ij}x_{ijl} \leq V_{cockpit} \quad \textbf{(4-32)}$$

$$\sum_{i}\sum_{j} v_{ij}x_{ijl} \leq V_{bay} \quad \textbf{(4-33)}$$

$$\sum_{i}\sum_{j} v_{ij}x_{ijl} \leq V_{m} \quad \textbf{(4-34)}$$

In brief, the mapping of avionics LRUs to installation locations is to minimise the weight of the chosen architecture according to the cost function defined in (4-22), f_6, while satisfying constraints defined from (4-24 to 4-34).

4.6 Multi-objective Optimisation

What have been discussed so far were single-objective optimisation of avionics architecture. Nevertheless, avionics architecture optimisation objectives like mass and reliability and/or mass and operational capability shall both be optimal. It is obvious that an optimum in both cannot be archived since the objectives are conflicting. Multiple and conflicting objectives are solved by some techniques from multi-objective optimisation. The main focus is on the calculation of Pareto optimum for which a Multi-objective Particle Swarm Optimisation (MOPSO) is described.

Multi-objective optimisation aims to find a solution x such that x minimises a set of cost functions $f_1, \dots, f_k$

$$\min\{f(x)\} = min\{(f_1, \dots, f_k)\} \quad \textbf{(4-35)}$$

$f_1, \dots, f_k$ do not share the same minimum such that f spans non-empty regions in an m-dimensional space. The region of the space comprised of objective values of valid solutions which is called the objective space [133]. In combinatorial optimisation, the objective space is discrete. The task is then to identify one or more points from the objective space. In fact, the precise calculation of minimum is one of the major issues in multi-objective optimisation. The weight sum method, the lexicographical optimisation, the max-ordering, and Pareto optimisation are the most important strategies to solve these types of problems [134]. In this thesis, the weight sum method and Pareto optimisation are used.

4.6.1 Weighted Sum Method

To be able to solve multi-objective optimisation problems with a number of conflicting objectives, a reformulation is required. One method to reformulate the problem is to find a scalar-valued function. The weighted sum method combines all the multi-objective functions into one scalar which represents a weighted combination or preference order of all objectives i.e.

$$f(x) = w_1 f_1(\mathrm{x}) + w_2 f_2(\mathrm{x}) + \cdots + w_m f_m(\mathrm{x}) \tag{4-36}$$

The issue is that the solutions strongly depend on weighting coefficient $w = (w_1, w_2, \dots, w_m)$. These weights have to be positive and satisfy the following condition in case of normalisation of cost functions

$$\sum_{i=1}^{m} w_i = 1\,, w_i \in (0,1) \tag{4-37}$$

The weights have to be chosen manually and require a priori knowledge on the importance of the single cost functions. One of the common approach is to use ranking methods [135], however, many researchers developed a systematic approach to gain the weight. In ranking approach, the different objective functions are ordered by importance, and other objectives are assigned with integers by an increment increase. This approach is like the other method called categorisation methods where various objective functions are categorised from highly important moderately important. In both approaches, decision makers have to value the relative importance of the objective functions. Finally, the optimisation problem can then be carried out by any standard optimisation algorithms and results in one solution.

4.6.2 Multi-Objective Solving with Pareto Optimality

Pareto optimisation calculates a set of solution $x^{PF} = \{x_1^{PF}, \dots, x_n^{PF}\}$ called the Pareto Front. All n solutions x_i^{PF} are called efficient solutions. What they have in common is that they do not perform worse in all objectives than any other solutions from the solution space. The efficient solutions are said to dominate non-efficient solutions which is describe as below

$$X^{PF} = \left\{x_i^{PF} \parallel \nexists x \in X \forall f_j \in f: f(x) < f(x_i^{PF}),\ \ i = 1, \dots, n\right\} \qquad \textbf{(4-38)}$$

All efficient and/or non-dominated solutions $x_1^{PF}, \dots, x_n^{PF}$ from the Pareto optimum are always located at the border of the objective space. In other words, if a solution has no dominating solutions, it belongs to the Pareto optimum. In fact, the Pareto optimal solution is the best way to find trade-offs in many cases including avionics architecture optimisation.

4.7 Solving Algorithms

In this thesis, two algorithms are selected to solve the above optimisation problems. For single objective optimisation branch-and-bound algorithm, and Particle Swarm Optimisation (PSO) for multi-objective optimisation problems are used.

4.7.1 Branch-and-Bound Algorithm

The Branch-and-Bound algorithm uses a “divide and conquer” technique to explore feasible solutions for the combinatorial optimisation problems. It was first appeared in literature [136] by Land in 1960 and further adopted to solve Mixed Integer Linear Programming (MILP) problems [137]. The basic idea is to show the search space as a tree. The root of the tree is the complete search space. Each branch divides the search space in disjunctive subspaces, and each leave is one solution candidate. The Branch-and-Bound algorithm starts from the root node and continues down to the leaves until the global optimum is found. Figure 4-2 represents a search tree for a Binary Programming.

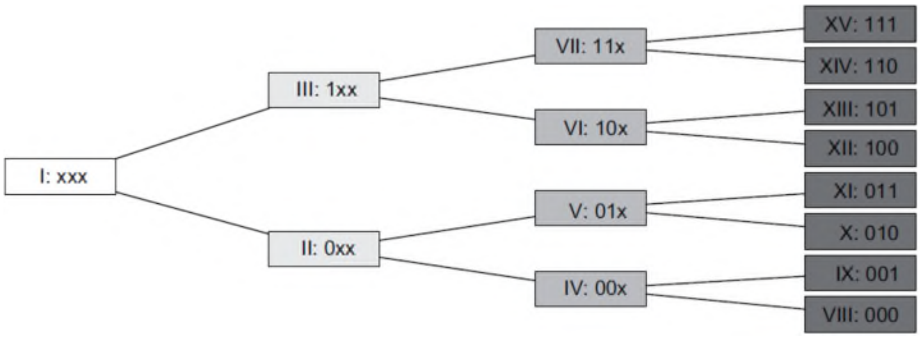


Figure 4-2 Branch-and-Bound Search Tree

In other words, the method create a tree which involves a subsets of the solutions and further identifies the branches of the tree. The branches are also called nodes. Consequently, the B&B algorithm checks each node using the upper and lower bound estimations on the variables. If the new subset solution is not better than the preceding solutions, the algorithm ignores that branch and explores the following branch to find the optimal solution. This iterative process continues until the global optimal solutions are found. A particular application of this algorithm whose concept is similar to this research and is used for investigating different flight control actuators technologies can be found in [48].

4.7.2 Multi-Objective Particle Swarm Optimisation (MOPSO)

Based on swarm intelligence, Particle Swarm Optimisation (PSO) is a stochastic and global searching optimisation developed by Kennedy and Eberhart in 1995 [138]. It is a population-based and evolutionary algorithm like the other evolutionary algorithms which is used for solving the complicated combinatorial optimisation problems [139]. The algorithm simulates the social behaviour of animals like birds, herds and fishes. It is basically suited for continuous variable problems, however, it can be adopted to discrete binary variables as well, and has successfully applied to many applications including neural networks, structural optimisation, shape topology optimisation and avionics architecture optimisation. Like the other evolutionary algorithms, its starts from population generation. The population is called swarm, and the individuals are called particles [140].

The position and velocity of the particles are defined as x_k^i and v_k^i respectively. The position of the particles updates as follows

$$x_{k+1}^i = x_k^i + v_{k+1}^i \tag{4-39}$$

The velocity is calculated as below

$$v_{k+1}^i = \omega v_k^i + c_1 r_1 \left(p_k^i - x_k^i\right) + c_2 r_2 (p_k^g - x_k^i) \tag{4-40}$$

Where parameters are defined as follows:

p_k^i: Best “remembered” individual particle position

p_k^g: Best “remembered” swarm position

ω: Inertia constant

c_1, c_2: Cognitive and social parameters

r_1, r_2: Random numbers between 0 and 1

Figure 4-3 represents the general algorithm for PSO.

1. For $i = 1$ to n_pop (n population)
   Initialize $p[i]$ randomly (p is the population of particle)
   Initialize $v[i] = 0$
   Evaluate $p[i]$
   GBEST=Best particle found in $p[i]$
2. End for
3. For $i = 1\ to\ n\_pop$
   PBEST $s[i] = p[i]$
   Initialize the "memory" of each particle
4. End for
5. Repeat
   For $i = 1\ to\ n\_pop$
   $v[i] = \omega \times v[i] + c_1 \times r_1 \times (PBEST\ s[i] - p[i]) + c_2 \times r_2 \times$
   $(PBEST\ s[GBEST] - p[i])$
   Calculate speed of each particle
   $pop[i] = p[i] + v[i]$
   Evaluate $p[i]$
   If new solution is better than PBEST $s[i] = p[i]$
   GBEST=Best particle found in $p[i]$
   End for
6. Until stopping condition is reached

Figure 4-3 Particle Swarm Optimisation Pseudo Code

Since the development of the PSO algorithm, it has been improved and new versions of the algorithm have been introduced [141]. In this thesis, PSO has been used for solving both single-objective and multi-objective cost functions. The single objective solving is compared by "Intlinprog" MATLAB function. It is further shown in next chapter that PSO successfully obtain the same cost value gained by exact algorithm i.e. branch-and-bound.

4.8 Encoding of Avionics System Architecture

In what follows, the encoding of the avionics assignment problem for Automatic Flight Control System (AFCS) is explained. Avionics LRUs and/or nodes are defined as an 11 arrays like Figure 4-4.

A	B	C	D	E	F	G	H	I	J	K

Figure 4-4 Avionics LRU Assignment Problem Encoding

Each letter is an avionics node. For instance, A is ADC which is at least 1 and maximum 5 from avionics database and its relevant ID. In other words, each integer means a particular type of ADC investigated that is ESCADU, ADUM, AADC, DADC, and AMADC which are the ID of each ADC in the database. Letter B represents the AHRS LRUs and integer 6 to 9 are AH1000, LCR100, LCR100N, and AHS3000. This continues for all the other LRUs, GPS receiver, VORILS, RA, FMS, PFD, MFD, HUD, FCC, and IAC respectively. Thus, according to the database for the AFCS architecture each letter is defined as follows:

$1 \leq A \leq 5$, $6 \leq B \leq 9$, $10 \leq C \leq 14$,

$15 \leq D \leq 17$, $18 \leq E \leq 20$, $21 \leq F \leq 25$,

$26 \leq G \leq 29$, $30 \leq H \leq 33$, $34 \leq I \leq 37$

$38 \leq J \leq 41$, $42 \leq K \leq 44$

This way of encoding also ensures that each avionics node will be allocated by one relevant avionics LRU. The same way is extended for 17 avionics node for the large scale optimisation problem. This has been implemented as a function in MATLAB, **CreateModle**, which further be called in main PSO algorithm.

5 Simulations and Results

For demonstration and evaluation of the proposed method, an avionics architecture was developed. The proposed architecture is divided into two parts including AFCS (The small scale) and avionics system architecture (the large scale). The architectures are artificial since there was no real aircraft as a reference for this study. However, the system functions, peripherals and the aircraft anatomy considered is like a single-aisle aircraft with two engines. For the installation of avionics LRUs, three installation locations are considered including cockpit, avionics bay and middle. Table 5 shows the installation locations and their related physical constraints.

Table 5 The Design Constraints of the Installation Locations

Installation Locations	Mass (Kg)	Volume (Inch m3)
Cockpit	50	3100
Avionics Bay	40	2500
Middle	30	1000

Each installation location has mass and volume limits e.g. each location has a maximum allowable mass and volume to install avionics LRUs.

In what follows, avionics architecture optimisation problems in different scenarios are simulated. The avionics assignment problem for the AFCS (the small scale) and the avionics architecture (the large scale) are simulated for single-objective and multi-objective optimisation problems. Various cost functions defined in chapter 4 are being used here in this section to investigate different avionics architectures as well as avionic architecture trade-offs.

5.1 Single-objective Optimisation of the SSA

The small scale architecture is related to Automatic Flight Control System (AFCS) architecture. The number of avionics nodes considered are 11 avionics LRUs including ADC, AHRS, GPS receiver, VOR/ILS, RA, FMS, PFD, MFD, HUD, FCC, and IAC. The following scenarios based on cost functions defined in chapter 4 are to investigate various avionics architectures. In short, in each scenario, the objective is to allocate the best possible avionics LRUs to the avionics architecture to achieve each goal defined like minimum weight while satisfying a number of design constraints imposed.

5.1.1 The Architecture with Minimum Weight for SSA

The minimum weight architecture for AFCS architecture is simulated based on the cost function defined in equation (4-8) while satisfying constraints from (a-h). The results of the optimisation problem for this scenario based on MATLAB "INTLINPROG" is as follows:

The weight of the architecture and/or the amount of cost function for eleven avionics nodes and/or LRUs is $z = 36.9$ Kg. In other words, the selected LRUs lead to an architecture with minimum weight. The allocated LRUs from the database for minimum weight are shown in table 6.

Table 6 Allocated LRUs for Minimum Weight for SSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC	IAC
Selected LRUs ID	ADUM	AH1000	GPSWAAS	ANS	KRA405	UNS1FW	EFD750	CMA6800	GHD	UKB501	PU3000

The above problem has also been solved by PSO algorithm as a single-objective optimisation. Figure (5-1) shows the value of weight cost function using PSO algorithm which is calculated $z = 36.99$ Kg. The PSO parameters used in this simulation are shown in table 7. As the problem has been solved before by exact method using MATLAB Intlinprog. The parameters, particularly,c_1, c_2 are chosen in way that the algorithm converge to $z = 36.99$ Kg. This happened by three different runs to reach to the desired value. In other words, they are chosen to result in the optimal solutions already calculated by exact method. There is a general rule for setting these two learning factors that is $c_1 + c_2 \leq 4$. In general, having a higher value of ω, $c_1, and\, c_2$ causes to explore and create newer solutions, however, the lower value of which exploit the current solutions and make them better.

Table 7 Single-objective PSO Optimisation Parameters for Min Weight in SSA

Iteration	Population	ω	c_1	c_2
300	50	1	0.3	0.3

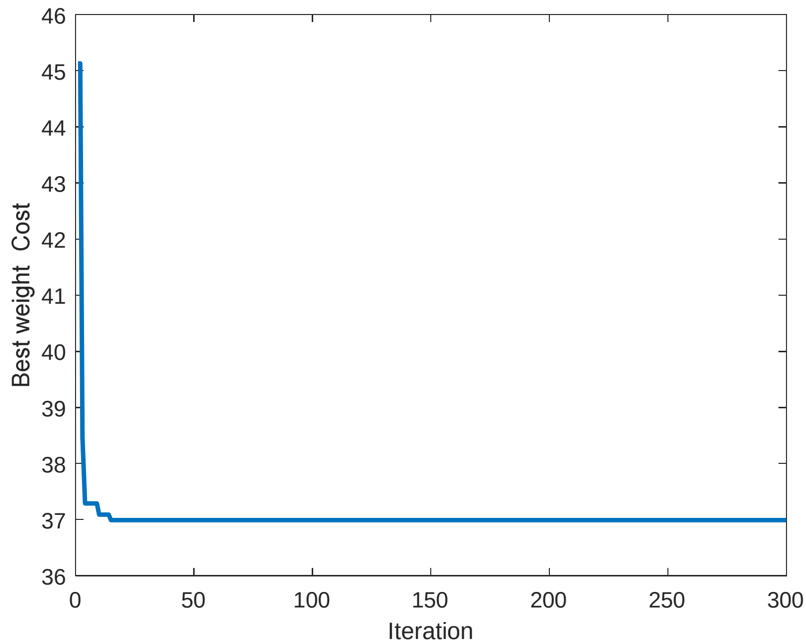


Figure 5-1 Best Weight Cost for AFCS Architecture Using PSO

As it is observed, the allocated LRUs and the value of cost function are the same in both methods. In other words, figure 5-1 represents that the PSO algorithm successfully converged to the optimal value $z = 36.99$ Kg which is the minimum weight architecture for the small scale architecture i.e. the selected LRUs lead to a lowest possible weight architecture while keeping the constraints within limits.

5.1.2 The Architecture with Minimum Power Consumption for SSA

The minimum power consumption architecture for AFCS architecture is simulated based on the cost function defined in equation (4-9) while satisfying constraints from (a-h). The results of the optimisation problem for this scenario based on MATLAB “INTLINPROG” is as follows:

The power consumption of the architecture and/or the amount of cost function for eleven avionics nodes and/or LRUs is $z = 388$ W. In other words, the LRUs selected lead to an architecture which needs the lowest possible power consumption. The allocated LRUs from the database for minimum power consumption are shown in table 8.

Table 8 Allocated Avionics LRUs for Minimum Power Consumption (AFCS)

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC	IAC
Selected LRUs ID	ADUM	AH1000	GPSWAAS	ANS	KRA405	GE6883	ISFD	CMA6800	GHD	UKB501	PU3000

The minimum power consumption architecture for AFCS is different from the minimum weight architecture in terms of the selected LRUs for FMS and PFD.

5.1.3 The Architecture with Minimum Volume for SSA

The minimum volume architecture for AFCS architecture is simulated based on the cost function defined in equation (4-10) while satisfying constraints from (a-h). The results of the optimisation problem for this scenario based on MATLAB “INTLINPROG” is as follows:

The volume of the architecture and/or the amount of cost function for eleven avionics nodes and/or LRUs is$z = 2448.5\ Inch\ m^3$. In other words, based on the selected LRUs the architecture is required the lowest possible volume which very important for small high performance military jets. The allocated LRUs from the database for minimum volume architecture are shown in table 9.

Table 9 Allocated LRUs for Minimum Volume for SSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC	IAC
Selected LRUs ID	AMADC	AH1000	GPSWAAS	ANS	KRA405	CMA9000	ProPFD	CMA6800	GHD	UKB501	PU3000

The minimum volume for AFCS architecture is different from the minimum weight architecture in terms of the selected LRUs for ADC, FMS and PFD. It is also different from minimum power consumption architecture in terms of the selected LRUs for ADC, FMS, and PFD.

5.1.4 The architecture with Maximum MTBF for SSA

The maximum reliability architecture for AFCS architecture is simulated based on the cost function defined in equation (4-11) while satisfying constraints from (a-h). The results of the optimisation problem for this scenario based on MATLAB "INTLINPROG" is as follows. It should be noted that the "INTLINPROG" is basically designed for minimisation problem, therefore, for maximisation problem the objective function must be multiplied by "-1".

The maximum reliability (MTBF) of the architecture and/or the amount of optimal cost function for eleven avionics nodes and/or LRUs is $z = 189700$ hours. In other words, the selected LRUs results in an architecture with maximum MTBF in terms of hour. The allocated LRUs from the database for maximum reliability are shown in table 10.

Table 10 Allocated LRUs for Maximum MTBF for SSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC	IAC
Selected LRUs ID	DADC	AH1000	CMA3024	ARN147	KRA405	UNS1ESPW	G3X	CMA6800	HGS	EFA	CMA5000

The maximum reliability architecture is different from minimum weight, minimum power consumption, and minimum volume architecture in terms of the selected LRUs for ADC, GPS, VOR/ILS, FMS, PFD, HUD, FCC and IAC. This confirms that minimum weight and maximum reliability architectures are different in many cases, therefore, weight and reliability are conflicting criteria which need multi-objective optimisation approaches for trade-offs studies. In other words, both weight and reliability need to be optimal that cannot be achieved by single objective solving.

The above problem has also been solved by PSO algorithm as a single-objective optimisation. Figure (5-2) shows the best MTBF cost value using PSO which is calculated $z = 189700$ hours. The PSO parameters used in this simulation are shown in table 11.

Table 11 Single-objective PSO Optimisation Parameters for Max MTBF

Iteration	Population	ω	c_1	c_2
300	50	1	0.1	0.3

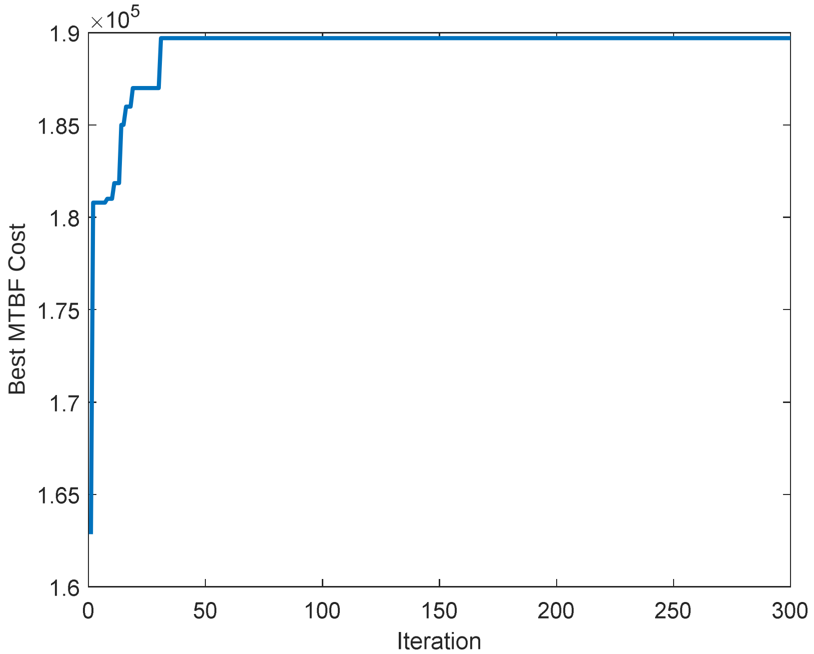


Figure 5-2 Best MTBF Cost for SSA Using PSO

In both methods, the value of the cost function and the allocated LRUs are the same. In other words, figure 5-2 shows that the PSO successfully converged to the optimal value $= 189700$ hours which is the maximum MTBF architecture for the small scale architecture i.e. the selected LRUs lead to an architecture whose LRUs have the highest MTBF.

5.1.5 The Architecture with Maximum Operational Capability for SSA

The maximum operational capability architecture for AFCS architecture is simulated based on the cost function defined in equation (4-12) while satisfying constraints from (a-h). The results of the optimisation problem for this scenario based on MATLAB "INTLINPROG" is as follows. It should be noted that the

"INTLINPROG" is basically designed for minimisation problem, therefore, for maximisation problem the objective function must be multiplied by "-1".

The operational capability of the architecture and/or the amount of cost function for eleven avionics nodes and/or LRUs is$z = 10.7$. The allocated LRUs from the database for maximum operational capability are shown in table 12.

Table 12 Allocated LRUs for Maximum Operational Capability for SSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC	IAC
Selected LRUs ID	AADC	LCR100N	CMA6024	ANV241A	DRA	UNS1ESPW	G3X	EFI890	Aviguide	UKB501	FV4000

The selected avionics LRUs for maximum operational capability architecture are totally different from the selected avionics LRUs for minimum weight architecture. This confirms that these two criteria are conflicting and will further be studied for trade-offs architecture. Regarding the selected avionics LRUs for maximum operational capability, the operational capability of the AFCS is described as follows:

The AFCS architecture is an open system architecture that meets the essential CNS/ATM requirements and its navigational performance provides crew GBAS CAT I, SBAS LPV, ILS CAT II approach capability, vertical navigation, B-RNAV, RNP down to 0.3, and is RVSM compliant.

The maximum operational capability architecture problem has also been solved by PSO as a single-objective optimisation problem as below. The PSO parameters used in this simulation are shown in table 13.

Table 13 Single-objective PSO Optimisation Parameters for Max OC in SSA

Iteration	Population	ω	c_1	c_2
300	50	1	0.1	0.3

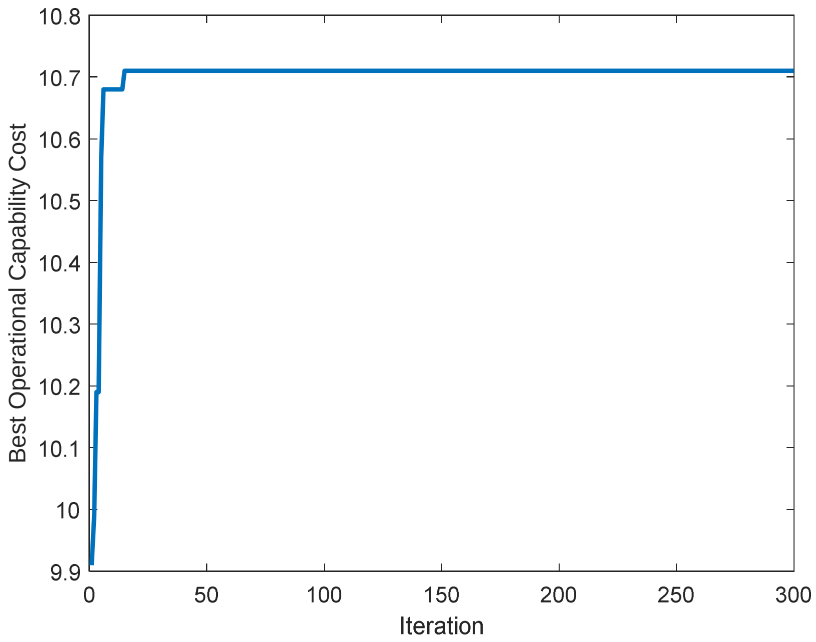


Figure 5-3 Best Operational Capability Cost for SSA Using PSO

In both methods, the value of the cost function and the allocated LRUs are the same. In other words, figure 5-3 illustrates that the PSO algorithm successfully converged to the optimal value which is $z = 10.7$ i.e. the maximum operational capability architecture for the small scale architecture that is the selected LRUs lead to an architecture whose LRUs have the highest operational capabilities.

5.2 Multi-objective Optimisation of the Small Architecture

In previous sections, it has been clear that weight and reliability as well as weight and operational capability are conflicting criteria as the selected LRUs are almost different in each scenario. Thus, to achieve an optimised avionics architecture with minimum weight and maximum reliability or minimum weight and maximum operational capability needs a multi-objective optimisation approach. Here, Particle Swarm Optimisation (PSO) algorithm is used for avionics architecture trade-offs. In what follows two scenarios are run.

5.2.1 MOPSO for Minimum Weight and Maximum MTBF

The avionics LRU assignment problem for optimisation of avionics architecture in terms of weight and reliability is defined as a multi-objective optimisation problem since weight and reliability are two conflicting criteria. Here, PSO algorithm is used to solve the problem. The cost functions are defined using the weighted sum approach by combination of equation (4-8) and (4-11) for weight minimisation and MTBF maximisation respectively. The results of this optimisation problem based on PSO algorithm is as follows. The PSO parameters used in this run is shown in table 14.

Table 14 MOPSO Optimisation Parameters for Min Weight and Max MTBF

Iteration	Population	ω	c_1	c_2
300	50	1	0.2	0.4

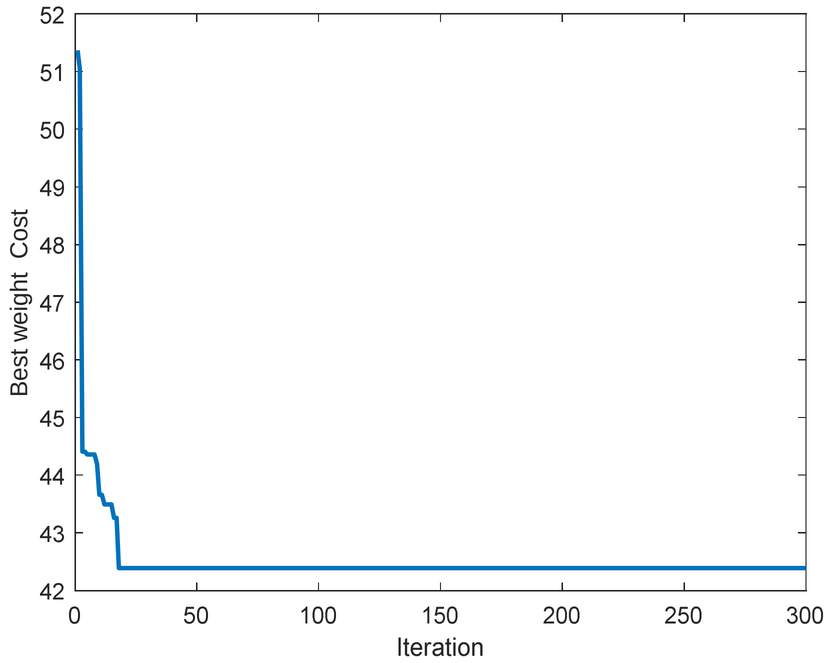


Figure 5-4 Best Weight Cost Based on Pareto Judgement for SSA

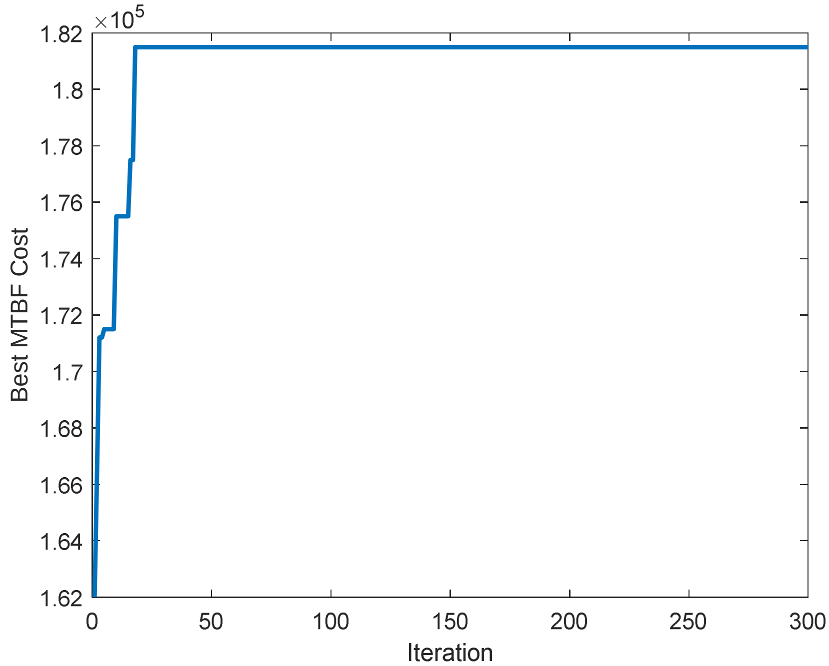


Figure 5-5 Best MTBF Cost Based on Pareto Judgement for SSA

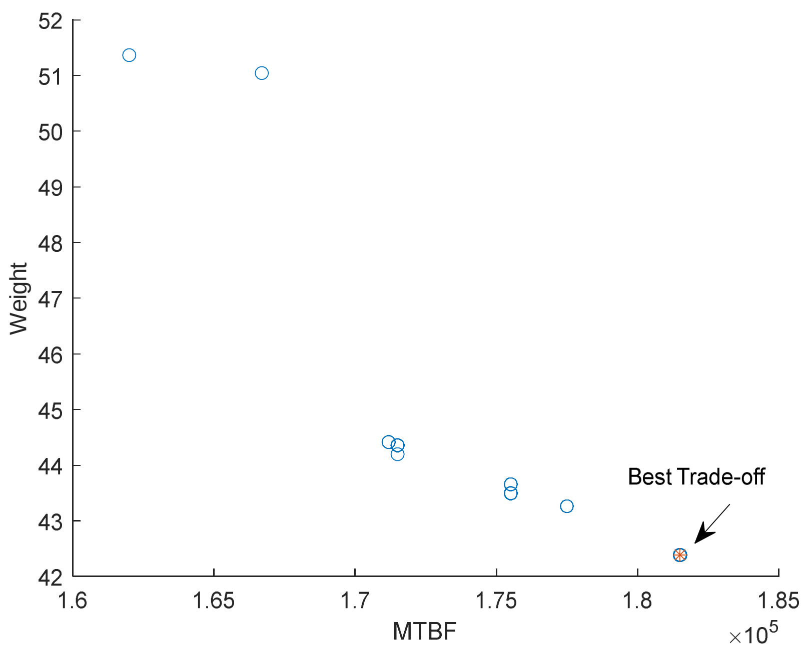


Figure 5-6 Pareto Optima for Min Weight and Max MTBF for SSA

Figure 5-4 and 5-5 represent the optimal value of minimum weight and maximum MTBF in Pareto-optimality respectively using PSO. In other words, the best trade-off architecture happens at $w = 42.39$ Kg and $MTBF = 181500$ Hours which is shown in figure 5-6. The allocated avionics LRUs for minimum weight and maximum MTBF in SSA is shown in table 15.

Table 15 Allocated LRUs for Min Weight and Max MTBF for SSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC	IAC
Selected LRUs ID	DADC	AH1000	CMA3024	ARN147	KRA405	UNS1FW	EFD750	CMA6800	HGS	UKB501	PU3000

The above problem has also been solved by weighted sum method i.e. the two cost functions f_4 for maximum MTBF and f_1 for minimum weight defined in chapter 4 are weighted using the mean of MTBF over weight $w_2 = \frac{MTBF}{Weight} = 3630,$ for the weight cost function, to obtain trade-off architecture. The weighting factor for MTBF is defined as $w_1 = 1$. This way of weighting is related to the decision-maker's preference function. In other words, the least important objective receives a weight of one, and integer weights with consistent increments are assigned to objectives that are more important [142]. The following figures represent the outcomes.

$$F = w_1 f_4 - w_2 f_2$$

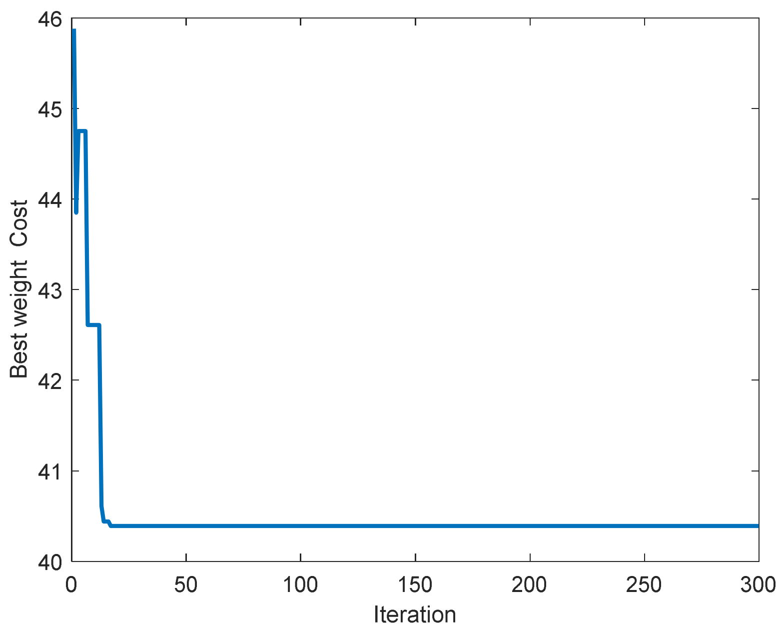


Figure 5-7 Best Weight Cost Based on WSM for SSA

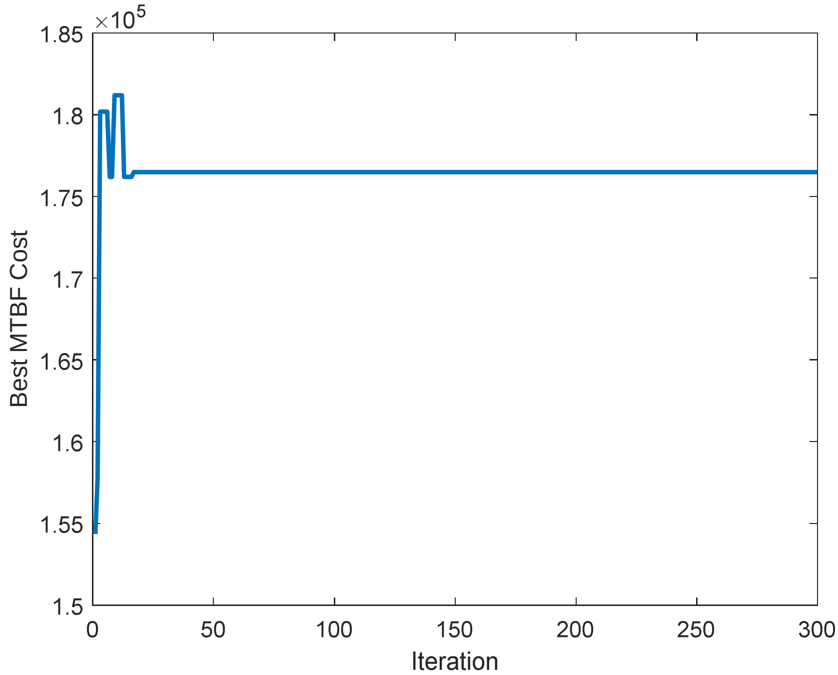


Figure 5-8 Best MTBF Cost Based on WSM for SSA

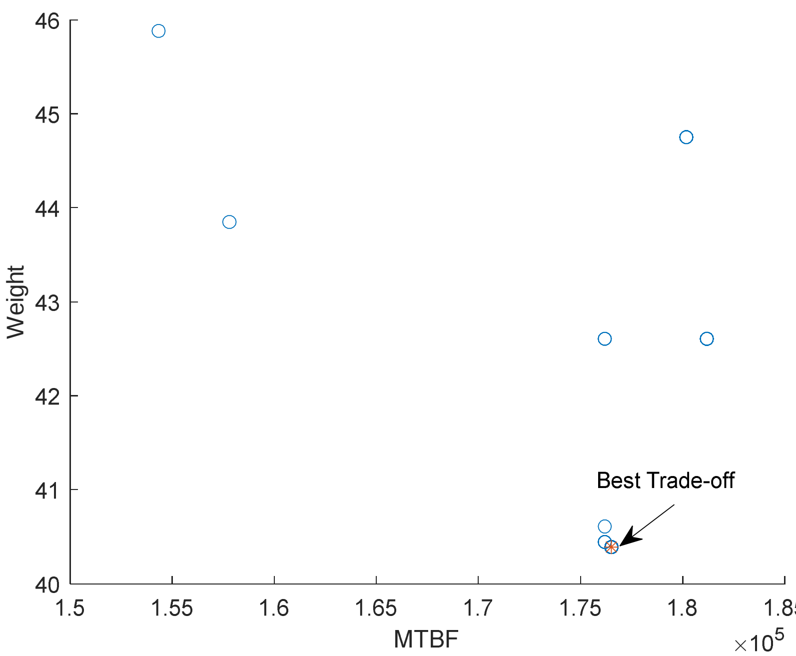


Figure 5-9 Pareto Optima Based on WSM for SSA

Figure 5-7 and 5-8 illustrate the optimal value of minimum weight and maximum MTBF respectively using weighted sum method by use of PSO algorithm. In other words, the best trade-off architecture happens at $w = 40.39$ Kg and $MTBF =$

176500 hours which is shown in figure 5-9. The allocated avionics LRUs for minimum weight and maximum MTBF in weighted sum method for SSA are shown in table 16.

Table 16 Allocated LRUs in Weighted Sum Method for SSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC	IAC
Selected LRUs ID	DADC	AH1000	GPSWAAS	ARN147	KRA405	UNS1FW	EFD750	CMA6800	HGS	UKB501	PU3000

The selected avionics LRUs in Pareto approach and weighted sum method are only different in GPS receiver, and the weight difference is 2Kg less compared to Pareto optima architecture in that the relative importance dedicated for the weight cost function assigned more than MTBF cost function. It consequently led to a lower MTBF from 181500 hour to 176500 (5000 hour less).

5.2.2 MOPSO for Min Weight and Max Operational Capability

The avionics LRU assignment problem for optimisation of avionics architecture in terms of weight and operational capability is defined as a multi-objective optimisation problem since weight and operational capability are two conflicting criteria. Here, PSO algorithm for Pareto optima and WSM are used to solve the problem. The PSO parameters used in this run is shown in table 17.

Table 17 MOPSO Optimisation Parameters for Min Weight and Max OC

Iteration	Population	ω	c_1	c_2
300	50	1	0.2	0.4

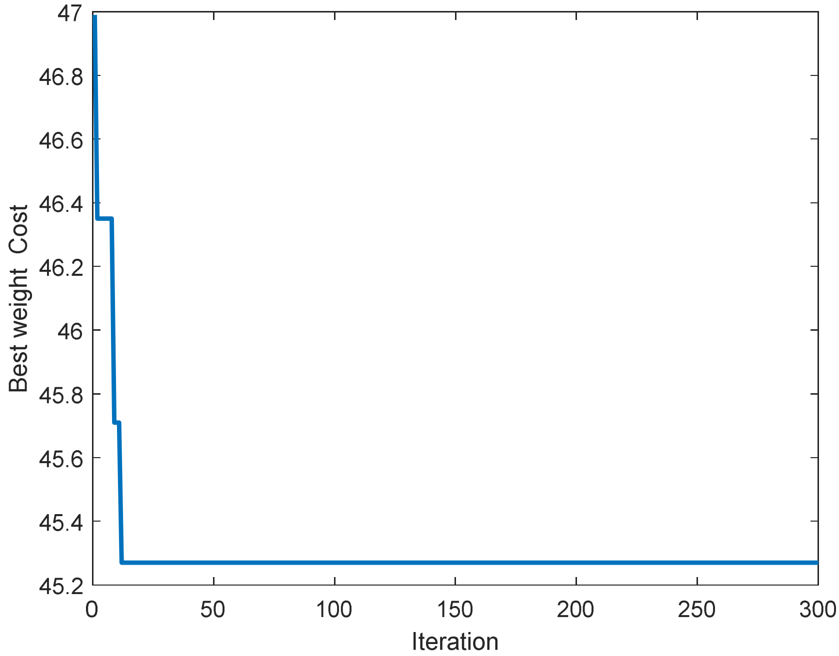


Figure 5-10 Best Weight Cost Based on Pareto Judgement for SSA

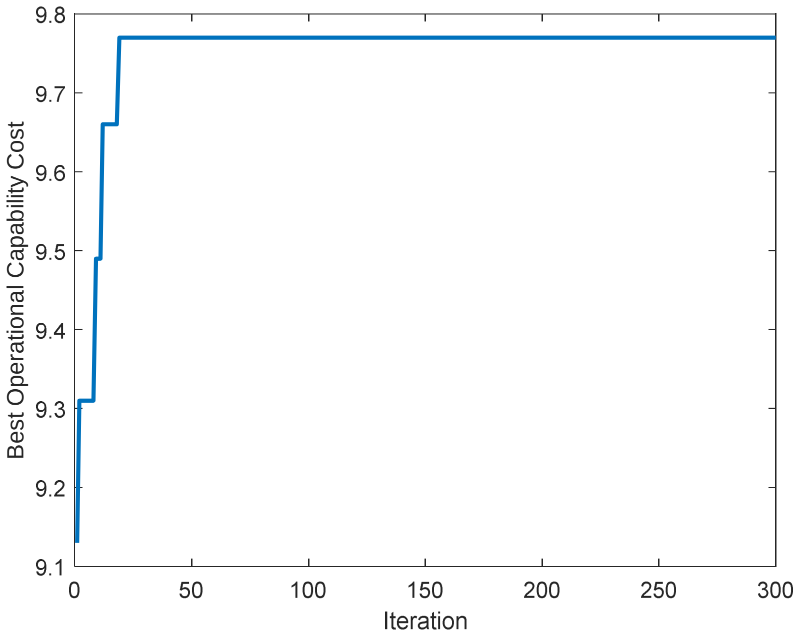


Figure 5-11 Best OC Cost Based on Pareto Judgement for SSA

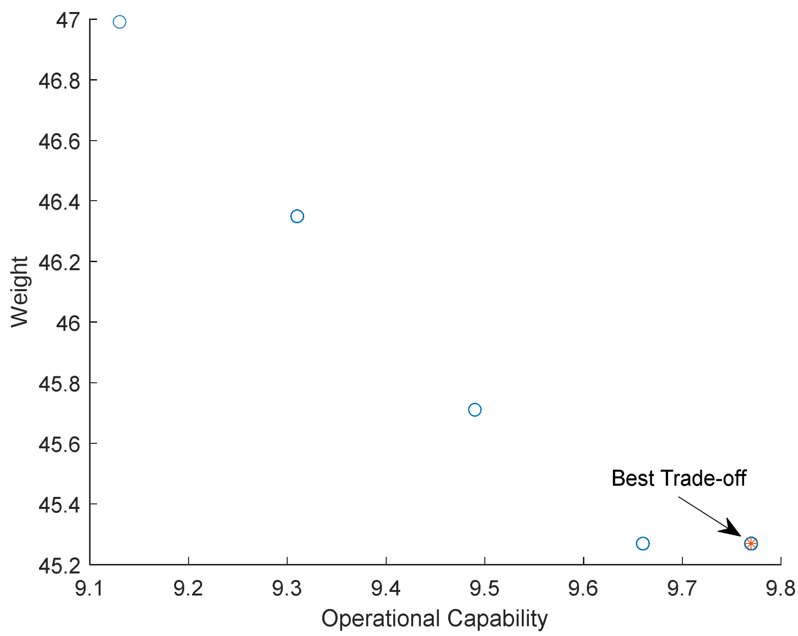


Figure 5-12 Pareto Optima for Min Weight and Max OC for SSA

Figure 5-10 and figure 5-11 represent the optimal value of minimum weight and maximum operational capability in Pareto-optimality respectively using PSO. In other words, the best trade-off architecture happens at $w = 45.39$ Kg and $OC = 9.79$. The allocated avionics LRUs for minimum weight and maximum operational capability in SSA is shown in table 18.

Table 18 Allocated LRUs in Pareto Judgement for Min Weight and Max OC

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC	IAC
Selected LRUs ID	DADC	LCR100N	CMA6024	ANS	LRA2100	UNS1FW	G3X	CMA6800	HUD2020	UKB501	PU3000

Regarding the selected avionics LRUs in this scenario, the operational capability of the AFCS architecture in minimum weight and maximum operational capability compared to the single-objective maximum operational capability is almost the same, however, the weight of the architecture is 5 Kg higher than minimum weight architecture. Therefore, the operational capability of the architecture based on the selected LRUs is GBAS CAT I, SBAS LPV, ILS CAT III approach capable by HUD2020, vertical navigation, B-RNAV, RNP down to 0.3, and is RVSM compliant.

The above problem has also been solved by weighted sum methods i.e. the two cost functions f_5 for maximum operational capability and f_1 for minimum weight defined in chapter 4 are weighted to obtain trade-off architecture. The weighting factor for operational capability is defined as $w_1 = 1$ and $w_2 = 5$ for the weight cost function. This is chosen based on decision-maker's preference which shows that the operational capability has the lowest important. In other words, the least important objective receives a weight of one, and integer weights with consistent increments are assigned to objectives that are more important. The relative importance of weight cost function is dedicated higher than operational capability. The following figures represents the outcomes.

$$F = w_1 f_5 - w_2 f_1$$

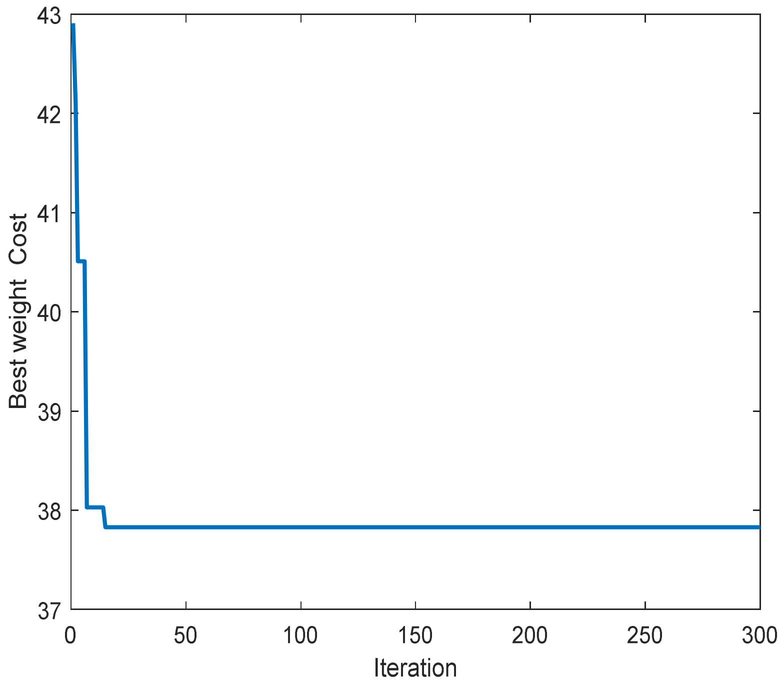


Figure 5-13 Best Weight Cost Based on WSM for SSA

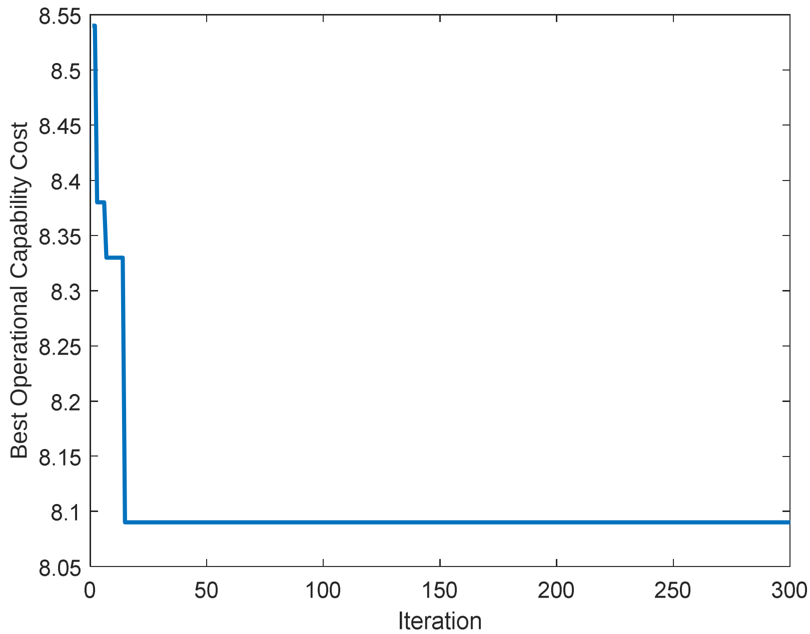


Figure 5-14 Best Operational Capability Cost Based on WSM

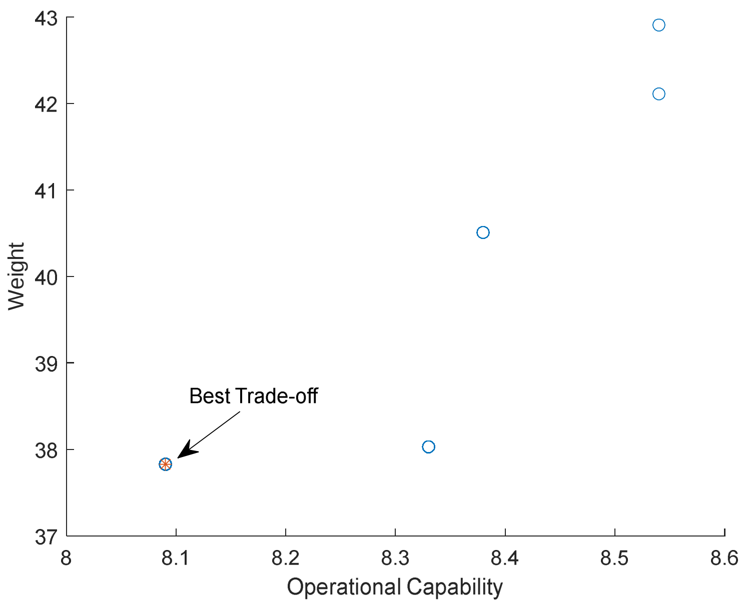


Figure 5-15 Pareto Optima Based on Weighted Sum Method

Figures 5-13 and 5-14 illustrate that the optimal value of minimum weight and maximum operational capability respectively using weighted sum method by using of PSO algorithm. In other words, the best trade-off architecture happens at $w = 37.9$ Kg and $OC = 8.12$. Compared to the Pareto optima, the weight of architecture is 7.49 Kg lighter, however, the operational capability is 1.67 less than the Pareto optima. This means a lower approach capability from ILS CATIII to ILS CAT I landing capability. The allocated avionics LRUs for minimum weight and maximum Operational Capability in weighted sum approach for SSA is shown in table 19.

Table 19 Allocated LRUs for Min Weight & Max OC in WSM for SSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC	IAC
Selected LRUs ID	ADUM	AH1000	GPSWAAS	ANS	KRA405	UNS1FW	G3X	CMA6800	GHD	UKB501	PU3000

The selected avionics LRUs in Pareto approach and weighted sum method are only different in ADC, AHRS, GPS receiver and HUD.

5.3 Single-Objective Optimisation of the Large Scale Architecture

The AFCS architecture has been extended to all avionics systems architecture shown in figure (3-5). Therefore, the avionics nodes considered are the previous eleven nodes plus WXR, TCAS, EFB, C/FDR, SATCOM, and VHF. The following scenarios based on cost functions defined in chapter 4 are to investigate various avionics architectures. In short, in each scenario, the objective is to allocate the best possible avionics LRUs to the avionics architecture to achieve each gaol defined like minimum weight while satisfying the number of design constraints imposed. The total number of avionics nodes in this case are 17.

5.3.1 The Architecture with Minimum Weight for LSA

The minimum weight architecture for avionics architecture (LSA) is simulated based on the cost function defined in equation (4-8) while satisfying constraints from (a-h). The results of the optimisation problem for this scenario based on MATLAB “INTLINPROG” is as follows:

The weight of the architecture and/or the amount of cost function for seventeen avionics nodes and/or LRUs is $z = 57.26$ Kg. In other words, the selected LRUs lead to an architecture with minimum weight.

The allocated avionics LRUs in minimum weight architecture for the large scale architecture are shown in table 20.

Table 20 Allocated LRUs for Min weight in LSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC
Selected LRUs ID	ADUM	AH1000	GPSWAAS	ANS	KRA405	UNS1FW	EFD750	CMA6800	GHD	UKB501
Avionics LRUs	IAC	WRX	TCAS	EFB	FDR	SATCOM	VHF			
Selected LRUs ID	PU3000	RDR7000	T2CAS	CMA1310	LDR	CMA2200SB	SRT700			

The above problem has also been solved as a single-objective optimisation problem using PSO algorithm. Figure (5-16) represents the optimal value for minimum weight cost function which is the same as the branch-and-bound algorithm solution ($z = 57.26$) Kg by INTLINPROG. Also, the PSO parameters chosen are similar to the small scale architecture, and are adjusted to result in and optimal value gained which was already solved by exact method.

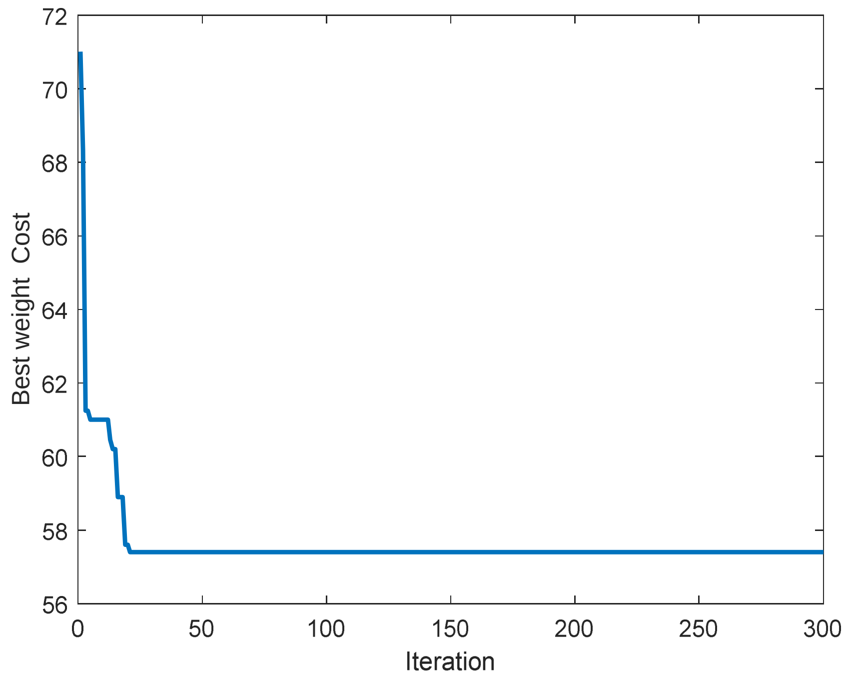


Figure 5-16 Best Weight Cost for the LSA

The value of the cost function and the allocated avionics LRUs are the same in both branch-and-bound and PSO algorithms.

5.3.2 The Architecture with Minimum Power Consumption for LSA

The minimum power consumption architecture for avionics architecture (LSA) is simulated based on the cost function defined in equation (4-9) while satisfying constraints from (a-h). The results of the optimisation problem for this scenario based on MATLAB "INTLINPROG" is as follows:

The power consumption of the architecture and/or the amount of cost function for seventeen avionics nodes and/or LRUs is $z = 593$ W. In other words, the selected LRUs lead to an architecture that need the lowest possible power consumption. The allocated LRUs from the database for minimum power consumption are shown in table 21.

Table 21 Allocated LRUs in Min Power for LSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC
Selected LRUs ID	ADUM	AH1000	GPSWAAS	ANS	KRA405	GE6883	ISFD	CMA6800	GHD	UKB501
Avionics LRUs	IAC	WRX	TCAS	EFB	FDR	SATCOM	VHF			
Selected LRUs ID	PU3000	RDR7000	T2CAS	CMA1310	LDR	CMA2200SB	SRT700			

The minimum power architecture and minimum weight architecture for the large scale are different in terms of selected LRUs including FMS and PFD.

5.3.3 The Architecture with Minimum Volume for LSA

The architecture for minimum volume the avionics architecture (LSA) is simulated based on the cost function defined in equation (4-10) while satisfying constraints from (a-h). The results of the optimisation problem for this scenario based on MATLAB "INTLINPROG" is as follows:

The volume of the architecture and/or the amount of cost function for seventeen avionics nodes and/or LRUs is $z = 3570.5\ Inch\ m^3$. In other words, the selected LRUs lead to an architecture with minimum volume. The allocated LRUs from the database for minimum power consumption are shown in table 22.

Table 22 Allocated LRUs in Min Volume for LSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC
Selected LRUs ID	AMADC	AH1000	GPSWAAS	ANS	KRA405	CMA9000	ProPFD	CMA6800	GHD	UKB501
Avionics LRUs	IAC	WRX	TCAS	EFB	FDR	SATCOM	VHF			
Selected LRUs ID	PU3000	RDR7000	T2CAS	CMA1310	LDR	IRT2110	SRT700			

The minimum volume architecture and minimum weight architecture for the large scale are different in terms of selected LRUs including ADC, FMS, PFD, and SATCOM.

5.3.4 The Architecture with Maximum Reliability for LAS

The maximum reliability architecture for the avionics architecture is simulated based on the cost function defined in equation (4-11) while satisfying constraints from (a-h). The results of the optimisation problem for this scenario based on MATLAB “INTLINPROG” is as follows. It should be noted that the “INTLINPROG” is basically designed for minimisation problem, therefore, for maximisation problem the objective function must be multiplied by “-1”.

The reliability (MTBF) of the architecture and/or the amount of cost function for seventeen avionics nodes and/or LRUs is $z = 257700$ hours. In other word, the selected LRUs lead to an architecture with maximum MTBF in terms of hours. The allocated LRUs from the database for maximum reliability are shown in table 23.

Table 23 Allocated LRUs for Max MTBF in LSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC
Selected LRUs ID	DADC	AH1000	CMA3024	ARN147	KRA405	UNS1ESPW	G3X	CMA6800	HGS	EFA
Avionics LRUs	IAC	WRX	TCAS	EFB	FDR	SATCOM	VHF			
Selected LRUs ID	CMA5000	RTA4100	T3CAS	CMA1310	CVFDR	CMA2200SB	VHF4000			

The selected LRUs for minimum weight and maximum MTBF are almost different which shows that these two criteria are conflicting and need to be handled by multi-objective optimisation approaches.

The above problem has also been solved using PSO algorithm where the cost function value and the allocated LRUs are the same as the branch-and-bound algorithm. Figure (5-17) shows the optimal value and/or maximum MTBF for the large scale architecture. In other words, PSO algorithm successfully converges to the optimal value calculated by exact method $z = 257700$ hours. The PSO parameters are also chosen similar to the small scale architecture.

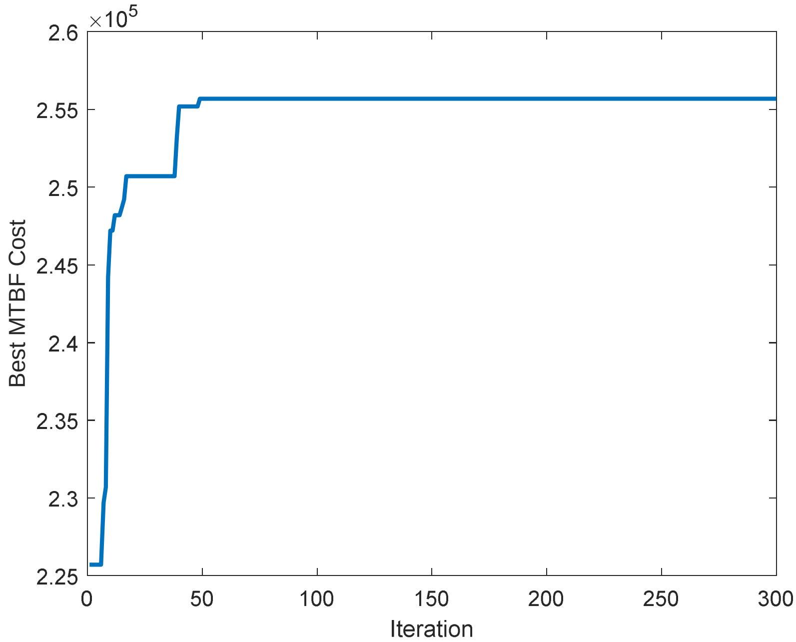


Figure 5-17 Best MTBF Cost for LSA

5.3.5 The Architecture with Maximum Operational Capability for LSA

The maximum operational capability architecture for the avionics architecture (LSA) is simulated based on the cost function defined in equation (4-12) while satisfying constraints from (a-h). The results of the optimisation problem for this scenario based on MATLAB “INTLINPROG” is as follows. It should be noted that the “INTLINPROG” is basically designed for minimisation problem, therefore, for maximisation problem the objective function must be multiplied by “-1”.

The operational capability of the architecture and/or the amount of cost function for seventeen avionics nodes and/or LRUs is $z = 17.3$. In other words, the selected LRUs lead to an architecture with the highest operational capability. The allocated LRUs from the database for maximum operational capability are shown in table 24.

Table 24 Alocated LRUs for Max Operational Capability in LSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC
Selected LRUs ID	DADC	LCR100N	CMA3024	ANV241A	DRA	UNS1ESPW	G3X	EFI890	Aviguide	UKB501
Avionics LRUs	IAC	WRX	TCAS	EFB	FDR	SATCOM	VHF			
Selected LRUs ID	FV4000	RDR7000	T3CAS	CMA1310	HFR5	SAT6100	SRT700			

The proposed avionics architecture is an open system architecture that meets the essential CNS/ATM requirements and its navigational performance provides crew GBAS CAT I, SBAS LPV, ILS CAT II approach capability, vertical navigation, B-RNAV, RNP down to 0.3, and is RVSM compliant. It enables an all-weather operations by RDR 7000. The T3CAS LRU also enables four functions in one LRU including TCAS II 7.1, TAWS, Transponder mode S, and ADS-B. Moreover, The EFB and HUD increase situational awareness by being connected to FMS and GPS.

The above problem has also been solved by PSO algorithm where the results in terms of both cost function and the allocated LRUs are the same as branch-and-bound algorithm. Figure 5-18 illustrates that PSO successfully converges to the optimal value $z = 17.3$ which was already solved by exact method. The PSO parameters are chosen similar to the small scale architecture.

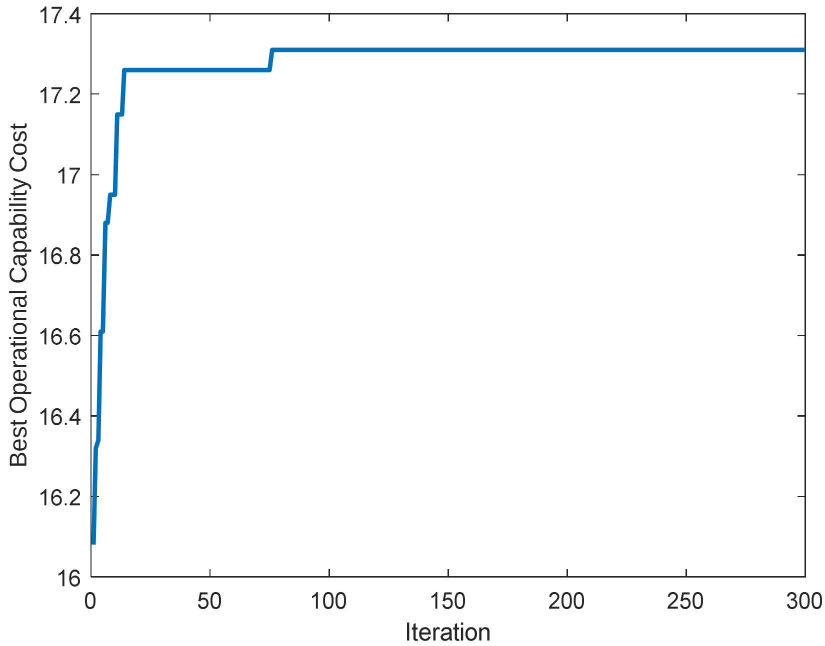


Figure 5-18 Best Operational Capability Cost for LSA

5.4 Multi-Objective Optimisation of the Large Scale Architecture

In previous sections, it has been clear that weight and reliability as well as weight and operational capability are conflicting criteria as the selected LRUs are almost different in each scenario for the large scale architecture. Thus, to achieve an optimised avionics architecture with minimum weight and maximum reliability or minimum weight and maximum operational capability needs a multi-objective optimisation approach. Here, Particle Swarm Optimisation (PSO) algorithm and WSM is used for avionics architecture trade-offs. In what follows two scenarios are run.

5.4.1 MOPSO for Minimum Weight and Maximum Reliability in LSA

The avionics LRU assignment problem for optimisation of avionics architecture in terms of weight and reliability is defined as a multi-objective optimisation problem since weight and reliability are two conflicting criteria. Here, PSO algorithm is used to solve the problem. The cost functions are defined using the weight sum approach by combination of equation (4-8) and (4-11) for weight minimisation and reliability maximisation respectively. The results of this optimisation problem based on PSO algorithm is as follows.

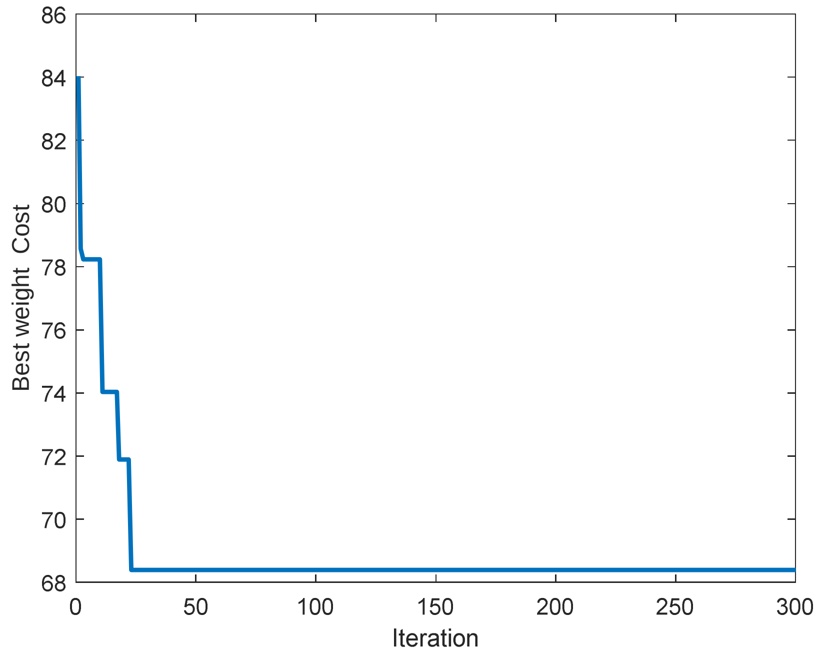


Figure 5-19 Best Weight Cost Based on Pareto Judgement for LSA

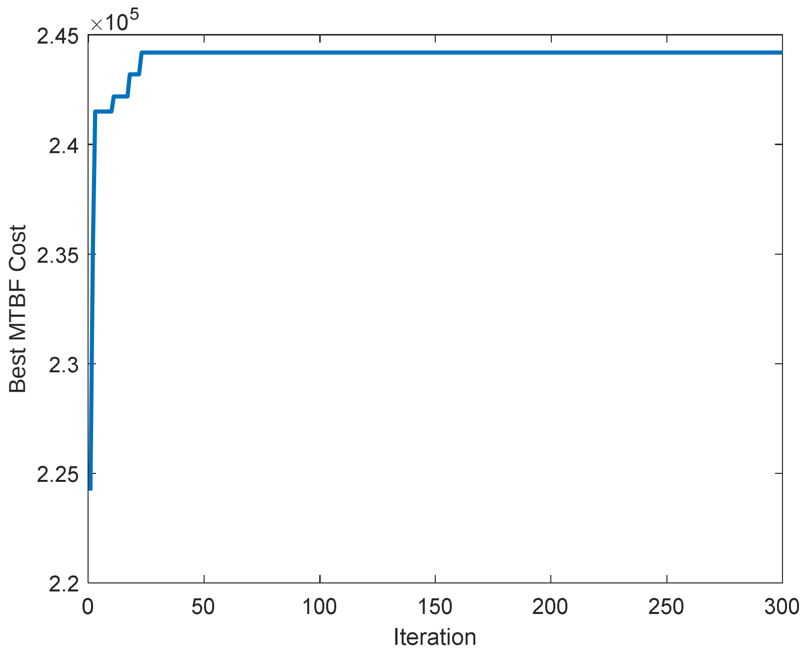


Figure 5-20 Best MTBF Cost Based on Pareto Judgement for LAS

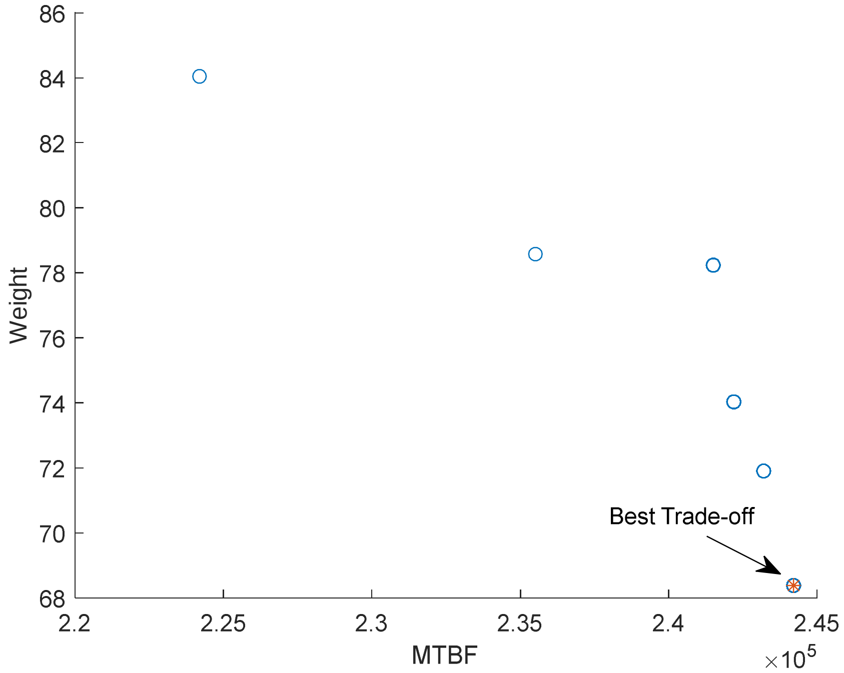


Figure 5-21 Pareto Optima for Min weight and Max MTBF in LSA

Figures 5-19 and 5-20 represent the optimal value of minimum weight and maximum MTBF in Pareto-optimality respectively using PSO. In other words, the best trade-off architecture happens at $w = 68.39$ Kg and $MTBF = 244200$ Hours which is shown in figure 5-21. The allocated avionics LRUs for minimum weight and maximum MTBF in LSA is shown in table 25.

Table 25 Allocated LRUs in Min Weight & Max MTBF for LSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC
Selected LRUs ID	DADC	AH1000	CMA3024	ARN147	KRA405	UNS1ESPW	ISFD	CMA6800	HGS	EFA
Avionics LRUs	IAC	WRX	TCAS	EFB	FDR	SATCOM	VHF			
Selected LRUs ID	PU3000	RDR7000	T2CAS	CMA1310	HFR5	CMA2200SB	SRT700			

The above problem has also been solved by weighted sum methods i.e. the two cost functions f_4 for maximum MTBF and f_1 for minimum weight defined in chapter 4 are weighted using the mean of MTBF over weight $w_2 = \frac{MTBF}{Weight} = 3630$, for the weight cost function, to obtain trade-off architecture. The weighting factor for MTBF is defined as $w_1 = 1$. This is to obtain trade-off architectures. The following figures represents the outcomes.

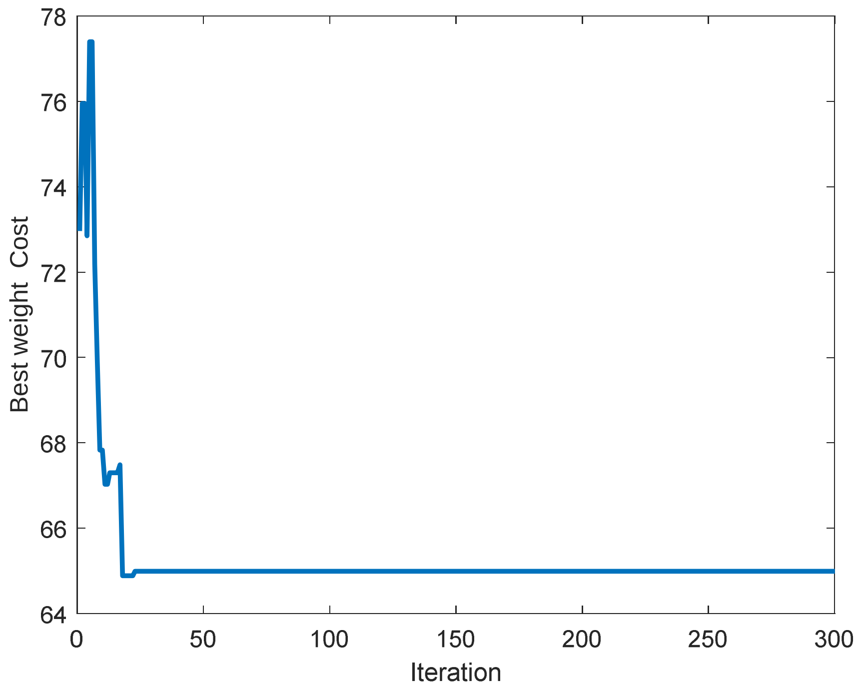


Figure 5-22 Best weight Cost for LSA Based on WSM

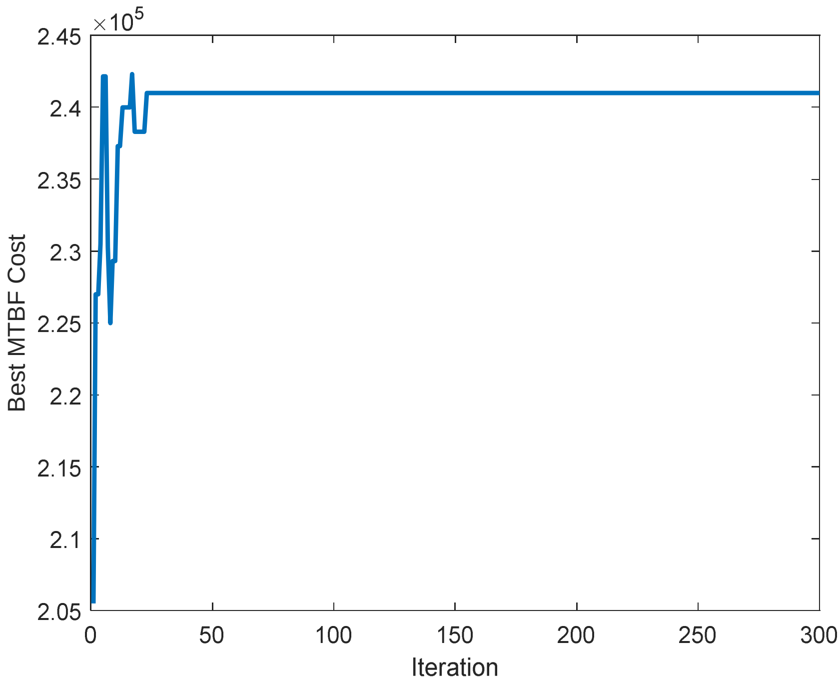


Figure 5-23 Best MTBF Cost for LSA Based on WSM

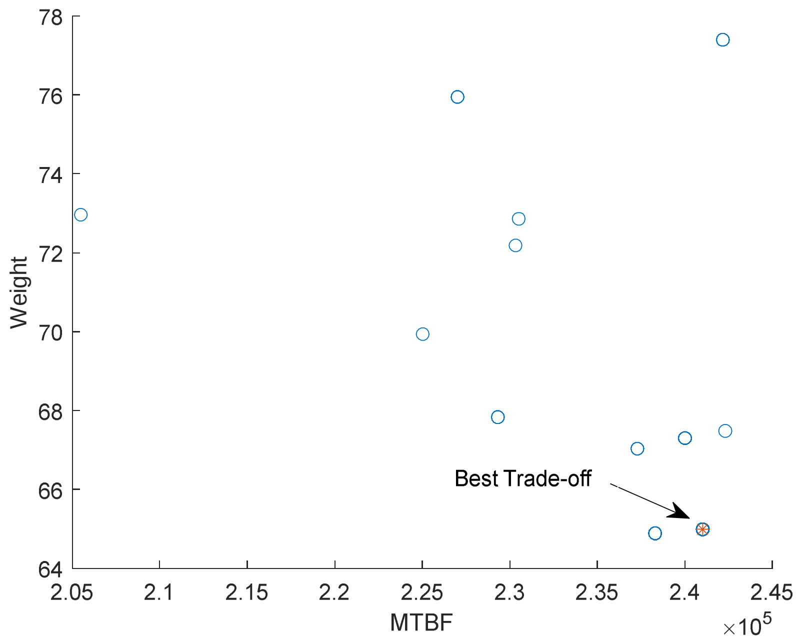


Figure 5-24 Pareto Optima for LSA Based on WSM

Figures 5-22 and 5-23 show the minimum value for weight and maximum value for MTBF achieved by PSO algorithms using weighted sum method. In other words, the best trade-off architecture happens at $w = 64.99$ Kg and $MTBF = 244000$ Hours. The weight of the architecture is 3.4 Kg lighter than the architecture in Pareto optima in that the weight cost function weighed more than MTBF cost function. The MTBF of the architecture is 200 hours less than the Pareto Optima. The allocated avionics LRUs for minimum weight and maximum MTBF in weighted sum method for LSA is shown in table 26.

Table 26 Allocated LRUs for Min Weight & Max MTBF in LSA Using WSM

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC
Selected LRUs ID	DADC	AH1000	CMA3024	ARN147	KRA405	UNS1FW	EFD750	MFDtr	HGS	UKB501
Avionics LRUs	IAC	WRX	TCAS	EFB	FDR	SATCOM	VHF			
Selected LRUs ID	PU3000	RDR7000	T2CAS	CMA1310	CVFDR	CMA2200SB	SRT700			

5.4.2 MOPSO for Min Weight and Max Operational Capability in LSA

The avionics LRU assignment problem for optimisation of avionics architecture in terms of weight and operational capability is defined as a multi-objective optimisation problem since weight and operational capability are two conflicting criteria. Here, PSO algorithm and weighted sum method are used to solve the problem.

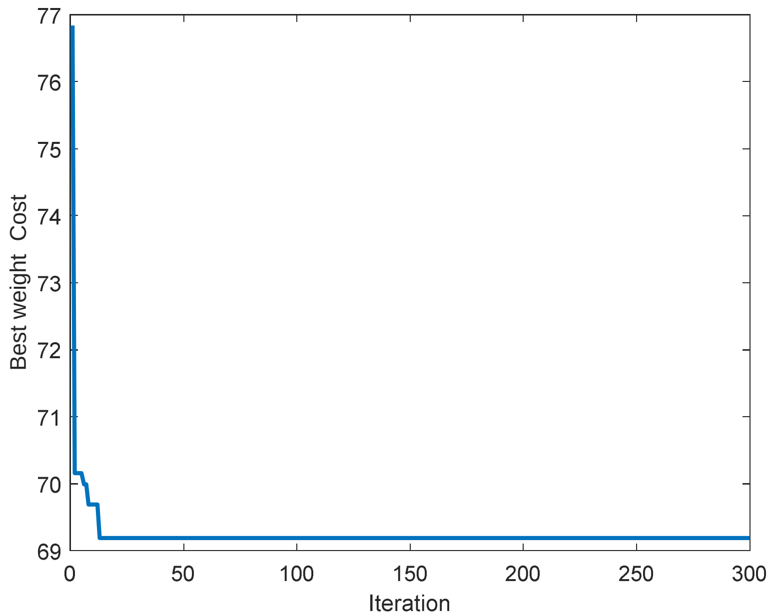


Figure 5-25 Best weight Cost for LSA Based on Pareto Judgement

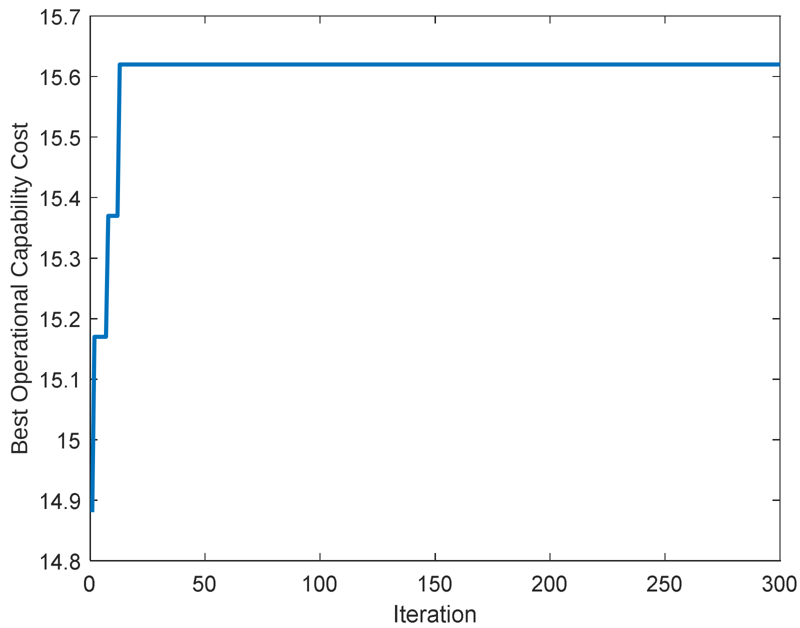


Figure 5-26 Best OC for LSA based on Pareto Judgement

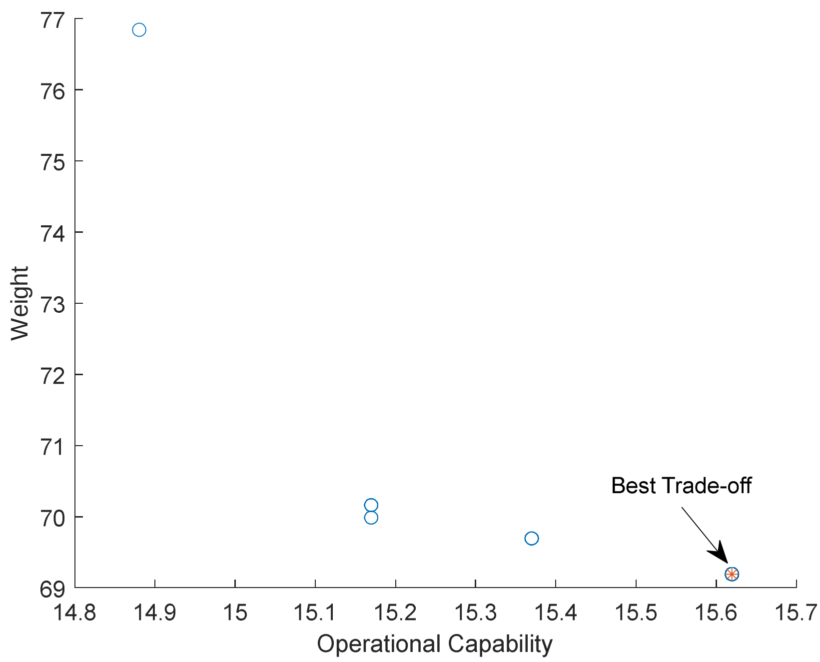


Figure 5-27 Pareto Optima for Min weight and Max OC for LSA

Figures 5-25 and 5-26 illustrate the optimal value for minimum weight and maximum operational capability respectively by using PSO algorithm. In other words, the best trade-off architecture happens at $w = 69.8$ Kg and $OC = 15.62$

which is shown is figure 2-27. The allocated avionics LRUs for minimum weight and maximum operational capability in LSA is shown in table 27.

Table 27 Allocated LRUs for Min Weight & Max OC for LSA

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC
Selected LRUs ID	DADC	LCR100N	CMA5024	ANS	DRA	UNS1FW	ISFD	CMA6800	HUD2020	UKB501
Avionics LRUs	IAC	WRX	TCAS	EFB	FDR	SATCOM	VHF			
Selected LRUs ID	PU3000	RDR7000	T3CAS	CMA1310	LDR	CMA2200SB	SRT700			

The proposed avionics architecture in minimum weight and maximum operational capability is not that different with maximum operational capability in terms of the operational capability. The only difference is the weight of the architecture. Its navigational performance provides crew GBAS CAT I, SBAS LPV, ILS CAT II approach capability, vertical navigation, B-RNAV, RNP down to 0.3, and is RVSM compliant. It enables an all-weather operations by RDR 7000. The T3CAS LRU also enables four functions in one LRU including TCAS II 7.1, TAWS, Transponder mode S, and ADS-B. Moreover, The EFB and HUD increase situational awareness by being connected to FMS and GPS.

The above problem has also been solved by weighted sum methods i.e. the two cost functions f_5 for maximum operational capability and f_1 for minimum weight defined in chapter 4 are weighted to obtain trade-off architecture. The weighting factor for operational capability is defined as $w_1 = 1$ and $w_2 = 5$ for the weight cost function. This is chosen based on decision-maker's preference which shows that the operational capability has the lowest important. In other words, the least important objective receives a weight of one, and integer weights with consistent increments are assigned to objectives that are more important. The relative importance of weight cost function is dedicated higher than operational capability. The following figures represents the outcomes.

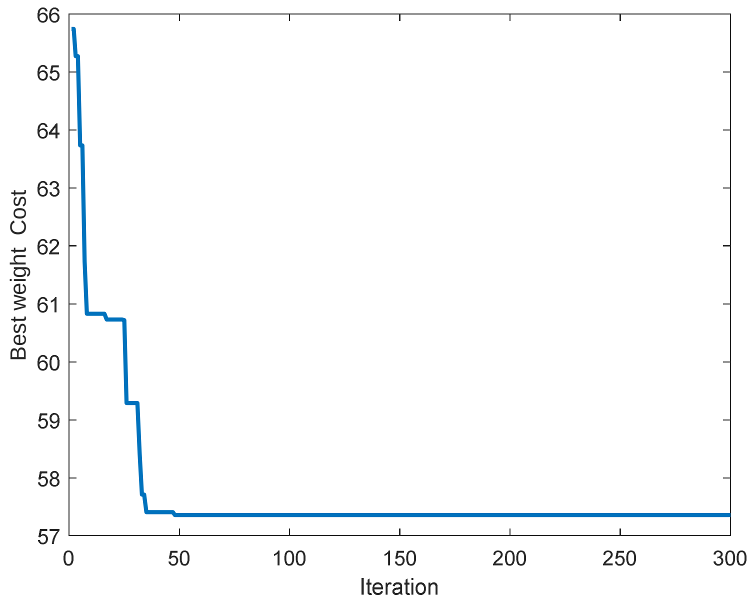


Figure 5-28 Best Weight Cost for LSA Based on WSM

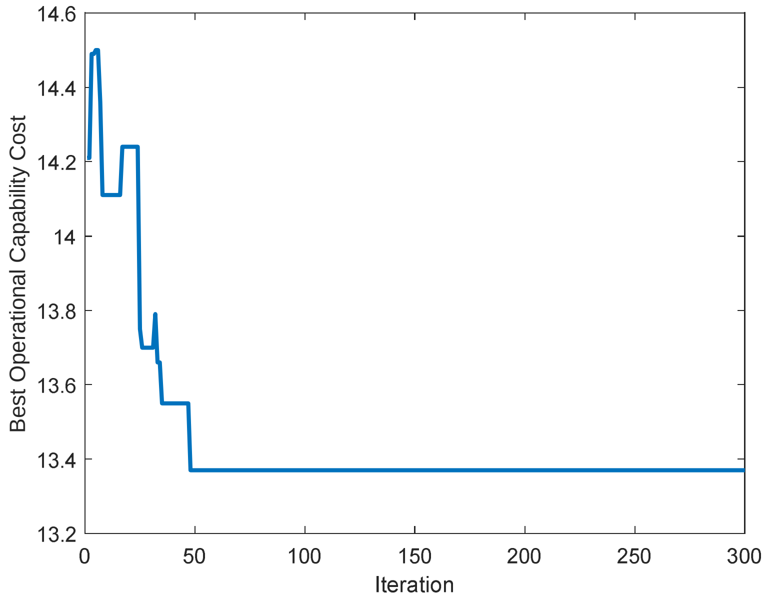


Figure 5-29 Best Operational Capability cost LSA Based on WSM

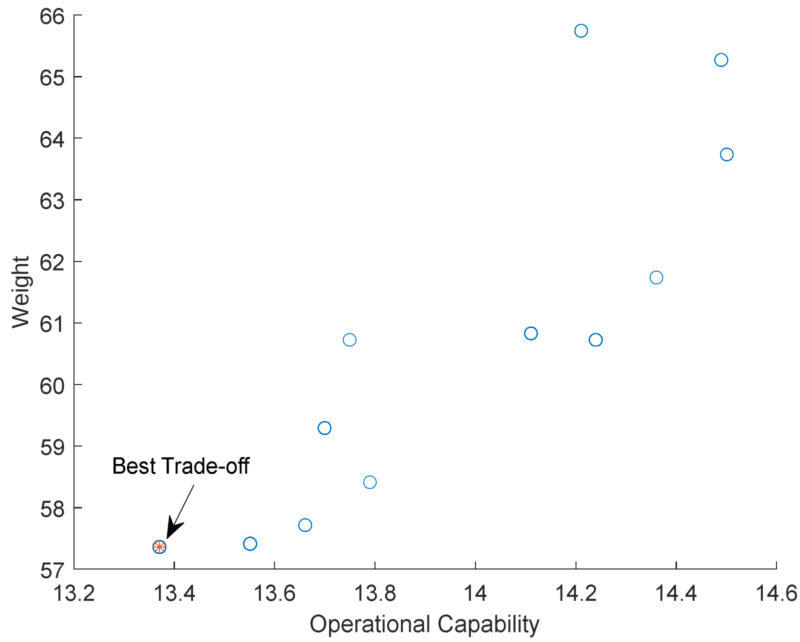


Figure 5-30 Pareto Optima for Min weight and Max OC for LSA

Figures 5-28 and 5-29 show the optimal value i.e. minimum weight and maximum operational capability using PSO algorithm in weighted sum method. In other words, the best trade-off architecture happens at $w = 57.5$ Kg and $OC = 13.42$ which is shown in figure 5-30. The weight of the architecture is 12.3 Kg lighter than the Pareto optima architecture in that the weight cost function weighed more than the Operational capability cost function. Therefore, the operational capability of the WSM architecture is 2.2 less than the Pareto optima. Based on the selected LRUs, this means lower approach capability from ILS CATII to ILS CATI landing capability. The allocated avionics LRUs for minimum weight and maximum Operational Capability in weighted sum approach for LSA is shown in table 28.

Table 28 Allocated LRUs for Min Weight & Max OC for LSA Using WSM

Avionics LRUs	ADC	AHRS	GPS	VORILS	RA	FMS	PFD	MFD	HUD	FCC
Selected LRUs ID	ADUM	AH1000	GPSWAAS	ARN147	KRA405	UNS1FW	EFD750	CMA6800	HUD2020	UKB501
Avionics LRUs	IAC	WRX	TCAS	EFB	FDR	SATCOM	VHF			
Selected LRUs ID	PU3000	RDR7000	T2CAS	CMA1310	LDR	CMA2200SB	SRT700			

5.5 Allocation of Avionics Nodes to Installation locations

In this section, the allocated LRUs to each avionics node are mapped to their installation locations. Based on the cost function defined in chapter 4 i.e. the equation (4-22) the objective is to minimise the architecture weight while satisfying constraint from (4-24) to (4-31). The problem is modelled in GAMS and solved by using branch-and-band algorithm. Figure (5-31) shows the results in GAMS environment.

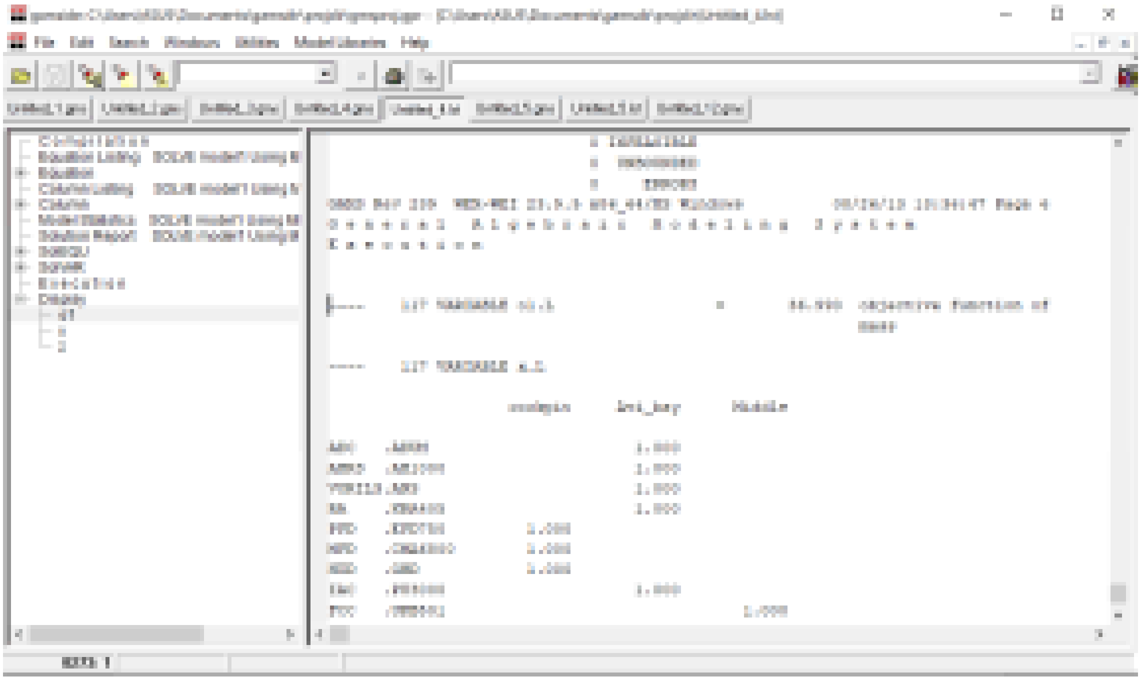


Figure 5-31 Mapping Avionics LRUs to Installation Locations (GAMS)

As it is observed, the cost function value for minimum weight architecture is the same as the value by INTLINPROG and PSO algorithm i.e. $z = 36.99$ Kg for the small scale architecture. Avionics LRUs based on the model defined in chapter 4 are mapped to their installation locations. The number 1 shows that the Avionics LRU is assigned to that location. The PFD, MFD, HUD, and FMS are located in Cockpit. The FCC is located in the Middle location, and the others including ADC, AHRS, GPS receiver, VORILS, RA, and IAC are located in the avionics bay. The associated ID of each assigned avionics LRU for weight minimization is also reported on the left side.

6 Conclusion and Future Work

6.1 Introduction

This chapter is to summarise the research undertaken in this thesis. It includes the major contributions, conclusions of the simulations as well as the recommendations for future studies.

6.2 Contributions

The main contributions of this thesis are outlined as follows

a) The development of a database of avionics LRUs from various venders with their technical specifications including their physical and performance (operational capability) features
b) The Operational Capability Assessment (OCS) of each avionics LRU against a set of criteria using SAW method
c) Capturing aircraft level avionics requirements from an LRU perspective. This is to ensure that the proposed avionics architecture meets the minimum requirement of the CNS/ATM as well as improving and/or upgrading a legacy architecture. The focus in this research was on regulatory point of view.
d) Proposing an Integrated Modular Avionics system architecture (not fully integrated) by using the existing avionics LRUs from the developed avionics database
e) The mathematical modelling of the proposed avionics system architecture using linear programming foundations. Defining the avionics system architecture design as an assignment problem in an integer programming form i.e. to allocate the best possible avionics LRUs to the proposed architecture in order to optimise and improve the operational capability of the architectures while satisfying a number of design constraints. The particular cost functions defined here in this thesis for the first time are minimum weight architecture, minimum volume architecture, minimum power consumption, maximum MTBF architecture, and maximum operational capability architecture as a single-objective optimisation

problems. Further, minimum weight and maximum MTBF architectures and minimum weight and maximum operational capability architectures are defined as multi-objective optimisation problems for trade-off studies.

f) Using three different optimisation algorithms to solve the problems including branch-and-bound, weighted sum method and multi-objective PSO algorithms as well as COTS-solver, GAMS.

6.3 Conclusion

The avionics LRUs assignment problem defined in this research has created a general method to investigate and evaluate avionics architectures at the early stage of design for various possibilities including minimum weight, minimum volume, minimum power consumptions, maximum MTBF and maximum operational capability as well as trade-off architectures while keeping the design constraints within limits. This method can also be used to upgrade a legacy avionics architecture to a particular operational capability and/or CNS/ATM requirements. Figure (6-1) shows the results of the various investigated avionics architectures (the large scale). The single-objective optimisation problems are solved by branch-and-bound algorithm and PSO algorithm and the installation location problem is modelled and solved by GAMS. The cost values and the allocated LRUs in both methods are the same for minimum weight, maximum MTBF and maximum operational capability architectures. The minimum weight architecture is different from maximum MTBF and maximum operational capability. This proves that weight and MTBF as well as weight and operational capability are conflicting criteria and need to be solved by a multi-objective approach. Thus, two methods including PSO and WSM are used to solve these problems.

The result architectures for the PSO and WSM are not that different, however, the WSM depends heavily on the weighting factors defined in each case. In this thesis, the weighting factor were determined by decision's maker perspective (the author), nevertheless, for industrial projects, weight factor can be determined by the system design team considering customers and stakeholders requirements. Therefore, the Multi-objective PSO (MOPSO) algorithm to gain the Pareto optima is realised to be the best method for avionics architecture trade-offs optimisation in this research. It is worth mentioning that, the proposed method and tool remained at simulation/algorithm level and for validation and verification of which a reference architecture (real aircraft avionics systems architecture) is needed. Finally, the proposed method and optimisation algorithms provide a tool for avionics systems architect to investigate various architectures at preliminary design phase to optimise avionics architecture and improve its operational capability from an LRU perspective. It can also help engineers to upgrade a legacy avionics architecture.

Avionics LRUs	Selected LRUs	Selected LRUs	Selected LRUs	Selected LRUs	Selected LRUs	Selected LRUs	Selected LRUs	Selected LRUs	Selected LRUs
Scenarios	Min Weight	Min Power	Min Volume	Max MTBF	Max OC	Min Weight and Max MTBF PSO	Min Weight and Max MTBF WSM	Min weight and Max OC PSO	Min weight and Max OC WSM
ADC	ADUM	ADUM	AMADC	DADC	DADC	DADC	DADC	DADC	ADUM
AHRS	AH1000	AH1000	AH1000	AH1000	LCR100N	AH1000	LCR100N	LCR100N	AH1000
GPS Receiver	GPSWAAS	GPSWAAS	GPSWAAS	CMA3024	CMA3024	CMA3024	CMA5024	CMA5024	GPSWAAS
VORILS	ANS	ANS	ANS	ARN147	ANV241A	ARN147	ANS	ANS	ARN147
RA	KRA405	KRA405	KRA405	KRA405	DRA	KRA405	DRA	DRA	KRA405
FMS	UNS1FW	GE6883	CMA9000	UNS1ESPW	UNS1ESPW	UNS1ESPW	UNS1FW	UNS1FW	UNS1FW
PFD	EFD750	ISFD	ProPFD	G3X	G3X	ISFD	ISFD	ISFD	EFD750
MFD	CMA6800	CMA6800	CMA6800	CMA6800	EFI890	CMA6800	CMA6800	CMA6800	CMA6800
HUD	GHD	GHD	GHD	HGS	Aviguide	HGS	HUD2020	HUD2020	HUD2020
FCC	UKB501	UKB501	UKB501	EFA	UKB501	EFA	UKB501	UKB501	UKB501
IAC	PU3000	PU3000	PU3000	CMA5000	FV4000	PU3000	PU3000	PU3000	PU3000
WXR	RDR7000	RDR7000	RDR7000	RTA4100	RDR7000	RDR7000	RDR7000	RDR7000	RDR7000
TCAS	T2CAS	T2CAS	T2CAS	T3CAS	T3CAS	T2CAS	T3CAS	T3CAS	T2CAS
EFB	CMA1310	CMA1310	CMA1310	CMA1310	CMA1310	CMA1310	CMA1310	CMA1310	CMA1310
FDR	LDR	LDR	LDR	CVFDR	HFR5	HFR5	LDR	LDR	LDR
SATCOM	CMA2200SB	CMA2200SB	IRT2110	CMA2200SB	SAT6100	CMA2200SB	CMA2200SB	CMA2200SB	CMA2200SB
VHF	SRT700	SRT700	SRT700	VHF4000	SRT700	SRT700	SRT700	SRT700	SRT700

Figure 6-1 Comparison of Avionics Architectures from LRUs Perspective

6.4 Future Work

The proposed method in this research can be extended in many aspects. First of all, the developed avionics database can be extended in terms of the number of LRUs for each avionics function. Currently, there are at least three and maximum five LRUs for each avionics function. Increasing the number of LRUs for each function would result in different possibilities and a larger exploration space for system architects. Further, the technical specifications recorded can also be expanded to a more detailed criteria of each avionics LRU like CPU and memory for considering them in a network. Also, the relationship of each avionics LRU (functional links) can be determined in the database for network optimisation problem. This needs that the mathematical model should be extended to a network topology and routing optimisation problem. The network optimisation would then need new constraints to be satisfied like network bandwidth and real-time scheduling constraints. The network topology and routing optimisation can complete the proposed method in this research by reduction of wiring needed with regards to the installation location of avionics LRUs and their distance from the associated sensors and/or actuators.

Moreover, the method can also be extended for the comparison of different field bus technologies to optimise the field bus networks and integrate this optimisation into the whole avionics architecture. Also, a more detailed requirements like requirements on avionics flight reliability and safety model in the form of Functional Failure Analysis (FFA) and/or Functional Hazard Analysis (FHA) at aircraft level as well as interface and network requirements are proposed for future work. The last but not least, new optimisation algorithms can also be tested to solve the above problems like Ant Colony Optimisation (ACO) or Simulated Annealing (SA) to compare the accuracy and performance of these algorithms in handling discrete combinatorial optimisation problems.

REFERENCES

[1] J. R. Newport, Avionic Systems Design, CRC Press, 1994.

[2] R. Collinson, Introduction to Avionics Systems, Third Edition, Springer, 2011.

[3] “NASA System Engineering Handbook, NASA/SP-2007-6105, Rev 1, Dec 2007,” NASA, 2007.

[4] P. O. L. d. Weck, “ Fundamental of System Engineering, MIT Lecture Notes,” MIT, 2015.

[5] “INCOSE System Engineering Handbook, a Guide for System Lifecycle Processes and Activities, INCOSE-TP-2003-002-03, version 3,,” International Council on Systems Engineering (INCOSE), June 2006 – version 4 was just issued in July 2015.

[6] A. Seabridge and I. Moir, Design and Development of Aircraft Systems, Third Edition, Wiley, 2020.

[7] D. H. Middleton, Avionics Systems, Longman Scientific and Technical, 1989.

[8] I. Moir and A. Seabridge, Military Avionics Systems, Wiley, 2006.

[9] I. Moir and A. Seabridge, Civil Avionics Systems, Second Edition, Wiley, 2013.

[10] C. R. Spitzer and W. G. Ferrell, Digital avionics handbook,, Boca Raton : CRC Press, 2015.

[11] B. Annigofer, Model-based Architecting and Optimisation of Distributed Integrated Modular Avionics, Shaker Verlag, 2015.

[12] E. R. R. K. Neumann, “Quantitative Evaluation Criteria for Modern Avionics System Architectures,” in *Deutscher Luft- und Raumfahrtkongress*, 2008.

[13] A. GmbH, “ARINC429 Specification Tutorial, Sasbacher Str. 2,” Freiburg, Germany , 2010.

[14] A. R. I. (ARINC), “629 Part 1-5- Multi-transmitter Data Bus, Part 1- Technical Description ARINC STD,” Aeronautical Radio Incorporated (ARINC), 1996.

[15] G. F. E. Systems, “AFDX/ARINC 664 Protocol Tutorial,” 2007.

[16] R. Bosch, “CAN Specification- Version 2,” Robert Bosch GmbH, 1991.

[17] H. Kopetz and G. Grunsteidl, “TTP-A Time-Triggered Protocol for Fault Tolerant Real-Time Systems,” in *Fault Tolerant Computing*, 1993.

[18] H. Butz, “Open Integrated Modular Avionic (IMA) : State of the Art and Future Development Road Map at Airbus Deutschland,” in *In Proceedings of the First International Workshop on Aircraft Systems Technologies, pp.211-222*, March 2007.

[19] G. Wang and W. Zhao, The Principles of Integrated Technology in Avionics Systems, Elsevier, 2020.

[20] P. E. James D. Nickum and R. H. Cormier, “Avionics Integration: A Process to Optimise Avionics Components to Meet Desired Operational Capabilities,” in *AIAA's Aircraft Technology, Integration, and Operations (ATIO)*, Los Angeles, California, 2002.

[21] O. M. G. (OMG), “https://www.omg.org/spec/UML/About-UML/,” OMG Std. [Online]. [Accessed March 2019].

[22] O. M. G. (OMG), “https://www.omg.org/spec/SysML/About-SysML/,” OMG Std. [Online]. [Accessed March 2019].

[23] AADL, "aadl.info," [Online]. [Accessed March 2017].

[24] P. H. Feiler, D. P. Gluch and J. J. Hudak, "The Architecture Analysus and Design Language (AADL): An Introduction," Carnegie Mellon University, Tech Report, 2006.

[25] Topcased, "The Open-source Toolkit for Critical Systems," [Online]. Available: Topcased.org.

[26] C. FRABOUL and F. MARTIN, "Modeling Advanced Modular Avionics Architectures for Early Real-time Performance Analysis," in *Proceedings of the Seventh Euromicro Workshop on Parallel and Distributed Processing*, 1999.

[27] J. Delange, A. Plantec and F. Kordon, "Validate, Simulate and Implement ARINC653 Systems Using the AADL," in *Proceedings of the ACM SIGAda International Conference*, Newyork, 2002.

[28] M. Lafaye, M. Gatti and David Faura, "Model Driven Early Exploration of IMA Execution Platform," in *in DASC conference* , 2011.

[29] S. F¨orster, M. Fischer, A. Windisch, B. Balser and D. Monjau, "A New Specification Methodology for Embedded Systems Based on the π-CALCULUS Process Algebra," in *Proceedings of the 14th IEEE International Workshop on Rapid Systems Prototyping*, 2003.

[30] H. Charara, J.-L. Scharbarg and J. Ermount, "Methods for Bounding End-to-end Delays on an AFDX Network," in *Real-time Systems, Euromicro Conference* , 2006.

[31] M. Lauer and J. Ermont, "Latency and freshness analysis on IMA systems," in *main technical program at IEEE ETFA*, 2011.

[32] M. D. A. a. S. M. R. Burkard, Assignment Problems, Society for Industrial and Applied Mathematics (SIAM), 2009.

[33] S. Bokhari, Assignment Problems in Parallel and Distributed Computing, DeGroot , 1987.

[34] V.Lo, “Heuristic Algorithms for Task Assignment in Distributed Systems,” in *Computers, IEEE Transaction on, Vol. 37, no 11, pp. 1384-1397*, 1988.

[35] M. Kafil and I. Ahmad, “Optimal Task Assignment in Hetrogeneous Computing Systems,” in *Hetrogeneous Computing Workshop*, 1997.

[36] R. R. U Salomon, “Automatic Design of IMA Systems,” in *IEEEAC*, 2013.

[37] R. R. U Salomon, “Automatic Safety Computation for IMA Systems,” in *IEEE*, 2011.

[38] B. Dougherty, D. Schmidt, J. White and R. Kegley, “Deployment Optimisation for Embedded Flight Avionics Systems,” in *CrossTalk*, 2011.

[39] P. Manolis, G. Subramanian and D. Vroon, “Automating Component-Based System Assembly,” in *ACM 978-1*, London, 2007.

[40] X. Shi and C. Zhang, “Integration Design Tool for Avionics System Based on Mathematical Programming,” in *IEEE*, 2016.

[41] C. Zhang and J. Xiao, “Modelling and Optimisation in Distributed Integrated Modular Avionics,” in *34th Digital Avionics Systems Conference* , 2013.

[42] H. Salzwedel and N. Fischer, “Aircraft Level Optimisation of Avionics Architecture,” in *27th Digital Avionics Systems Conference* , 2008.

[43] B. Annighofer, E. Kleemann and F. Thielecke, “Model-Based Development of Integrated Modular Avionics Architecture on Aircraft Level,” in *Digital Avionics Systems Conference* , 2014.

[44] B. Annighofer and F. T. E. Kleemann, “Automated Selection, Sizing, and Mapping of Integrated Modular Avionics Modules,” in *32nd Digital Avionics Systems Conference* , 2013.

[45] B. Annighofer, E. Kleemann and F. Thielecke, “Network Topology Optimisation for Distributed Integrated Modular Avionics,” in *33rd Digital Avionics Systems Conference* , 2014.

[46] Y. Lee and M. Younis, “Scheduling tool and algorithm for Integrated Modular Avionics Systems,” in *IEEE*, 2000.

[47] S. Zhang, “Communication Infrastructure Supporting Real-Time Applications, PhD Thesis,” Tchnical University of Hamburg , 2008.

[48] C. Bauer, K. Lagadec and M. M. christian Bes, “Flight Control System Architecture Optimisation for Fly-By-Wire Airliners,” *Journal of Guidance, Control and Dynamics ,* vol. 30, no. 4, 2007.

[49] B. Jackson and J. Norgard, “A Stochastic Optimisation Tool for Determing Spacecraft Avionics Box Placement,” in *IEEE*, 2001.

[50] F. Sousa, R. Louiz, E. Marconi and J. Calos, “A Tool for Multidisciplinary Design Conception OF Spacecraft Equipment Layout,” in *22nd International Congress of Mechanical Engineering (COBEM 2013)*, Brazil, 2013.

[51] J. Yang, X. Chen and W. Yao, “A Rectangular Cuboid Satellite Module Layouts Method Based on Integer Optimisation,” in *6th International Conference on Advanced Design and Manufacturing Engineering* , 2016.

[52] N. R. Dayama, K. Todi, T. Saarelainen and A. Oulasvirta, “GRIDS: Interactive Layout Design with Integer Programming,” in *Association for Computing Machinery*, Honolulu, HI, USA., 2020.

[53] B. Annighöfer, Ç. Nil, J. Sebald and F. Thielecke, “ARIANE-5-based Studies on Optimal Integrated Modular Avionics Architectures for Future Launchers,” in *6TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)*, 2015.

[54] B. Annighoefer, C. Nil, J. Sebald and F. Thielecke, “Structured and Symmetric IMA Architecture Optimization: Use Case ARIANE Launcher,” in *34th Digital Avionics Systems Conference (DASC)*, 2015.

[55] PREEvision, “Vector.com,” Vector, [Online]. Available: https://www.vector.com/int/en/products/products-a-z/software/preevision/#c1789. [Accessed 2020].

[56] B. Hardung, “Optimisation of the Allocation of Functions in Vehicle Networks, PhD Thesis,” Erlangen University, Erlangen, 2006.

[57] H. Andersson, “Aircraft Systems Modeling: Model Based Systems Engineering in Avionics Design and Aircraft Simulation,” linkoping university, Linkoping, 2009.

[58] H. Andersson and Sundkvist, “Method and Integrated Tools for Efficient Design of Aircraft Control Systems,” in *25th International Congress of the Aeronautical Sciences*, 2006.

[59] N. Suh, Axiomatic Design Advances and Applications, Newyork, USA: Oxford University Press, 2001.

[60] H. Gavel, “On Aircraft Fuel Systems: Conceptual Design and Modelling. Dissertation,” Linkoping University , 2007.

[61] O. Johansson and P. Krus, “Configurable Design Matrixes for System Engineering Applications,” in *In the proceeding ofIDETC/CIE ASME*, USA, 2006.

[62] O. Johansson, H. Andersson and P. Krus, “Conceptual Design Using Generic Object Inheritence,” in *In Proceedings of the ASME International Design Engineering Technical Conference*, Newyork, 2008.

[63] C. Fraboul and F. Martin, “Modelling and Simulation of Integrated Modular Avionics,” in *in Proceedings of the Six Euromicro Workshop on Parallel and Distributed Processing*, 1998.

[64] R. Weber, E. Thaden and P. Reinkemeier, “Specification of an Architecture Meta-Model,” SPES, Report D3.2C_TR, 2012.

[65] F. Minot, “Method for Optimisation of an Avionics Platform”. Patent US 2010/0292969 A1, 17 May 2010.

[66] L. Galli, “Algorithms for Integer Programming,” in *IEEE*, 2018.

[67] P. A. Jensen, Operation Research Models and Methods, Wiley , 2003.

[68] C. J. Guerra, N. E. Carmichael and J. T. Nielson, “Aircraft Avionics Strategic Fleet Update Using Optimal Methods,” in *IEEE*, 2016.

[69] T. Sivanthi, s. Zhang and U. Killat, “A Holistic Framework for Optimal Avionics System Resource Planning,” In AST Workshop on Aircraft System Technology , 2007.

[70] Y. Lee, D. Kim, M. Younis and J. Zhou, “Scheduling Tool and Algorithm for Integrated Modular Avionics Systems,” in *In 19th Digital Avionics Systems Conference* , 2000.

[71] L. Sagaspe and P. Bieber, “Constraint-based Design and Allocation of Shared Avionics Resources,” in *in 26th Digital Avionics Systems Conference*, 2007.

[72] D. Carta, J. D. Oliveira and R. Starr, “Allocation of Avionics Communication Using Boolean Satisfiability,” in *in 31st Digital Avionics Systems Conference*, 2012.

[73] M. Jakovljevic and A. Ademaj, “2nd Generation IMA: Extended Virtualization Capabilities for Optimized Architectures,” in *32nd Digital Avionics Systems Conference (DASC)*, 2013.

[74] C. Viraktamath, G. Peitz, U. Strauss, J. Ressler and O. Strobel, “Evaluation of Software Performance Characteristics of Flexray Communication Protocol for Its Suitability in Avionics,” in *3rd ICTON Mediterranean Winter Conference (ICTON-MW)*, 2009.

[75] S. Görke, R. Riebeling, F. Kraus and R. Reichel, “Flexible Platform Approach for Fly-by-wire Systems,” in *32nd Digital Avionics Systems Conference (DASC)*, 2013.

[76] Y. Maryanka, “Wiring Reduction by Battery Power Line Communication,” in *IEE Seminar on Passenger Car Electrical Architecture (Ref. No. 2000/088)*, 2000.

[77] WAIC, “https://waic.avsi.aero/,” [Online]. [Accessed March 2018].

[78] D. Suo, J. An and J. Zhu, “A New Approach to Improve Safety of Reconfiguration in Integrated Modular Avionics,” in *30th Digital Avionics Systems Conference*, 2011.

[79] P. Bieber, E. Noulard and C. Pagetti, “Preliminary Design of Future Reconfigurable IMA Platform Safety Assessment,” in *ICAS*, 2010.

[80] B. Annighoefer, M. Riedlinger, O. Marquard, R. Ahmadi, B. Schulz, M. Brunner and R. Reichel, “The Adaptive Avionics Platform,” *IEEE Aerospace and Electronic Systems Magazine ,* vol. 34 , no. 3, March 2019.

[81] B. Annighoefer, M. Riedlinger and O. Marquardt, “How to Tell Configuartion-free Integrated Modular Avionics What to Do?,” in *36th Digital Avionics Systems Conference (DASC)*, 2017.

[82] K. Sampigethaya and R. Poovendran, “Cyber-physical Integration in Future Aviation Information Systems,” in *31st Digital Avionics Systems Conference (DASC)*, 2012.

[83] V. Skormin, V. Gorodetski and L. Popyack, “Data Mining Technology for Failure Prognostic of Avionics,” *IEEE Transactions on Aerospace and Electronic Systems ,* vol. 38, no. 2, Apr 2002 .

[84] Y. LeCun, Y. Bengio and G. Hinton, “Deep learning,” *Nature,* p. 436–444, 2015.

[85] Z. Li, Q. Li and H. Xiong, “Avionics Couds: A Generic Scheme for Future Avionics Systems,” in *31st Digital Avionics Systems Conference (DASC)*, 2012.

[86] M. Jakovljevic, C. C. Insaurralde and A. Ademaj, “Embedded Cloud Computing for Critical Systems,” in *33rd Digital Avionics Systems Conference (DASC)*, 2014.

[87] R. Edwards and B. Gleaves, “The Death of Smart Sensors and Effectors for Future Combat Aircraft,” *Microprocessors and Microsystems,* vol. 21, pp. 1-12, 1997.

[88] S. Allan, “Lecture Notes on Avionics Data Networking, Hardware Intergration and Testing,” Cranfield University, 2017.

[89] Airbus, “A319/A320/A321 Flight Deck and Systems Briefing for Pilots,” 1998.

[90] Airbus, “A330/A340 Flight Deck and Systems Briefing for Pilots,” 1999.

[91] Airbus, “A350 Flight Deck and Systems Briefing for Pilots,” 2011.

[92] Airbus, “A380 Flight Deck and Systems Briefing for Pilots,” 2006.

[93] ATR, “ATR 72-600 Maintenance Training Manual,” 2015.

[94] Cessna, “Cessna Citation XLS Instrumentation and Avionics,” 2000.

[95] BAE, “BAE 146/AVRO 146 RJ Seriess, Aircraft Flight Manual,” 2007.

[96] EUROCONTROL, “Avionics Requirements for State Aircraft,” 2017.

[97] EUROCONTROL, “Avionics Requirements for Civil Aircraft,” 2017.

[98] ICAO, “Avionics Requirements for Civil Aircraft,” 2013.

[99] Airbus, “Getting to Grips with FANS (Future Air Navigation System),” 2007.

[100] ICAO, “ICAO Doc 4444: PANS-ATM (Procedures for Air Navigation Services), Fifteenth Edition,” 2007.

[101] ICAO, “Annex 14, Aerodromes, Volume 1 Aerodrome and Operations, Sixth edition,” 2013.

[102] ICAO, “Annex 2, Rules of the Air, Tenth Edition,” 2005.

[103] ICAO, “Annex 10, Aeronautical Telecommunications, Volume 1, Radio Navigation Aids, Sixth Edition,” 2006.

[104] J. Avionics, “Jane's Defence Equipment and Technology,” Janes , 2017.

[105] NASA, “Final Report of the NASA Technology Readiness Assessment (TRA) Study Team,” 2016.

[106] DOE, “Technology Readiness Assessment Guide,” 2015.

[107] DOD, “Technology Readiness Assessment Guide,” 2019.

[108] M. R. Kirby, “Technology Identification, Evaluation and Selection,” Aerospace Systems Design Laboratory, Georgia Institute of Technology, 2002.

[109] E. Triantaphyllou, Multi-criteria Decision Making Methods, A Comparative Study, Springer , 2000.

[110] G. Tzeng and J.-J. Huang, Multiple Attribute Decision Making: Methods and Applications, CRC Press, Taylor & Francis Group , 2011.

[111] H. Zimmermann, Fuzzy Set Theory and Its Applications, Third Revised Edition. Boston. MA. USA., 1996.

[112] S. Chen and C. Hwang, Fuzzy Multiple Attribute Decision Making: Methods, Lecture Notes in Economics and Mathematical Systems, No.375, Sringer-Verlag, Berlin, Germany, 1991.

[113] T. Saaty, "An Exposition of the AHP in Reply to the Paper "Remarks on the Analytic Hierarchy Process," *Management Science,* vol. 36, no. 3, pp. 259-268, 1990.

[114] R. B. R. Benayoun and N. Sussman, "Manual de Reference du Programme Electre," *Note de Synthese et Formaton,* no. 25, 1996.

[115] H. C.L and K. Yoon, Multiple Attribute Decision Making: Methods and Applications, Springer-Verlag, New York, NY, USA., 1981.

[116] D. N. Mavris and O. J. Pinon, "An Overview of Design Challenges and Methods in Aerospace Engineering,," in *Complex Design and Management*, Berlin, Springer , 2012, pp. pp. 1-25.

[117] M. R. Kirby, "A methodology for Technology Identification, Evaluation, And Selection in Conceptual and Preliminary Aircraft Design, PhD Thesis," School of Aerospace Engineering, Georgia Institute of Technology, 2001.

[118] Y. Wang, "Applying MCDM to Evaluate Financial Performance of Domestic Airliners in Taiwan," in *Expert Systems with Applications* , 2008.

[119] F. Meller, "Key Buying Factors and Added Value: A New Approach to Aircraft Evaluation," in *In DGLR Workshop Aircraft Evaluation*, 1998.

[120] G. Driks and F. Meller, "Multidiciplinary Design Optimisation- Enhanced Methodology for Aircraft and Technology Evaluation," in *in 8th AIAA/USAF Symposium on Multidiciplinary Analysis and Optimisation* , Long Beach, CA, 2000.

[121] O. Čokorilo, S. Gvozdenović, P. Mirosavljević and L. Vasov, “Multi Attribute Decision Making: Assessing the Technological and Operational Parameter of Aircraft,” 2010.

[122] P. Konidari and D. Mavrakis, “ A Multi-Criteria Evaluation Method for Climate Change Mitigation Policy,” in *Energy Policy*, 2007.

[123] E. KERRE, “A Call for Crispiness in Fuzzy Set Theory,” *Fuzzy Sets and Systems,* vol. Volume 29, no. 1, pp. Pages 57-65, 1989.

[124] K. Mukherjee and A. Bera, “Application of Goal programming in Project Selection Decision – A Case Sudy fromthe Indian Coal mining industry.,” *European Journal of Operational Research,* pp. 18-25, 1995.

[125] S. TL, “The Analytic Hierarchy Process,” in *McGraw-Hill*, New York, 1980.

[126] R. Sabatini, “MB-339CD Aircraft Development: COTS Integration in a Modern Avionics Architecture,” in *RTO Symposium on "Strategies to Mitigate Obsolescence in Defence Systems"*, Budapest, Hungary, 2000.

[127] R. Gomory, “Outline of an Algorithm for Integer Solutions to Linear Programs,” Bulletin of the American Mathematical Society, 1958.

[128] G. Nemhauser and L. Wolsey, Integer and Combinatorial Optimisation, Wiley, 1988.

[129] W. L. Winston, Operations Research: Applications and Algorithms, Fourth Edition, Thomson Learning, 1998.

[130] M. S. Bazaraa, J. J. Jarvis and H. D. Sherali, Linear Programming and Network Flows, Fourth edition, Wiley, 2010.

[131] F. T. Matthias Krings, “An Integrated Approach to Predective Flight Guidance and Evelope Protection,” in *In AIAA GNC*, MN, USA, 2012.

[132] K. Genova and V. Guliashki, “Linear Integer Programming Methods and Approaches- A Survey,” *CYBERNETICS AND INFORMATION TECHNOLOGIES ,* Vols. Volume 11, No 1, 2011.

[133] M. Ehrgott, Multicriteria Optimisation, Second Edition, Springer , 2005.

[134] X.-S. Yang, Nature-Inspired Optimization Algorithms, Elsevier, 2014.

[135] K. Yoon and C. Hwang, Multiple Attribute Decision Making: An Introduction, London: Sage , 1995.

[136] A. H. Land and A. G. Doig, “An Automatic Method of Solving Discrete Programming Problems,” *Econometrica: Journal of the Econometric Society,* pp. 497-520, 1960.

[137] R. J. Dakin, “A Tree Search Algorithm for Mixed Integer Programming Problems,” *Computer Journal ,* vol. 8, pp. 250-255, 1965.

[138] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” in *Proceedings of IEEE International Conference on Neural Networks IV*, Perth, Western Autralia, 1995.

[139] D. Wang, D. Tan and L. Liu, “Particle Swarm Optimization Algorithm: An Overview,” in *Springer-Verlag Berlin Heidelberg*, 2017.

[140] C. Coello and G. B. Lamont, Evolutionary Algorithm for Solving Multi-objective Problems, Second Edition, Springer , 2006.

[141] C. A. C. Coello, G. T. Pulido and M. S. Lechuga, “Handling Multiple Objectives With Particle Swarm Optimization,” *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION,* vol. 8, no. 3, JUNE 2004.

[142] R.T.Marler and J.S.Arora, “Survey of Multi-objective Optimization Methods for Engineering,” *struct multidisc optim,* no. 26, pp. 369-395, 2004.

APPENDICES

Appendix A MATLAB CODE

A.1 The Architecture with Minimum weight for SSA

The following code is for implementation of the weight minimisation of the small scale architecture

The input file:

```
%Input Files
%% The Mass of each avionics Node for different LRUs
m_node1=xlsread('AFCS_Database.xlsx', 'D2:D6');
m_node2=xlsread('AFCS_Database.xlsx', 'D7:D10');
m_node3=xlsread('AFCS_Database.xlsx', 'D11:D15');
m_node4=xlsread('AFCS_Database.xlsx', 'D16:D18');
m_node5=xlsread('AFCS_Database.xlsx', 'D19:D21');
m_node6=xlsread('AFCS_Database.xlsx', 'D22:D26');
m_node7=xlsread('AFCS_Database.xlsx', 'D27:D30');
m_node8=xlsread('AFCS_Database.xlsx', 'D31:D34');
m_node9=xlsread('AFCS_Database.xlsx', 'D35:D38');
m_node10=xlsread('AFCS_Database.xlsx', 'D39:D42');
m_node11=xlsread('AFCS_Database.xlsx', 'D43:D45');


%% The Volume of each avionics Node for different LRUs
v_node1=xlsread('AFCS_Database.xlsx', 'G2:G6');
v_node2=xlsread('AFCS_Database.xlsx', 'G7:G10');
v_node3=xlsread('AFCS_Database.xlsx', 'G11:G15');
v_node4=xlsread('AFCS_Database.xlsx', 'G16:G18');
v_node5=xlsread('AFCS_Database.xlsx', 'G19:G21');
v_node6=xlsread('AFCS_Database.xlsx', 'G22:G26');
v_node7=xlsread('AFCS_Database.xlsx', 'G27:G30');
v_node8=xlsread('AFCS_Database.xlsx', 'G31:G34');
v_node9=xlsread('AFCS_Database.xlsx', 'G35:G38');
v_node10=xlsread('AFCS_Database.xlsx', 'G39:G42');
v_node11=xlsread('AFCS_Database.xlsx', 'G43:G45');
```

The main file:

```
clc;

clear;
close all;

%% Calling Input file
Inputs();

%% Creating Matrices

f=[m_node1;m_node2;m_node3;m_node4;m_node5;m_node6;m_node7;m_node8;m_node9;m
_node10;m_node11];
```

```
A=[m_node1' m_node2' m_node3' m_node4' m_node5' m_node6' m_node7' m_node8'
m_node9' m_node10' m_node11'
   m_node6' m_node7' m_node8' m_node9' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   m_node1' m_node2' m_node3' m_node4' m_node5' m_node11' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0
   m_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   v_node6' v_node7' v_node8' v_node9' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   v_node1' v_node2' v_node3' v_node4' v_node5' v_node11' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
   v_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];


 b= [100 50 40 30 3100 2500 1000];
Aeq=[1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1];

beq=[1;1;1;1;1;1;1;1;1;1;1];

lb=zeros(44,1);
ub=ones(44,1);
intcon=1:44;

%% Solving
[x z exitflag output]= intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)


T=table(x);
fileName='AFCS_database.xlsx';
writetable(T,fileName,'sheet',1,'range','J1')
```

A.2 The Architecture with Minimum Power Consumption

The Input file:

```
%data1=xlsread('AFCS_Database.xlsx','D:D');
%% The Mass of each avionics Node for different LRUs
m_node1=xlsread('AFCS_Database.xlsx', 'D2:D6');
m_node2=xlsread('AFCS_Database.xlsx', 'D7:D10');
m_node3=xlsread('AFCS_Database.xlsx', 'D11:D15');
m_node4=xlsread('AFCS_Database.xlsx', 'D16:D18');
m_node5=xlsread('AFCS_Database.xlsx', 'D19:D21');
m_node6=xlsread('AFCS_Database.xlsx', 'D22:D26');
m_node7=xlsread('AFCS_Database.xlsx', 'D27:D30');
m_node8=xlsread('AFCS_Database.xlsx', 'D31:D34');
m_node9=xlsread('AFCS_Database.xlsx', 'D35:D38');
m_node10=xlsread('AFCS_Database.xlsx', 'D39:D42');
m_node11=xlsread('AFCS_Database.xlsx', 'D43:D45');
```

```
%% Power consumption of each avionics Node for different LRUs
p_node1=xlsread('AFCS_Database.xlsx', 'E2:E6');
p_node2=xlsread('AFCS_Database.xlsx', 'E7:E10');
p_node3=xlsread('AFCS_Database.xlsx', 'E11:E15');
p_node4=xlsread('AFCS_Database.xlsx', 'E16:E18');
p_node5=xlsread('AFCS_Database.xlsx', 'E19:E21');
p_node6=xlsread('AFCS_Database.xlsx', 'E22:E26');
p_node7=xlsread('AFCS_Database.xlsx', 'E27:E30');
p_node8=xlsread('AFCS_Database.xlsx', 'E31:E34');
p_node9=xlsread('AFCS_Database.xlsx', 'E35:E38');
p_node10=xlsread('AFCS_Database.xlsx', 'E39:E42');
p_node11=xlsread('AFCS_Database.xlsx', 'E43:E45');

%% The volume of each avionics Node for different LRUs
v_node1=xlsread('AFCS_Database.xlsx', 'G2:G6');
v_node2=xlsread('AFCS_Database.xlsx', 'G7:G10');
v_node3=xlsread('AFCS_Database.xlsx', 'G11:G15');
v_node4=xlsread('AFCS_Database.xlsx', 'G16:G18');
v_node5=xlsread('AFCS_Database.xlsx', 'G19:G21');
v_node6=xlsread('AFCS_Database.xlsx', 'G22:G26');
v_node7=xlsread('AFCS_Database.xlsx', 'G27:G30');
v_node8=xlsread('AFCS_Database.xlsx', 'G31:G34');
v_node9=xlsread('AFCS_Database.xlsx', 'G35:G38');
v_node10=xlsread('AFCS_Database.xlsx', 'G39:G42');
v_node11=xlsread('AFCS_Database.xlsx', 'G43:G45');
```

The main file:

```
clc;

clear;
close all;

%Calling Input File
Input();


%% Creating Matrices

f=[p_node1;p_node2;p_node3;p_node4;p_node5;p_node6;p_node7;p_node8;p_node9;p_node
10;p_node11];

A=[m_node1' m_node2' m_node3' m_node4' m_node5' m_node6' m_node7' m_node8'
m_node9' m_node10' m_node11'
  m_node6' m_node7' m_node8' m_node9' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  m_node1' m_node2' m_node3' m_node4' m_node5' m_node11' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0
  m_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  v_node6' v_node7' v_node8' v_node9' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  v_node1' v_node2' v_node3' v_node4' v_node5' v_node11' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
  v_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];


 b=[100 50 40 30 3100 2500 1000];
Aeq=[1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
    0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1];

beq=[1;1;1;1;1;1;1;1;1;1;1];

lb=zeros(44,1);
ub=ones(44,1);
intcon=1:44;

%% Solving
[x z exitflag output]= intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)


%load AFCS_database.xlsx;
T=table(x);
fileName='AFCS_database.xlsx';
writetable(T,fileName,'sheet',1,'range','J1')
```

A.3 The Architecture with Minimum Volume for SSA

The input File:

```
%data1=xlsread('AFCS_Database.xlsx','D:D');
%% The Mass of each avionics Node for different LRUs
m_node1=xlsread('AFCS_Database.xlsx', 'D2:D6');
m_node2=xlsread('AFCS_Database.xlsx', 'D7:D10');
m_node3=xlsread('AFCS_Database.xlsx', 'D11:D15');
m_node4=xlsread('AFCS_Database.xlsx', 'D16:D18');
m_node5=xlsread('AFCS_Database.xlsx', 'D19:D21');
m_node6=xlsread('AFCS_Database.xlsx', 'D22:D26');
m_node7=xlsread('AFCS_Database.xlsx', 'D27:D30');
m_node8=xlsread('AFCS_Database.xlsx', 'D31:D34');
m_node9=xlsread('AFCS_Database.xlsx', 'D35:D38');
m_node10=xlsread('AFCS_Database.xlsx', 'D39:D42');
m_node11=xlsread('AFCS_Database.xlsx', 'D43:D45');

%% The volume of each avionics Node for different LRUs
v_node1=xlsread('AFCS_Database.xlsx', 'G2:G6');
v_node2=xlsread('AFCS_Database.xlsx', 'G7:G10');
v_node3=xlsread('AFCS_Database.xlsx', 'G11:G15');
v_node4=xlsread('AFCS_Database.xlsx', 'G16:G18');
v_node5=xlsread('AFCS_Database.xlsx', 'G19:G21');
v_node6=xlsread('AFCS_Database.xlsx', 'G22:G26');
v_node7=xlsread('AFCS_Database.xlsx', 'G27:G30');
v_node8=xlsread('AFCS_Database.xlsx', 'G31:G34');
v_node9=xlsread('AFCS_Database.xlsx', 'G35:G38');
v_node10=xlsread('AFCS_Database.xlsx', 'G39:G42');
v_node11=xlsread('AFCS_Database.xlsx', 'G43:G45');
```

The Main File

```
clc;

clear;
close all;

%Calling Input File
Input();


%% Creating Matrices
f=[v_node1;v_node2;v_node3;v_node4;v_node5;v_node6;v_node7;v_node8;v_node9;v_node1
0;v_node11];

A=[m_node1' m_node2' m_node3' m_node4' m_node5' m_node6' m_node7' m_node8'
m_node9' m_node10' m_node11'
   m_node6' m_node7' m_node8' m_node9' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   m_node1' m_node2' m_node3' m_node4' m_node5' m_node11' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0
   m_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   v_node6' v_node7' v_node8' v_node9' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   v_node1' v_node2' v_node3' v_node4' v_node5' v_node11' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
   v_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];


 b=[100 50 40 30 3100 2500 1000];
Aeq=[1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1];

beq=[1;1;1;1;1;1;1;1;1;1;1];

lb=zeros(44,1);
ub=ones(44,1);
intcon=1:44;

%% Solving
[x z exitflag output]= intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)

%load AFCS_database.xlsx;
T=table(x);
fileName='AFCS_database.xlsx';
writetable(T,fileName,'sheet',1,'range','J1')
```

A.4 The Architecture with Maximum Reliability for SSA

The Input file:

```
%data1=xlsread('AFCS_Database.xlsx','D:D');
```

```
%% The Mass of each avionics Node for different LRUs
m_node1=xlsread('AFCS_Database.xlsx', 'D2:D6');
m_node2=xlsread('AFCS_Database.xlsx', 'D7:D10');
m_node3=xlsread('AFCS_Database.xlsx', 'D11:D15');
m_node4=xlsread('AFCS_Database.xlsx', 'D16:D18');
m_node5=xlsread('AFCS_Database.xlsx', 'D19:D21');
m_node6=xlsread('AFCS_Database.xlsx', 'D22:D26');
m_node7=xlsread('AFCS_Database.xlsx', 'D27:D30');
m_node8=xlsread('AFCS_Database.xlsx', 'D31:D34');
m_node9=xlsread('AFCS_Database.xlsx', 'D35:D38');
m_node10=xlsread('AFCS_Database.xlsx', 'D39:D42');
m_node11=xlsread('AFCS_Database.xlsx', 'D43:D45');

%% MTBF of each avionics Node for different LRUs
MTBF_node1=xlsread('AFCS_Database.xlsx', 'F2:F6');
MTBF_node2=xlsread('AFCS_Database.xlsx', 'F7:F10');
MTBF_node3=xlsread('AFCS_Database.xlsx', 'F11:F15');
MTBF_node4=xlsread('AFCS_Database.xlsx', 'F16:F18');
MTBF_node5=xlsread('AFCS_Database.xlsx', 'F19:F21');
MTBF_node6=xlsread('AFCS_Database.xlsx', 'F22:F26');
MTBF_node7=xlsread('AFCS_Database.xlsx', 'F27:F30');
MTBF_node8=xlsread('AFCS_Database.xlsx', 'F31:F34');
MTBF_node9=xlsread('AFCS_Database.xlsx', 'F35:F38');
MTBF_node10=xlsread('AFCS_Database.xlsx', 'F39:F42');
MTBF_node11=xlsread('AFCS_Database.xlsx', 'F43:F45');

%% The volume of each avionics Node for different LRUs
v_node1=xlsread('AFCS_Database.xlsx', 'G2:G6');
v_node2=xlsread('AFCS_Database.xlsx', 'G7:G10');
v_node3=xlsread('AFCS_Database.xlsx', 'G11:G15');
v_node4=xlsread('AFCS_Database.xlsx', 'G16:G18');
v_node5=xlsread('AFCS_Database.xlsx', 'G19:G21');
v_node6=xlsread('AFCS_Database.xlsx', 'G22:G26');
v_node7=xlsread('AFCS_Database.xlsx', 'G27:G30');
v_node8=xlsread('AFCS_Database.xlsx', 'G31:G34');
v_node9=xlsread('AFCS_Database.xlsx', 'G35:G38');
v_node10=xlsread('AFCS_Database.xlsx', 'G39:G42');
v_node11=xlsread('AFCS_Database.xlsx', 'G43:G45');
```

The main file:

```
clc;
clear;
close all;

%Calling Input File
Input();


%% Creating Matrices

f=[MTBF_node1;MTBF_node2;MTBF_node3;MTBF_node4;MTBF_node5;MTBF_node6;MTBF_
node7;MTBF_node8;MTBF_node9;MTBF_node10;MTBF_node11];
f=-f;


A=[m_node1' m_node2' m_node3' m_node4' m_node5' m_node6' m_node7' m_node8'
m_node9' m_node10' m_node11'
```

```
   m_node6' m_node7' m_node8' m_node9' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   m_node1' m_node2' m_node3' m_node4' m_node5' m_node11' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0
   m_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   v_node6' v_node7' v_node8' v_node9' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   v_node1' v_node2' v_node3' v_node4' v_node5' v_node11' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
   v_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];


 b=[100 50 40 30 3100 2500 1000];
Aeq=[1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1];

beq=[1;1;1;1;1;1;1;1;1;1;1];

lb=zeros(44,1);
ub=ones(44,1);
intcon=1:44;

%% Solving
[x z exitflag output]= intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)

%load AFCS_database.xlsx;
T=table(x);
fileName='AFCS_database.xlsx';
writetable(T,fileName,'sheet',1,'range','J1')
```

A.5 The Architecture with Maximum Operational Capability SSA

The Input File

```
%data1=xlsread('AFCS_Database.xlsx','D:D');
%% The Mass of each avionics Node for different LRUs
m_node1=xlsread('AFCS_Database.xlsx', 'D2:D6');
m_node2=xlsread('AFCS_Database.xlsx', 'D7:D10');
m_node3=xlsread('AFCS_Database.xlsx', 'D11:D15');
m_node4=xlsread('AFCS_Database.xlsx', 'D16:D18');
m_node5=xlsread('AFCS_Database.xlsx', 'D19:D21');
m_node6=xlsread('AFCS_Database.xlsx', 'D22:D26');
m_node7=xlsread('AFCS_Database.xlsx', 'D27:D30');
m_node8=xlsread('AFCS_Database.xlsx', 'D31:D34');
m_node9=xlsread('AFCS_Database.xlsx', 'D35:D38');
m_node10=xlsread('AFCS_Database.xlsx', 'D39:D42');
m_node11=xlsread('AFCS_Database.xlsx', 'D43:D45');

%% Operational Capability of each avionics Node for different LRUs
OC_node1=xlsread('AFCS_Database.xlsx', 'H2:H6');
```

```
OC_node2=xlsread('AFCS_Database.xlsx', 'H7:H10');
OC_node3=xlsread('AFCS_Database.xlsx', 'H11:H15');
OC_node4=xlsread('AFCS_Database.xlsx', 'H16:H18');
OC_node5=xlsread('AFCS_Database.xlsx', 'H19:H21');
OC_node6=xlsread('AFCS_Database.xlsx', 'H22:H26');
OC_node7=xlsread('AFCS_Database.xlsx', 'H27:H30');
OC_node8=xlsread('AFCS_Database.xlsx', 'H31:H34');
OC_node9=xlsread('AFCS_Database.xlsx', 'H35:H38');
OC_node10=xlsread('AFCS_Database.xlsx', 'H39:H42');
OC_node11=xlsread('AFCS_Database.xlsx', 'H43:H45');

%% The volume of each avionics Node for different LRUs
v_node1=xlsread('AFCS_Database.xlsx', 'G2:G6');
v_node2=xlsread('AFCS_Database.xlsx', 'G7:G10');
v_node3=xlsread('AFCS_Database.xlsx', 'G11:G15');
v_node4=xlsread('AFCS_Database.xlsx', 'G16:G18');
v_node5=xlsread('AFCS_Database.xlsx', 'G19:G21');
v_node6=xlsread('AFCS_Database.xlsx', 'G22:G26');
v_node7=xlsread('AFCS_Database.xlsx', 'G27:G30');
v_node8=xlsread('AFCS_Database.xlsx', 'G31:G34');
v_node9=xlsread('AFCS_Database.xlsx', 'G35:G38');
v_node10=xlsread('AFCS_Database.xlsx', 'G39:G42');
v_node11=xlsread('AFCS_Database.xlsx', 'G43:G45');
```

The Main File

```
clc;

clear;
close all;

%Calling Input File
Input();


%% Creating Matrices

f=[OC_node1;OC_node2;OC_node3;OC_node4;OC_node5;OC_node6;OC_node7;OC_node8;
OC_node9;OC_node10;OC_node11];
f=-f;


A=[m_node1' m_node2' m_node3' m_node4' m_node5' m_node6' m_node7' m_node8'
m_node9' m_node10' m_node11'
   m_node6' m_node7' m_node8' m_node9' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   m_node1' m_node2' m_node3' m_node4' m_node5' m_node11' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0
   m_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   v_node6' v_node7' v_node8' v_node9' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   v_node1' v_node2' v_node3' v_node4' v_node5' v_node11' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
   v_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];


 b=[100 50 40 30 3100 2500 1000];
Aeq=[1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1];

beq=[1;1;1;1;1;1;1;1;1;1;1];

lb=zeros(44,1);
ub=ones(44,1);
intcon=1:44;

%% Solving
[x z exitflag output]= intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)

%load AFCS_database.xlsx;
T=table(x);
fileName='AFCS_database.xlsx';
writetable(T,fileName,'sheet',1,'range','J1')
```

A.6 The Architecture with Minimum weight for LSA

The Input file

```
%Input Files
%% The Mass of each avionics Node for different LRUs
m_node1=xlsread('Avionics_Database.xlsx', 'D2:D6');
m_node2=xlsread('Avionics_Database.xlsx', 'D7:D10');
m_node3=xlsread('Avionics_Database.xlsx', 'D11:D15');
m_node4=xlsread('Avionics_Database.xlsx', 'D16:D18');
m_node5=xlsread('Avionics_Database.xlsx', 'D19:D21');
m_node6=xlsread('Avionics_Database.xlsx', 'D22:D26');
m_node7=xlsread('Avionics_Database.xlsx', 'D27:D30');
m_node8=xlsread('Avionics_Database.xlsx', 'D31:D34');
m_node9=xlsread('Avionics_Database.xlsx', 'D35:D38');
m_node10=xlsread('Avionics_Database.xlsx', 'D39:D42');
m_node11=xlsread('Avionics_Database.xlsx', 'D43:D45');
m_node12=xlsread('Avionics_Database.xlsx', 'D46:D49');
m_node13=xlsread('Avionics_Database.xlsx', 'D50:D52');
m_node14=xlsread('Avionics_Database.xlsx', 'D53:D55');
m_node15=xlsread('Avionics_Database.xlsx', 'D56:D59');
m_node16=xlsread('Avionics_Database.xlsx', 'D60:D62');
m_node17=xlsread('Avionics_Database.xlsx', 'D63:D65');

%% The Volume of each avionics Node for different LRUs
v_node1=xlsread('Avionics_Database.xlsx', 'G2:G6');
v_node2=xlsread('Avionics_Database.xlsx', 'G7:G10');
v_node3=xlsread('Avionics_Database.xlsx', 'G11:G15');
v_node4=xlsread('Avionics_Database.xlsx', 'G16:G18');
v_node5=xlsread('Avionics_Database.xlsx', 'G19:G21');
v_node6=xlsread('Avionics_Database.xlsx', 'G22:G26');
v_node7=xlsread('Avionics_Database.xlsx', 'G27:G30');
v_node8=xlsread('Avionics_Database.xlsx', 'G31:G34');
```

```
v_node9=xlsread('Avionics_Database.xlsx', 'G35:G38');
v_node10=xlsread('Avionics_Database.xlsx', 'G39:G42');
v_node11=xlsread('Avionics_Database.xlsx', 'G43:G45');
v_node12=xlsread('Avionics_Database.xlsx', 'G46:G49');
v_node13=xlsread('Avionics_Database.xlsx', 'G50:G52');
v_node14=xlsread('Avionics_Database.xlsx', 'G53:G55');
v_node15=xlsread('Avionics_Database.xlsx', 'G56:G59');
v_node16=xlsread('Avionics_Database.xlsx', 'G60:G62');
v_node17=xlsread('Avionics_Database.xlsx', 'G63:G65');
```

The main file

```
clc;

clear;
close all;

%% Calling Input file
Inputs();

%% Creating Matrices

f=[m_node1;m_node2;m_node3;m_node4;m_node5;m_node6;m_node7;m_node8;m_node9;m
_node10;m_node11;m_node12;m_node13;m_node14;m_node15;m_node16;m_node17];


A=[m_node1' m_node2' m_node3' m_node4' m_node5' m_node6' m_node7' m_node8'
m_node9' m_node10' m_node11' m_node12' m_node13' m_node14' m_node15' m_node16'
m_node17'
   m_node6' m_node7' m_node8' m_node9' m_node14' m_node16' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   m_node1' m_node2' m_node3' m_node4' m_node5' m_node11' m_node12' m_node13'
m_node15' m_node17' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   m_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   v_node6' v_node7' v_node8' v_node9' v_node14' v_node16' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   v_node1' v_node2' v_node3' v_node4' v_node5' v_node11' v_node12' v_node13' v_node15'
v_node17' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   v_node10' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];


 b=[100 50 40 30 3100 2500 1000];
Aeq=[1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1];

beq=[1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1];

lb=zeros(64,1);
ub=ones(64,1);
intcon=1:64;

%% Solving
[x z exitflag output]= intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)


%load AFCS_database.xlsx;
T=table(x);
fileName='Avionics_database.xlsx';
writetable(T,fileName,'sheet',1,'range','J1')
```

The other single-objective optimisation codes are the same as above, and the objective function only changes in each case.

A.7 The Architecture with Minimum weight for LSA Using PSO

PSO main file

```
clc;
clear;
close all;

%% Problem Definition

global NFE;
NFE=0;

model=CreateModel();
```

```
nVar=model.N;       % Number of Decision Variables

VarSize=[1 nVar];   % Size of Decision Variables Matrix


VarMin=model.lbound;
VarMax=model.ubound;


%% PSO Parameters

MaxIt=300;      % Maximum Number of Iterations

nPop=50;        % Population Size (Swarm Size)

w=1;            % Inertia Weight
wdamp=0.99;     % Inertia Weight Damping Ratio
c1=0.2;         % Personal Learning Coefficient
c2=0.4;         % Global Learning Coefficient

% Velocity Limits
VelMax=0.1*(VarMax-VarMin);
VelMin=-VelMax;

%% Initialization

empty_particle.Position=[];
empty_particle.Cost=[];
empty_particle.Sol=[];
empty_particle.Velocity=[];
empty_particle.Best.Position=[];
empty_particle.Best.Cost=[];
empty_particle.Best.Sol=[];

particle=repmat(empty_particle,nPop,1);

GlobalBest.Cost.weight=inf;
GlobalBest.Cost.MTBF=0;
GlobalBest.Cost.overall=-inf;
BestMTBF=zeros(MaxIt,1);
Bestweight=zeros(MaxIt,1);

for i=1:nPop

    % Initialize Position
    particle(i).Position=Positioning(VarMin,VarMax,nVar,model);%

    % Initialize Velocity
    particle(i).Velocity=zeros(VarSize);

    % Evaluation
    particle(i).Cost=CostFunction(particle(i).Position,model);

    % Update Personal Best
    particle(i).Best.Position=particle(i).Position;
    particle(i).Best.Cost=particle(i).Cost;
```

```
  % Update Global Best
  if judge(particle(i).Best.Cost,GlobalBest.Cost)


      GlobalBest=particle(i).Best;

  end
end
bestcost.weight=10000;
bestcost.MTBF=0;

BestCost=repmat(bestcost,MaxIt,1);


%% PSO Main Loop

for it=1:MaxIt

  for i=1:nPop
    flg=false;
    while flg==false
     % Update Velocity
     for vl=1:nVar
     particle(i).Velocity(vl) = w*particle(i).Velocity(vl) ...
       +c1*rand.*(particle(i).Best.Position(vl)-particle(i).Position(vl)) ...
       +c2*rand.*(GlobalBest.Position(vl)-particle(i).Position(vl));


     % Update Position
     particle(i).Position(vl) =particle(i).Position(vl) + particle(i).Velocity(vl);

     % Velocity Mirror Effect
     IsOutside=(particle(i).Position(vl)<VarMin(vl) | particle(i).Position(vl)>VarMax(vl));
     particle(i).Velocity(IsOutside)=-particle(i).Velocity(IsOutside);

     % Apply Position Limits
     particle(i).Position(vl) =round( max(particle(i).Position(vl),VarMin(vl)));
     particle(i).Position(vl) = round(min(particle(i).Position(vl),VarMax(vl)));
     end
     if checksol(particle(i).Position,model)
       flg=true;
     end
    end
     particle(i).Cost  = CostFunction(particle(i).Position,model);



          if judge(particle(i).Cost,particle(i).Best.Cost)


          particle(i).Best.Position=particle(i).Position;
          particle(i).Best.Cost=particle(i).Cost;
          particle(i).Best.Sol=particle(i).Sol;
       end
% % % % %       end

     % Update Global Best
      if judge(particle(i).Best.Cost,GlobalBest.Cost)
```

```
            GlobalBest=particle(i).Best;

        end

    end

BestCost(it).weight=GlobalBest.Cost.weight;
BestCost(it).MTBF=GlobalBest.Cost.MTBF;
BestMTBF(it)=BestCost(it).MTBF;
Bestweight(it)=BestCost(it).weight;
Bestapproach=GlobalBest.Position;
nfe(it)=NFE;

disp(['Iteration ' num2str(it) ': NFE = ' num2str(nfe(it)) ', Best weight = '
num2str(BestCost(it).weight) ', Best MTBF = ' num2str(BestCost(it).MTBF)]);

w=w*wdamp;
end
disp(['The best Position is:  ' num2str(Bestapproach) ]);
% celldisp(model.Bxes(Bestapproach));
mm=(model.Bxes(Bestapproach));
disp(['The best Position name:  ' (mm') ]);
figure(1);

plotpareto(BestMTBF,Bestweight,MaxIt);


nfe=(1:MaxIt);

%% Results
hold
figure;
plot(nfe,BestMTBF,'LineWidth',2);
xlabel('Iteration');
ylabel('Best MTBF Cost');
hold
figure;
plot(nfe,Bestweight,'LineWidth',2);
xlabel('Iteration');
ylabel('Best weight  Cost');
```

Appendix B Avionics Operational Capability Assessment

In this section, the operational capability of the seventeen avionics LRUs is discussed. For each avionics LRU and/or avionics function at least three various LRUs are recorded that are evaluated against a set of criteria using SAW method in excel datasheet.

B.1 ADC Operational Capability Assessment

Air Data Computer (ADC)	Criteria	PA Accuracy	IAS Accuracy	Temp Accuracy	Mach	RVSM compliant

Decision Matrix

Weighting	0.3	0.2	0.1	0.2	0.4
Criteria	C1	C2	C3	C4	C5
ESCADU	5	5	7	9	9
ADUM	5	5	5	5	9
AADC	9	7	7	7	9
DADC	7	9	9	7	9
AMADC	9	7	7	7	9

Normalized Matrix

	C1	C2	C3	C4	C5
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

The selected ADC guarantees the following performance:

PA accuracy: ± 5 ft

IAS accuracy: ± 0.3 Kts

Temperature Accuracy: ± 0.5 Deg/C

RVSM compliant

B.2 GPS Receiver Operational Capability Assessment

GPS Receiver	Criteria	Altitude Accuracy	Velocity Accuracy	Position Accuracy	Approach Capability(SBAS/LPV or GBAS/GLS)		
Decision Matrix							
	Weighting	0.2	0.2	0.3	0.4		
	Criteria	C1	C2	C3	C4		
	CMA5024	7	5	5	7		
	CMA6024	9	7	7	9		
	CMA3024	7	7	7	5		
	GPSWAAS	7	5	5	5		
	GPS400S	7	5	5	5		
Normalized Matrix							
	[illegible]	C1	C2	C3	C4		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	1	1	1	1		
	[illegible]	[illegible]	1	1	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
[illegible]							
	[illegible]	C1	C2	C3	C4	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

The selected GPS receiver guarantees the following performance:

Altitude accuracy: typically < 3 meters, with SBAS

Velocity accuracy: < 0.5 knots

Position accuracy: typically < 2 meters with SBAS

Approach Capability: SBAS LPV, LP, LNAV/VNAV

B.3 VORILS Operational Capability Assessment

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
[illegible]						
	[illegible]	[illegible]	[illegible]	[illegible]		
	Criteria	C1	C2	C3		
	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]		
[illegible]						
	Criteria	C1	C2	C3		
	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	1	[illegible]		
	[illegible]	1	1	1		
[illegible]						
	Criteria	C1	C2	C3	Results	Rank
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

The selected VORILS guarantees the following performance:

Deviation Accuracy: < $\pm 3^o$

Number of channels: 160 channels LOC/ 40 channels GS

Capability: VOR/ILS/MLS.

B.4 RA Operational Capability Assessment

[illegible]	Criteria	Height Accuracy	Altitude Range	Attitude (Pitch/Roll) Range

Decision Matrix

Weighting	0.4	0.3	0.3
Criteria	C1	C2	C3
[illegible]	7	9	9
[illegible]	9	7	7
[illegible]	5	5	5

Normalized Matrix

Criteria	C1	C2	C3
[illegible]	0.777777778	1	1
[illegible]	1	0.777777778	0.777777778
[illegible]	0.555555556	0.555555556	0.555555556

Normalized Weighted Matrix

Criteria	C1	C2	C3	Results	Rank
[illegible]	0.311111111	0.3	0.3	0.911111	1
[illegible]	0.4	0.233333333	[illegible]	[illegible]	2
[illegible]	0.222222222	0.166666667	[illegible]	0.5	3

The selected RA guarantees the following performance:

Height Accuracy: ± 2 ft

Altitude range: 0 to 5000 ft (another version to 10000ft)

Attitude range: $\pm 60^o$

B.5 FMS Operational Capability Assessment

Flight Management System	Criteria	LNAV/VNAV Capability	RNP/RNAV Capability	RTA Capability	DataLink/CPDLC		
Decision Matrix							
	Weighting	0.2	0.3	0.4	0.3		
	Criteria	C1	C2	C3	C4		
	CMA9000	7	7	5	7		
	GE6883	5	5	5	5		
	FDMS	7	5	9	7		
	UNS1ESPW	9	7	7	9		
	UNS1FW	9	7	7	9		
Normalized Matrix							
	Criteria	C1	C2	C3	C4		
	CMA9000	0.777777778	1	0.555555556	0.777777778		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
[illegible]							
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

The selected FMS guarantees the following performance:

It is capable of SBAS LPV, ILS CAT II approach capability, vertical navigation, B-RNAV, RNP down to 0.3. CPDLC and ADS-B software also loaded in this FMS.

B.6 PFD Operational Capability Assessment

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
[illegible]							
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
[illegible]							
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
[illegible]							
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

The selected PFD guarantees the following performance:

Display Area: 10.4” (220.2 mm) diagonal

Resolution: 1280 pixels (W) x 768 pixels (H)

Viewing Angle:$\pm 60^o$

Brightness: >500 CD/m2

B.7 MFD Operational Capability Assessment

Multi Function Display (MFD)

Criteria	Display Area	Resolution	Viewing angle	Brightness

Decision Matrix

Weighting	0.2	0.2	0.2	0.2
Criteria	C1	C2	C3	C4
[illegible]	9	9	5	7
[illegible]	7	9	7	7
[illegible]	7	9	6	7
[illegible]	9	7	7	9

Normalized Matrix

Criteria	C1	C2	C3	C4
[illegible]	1	1	0.555555556	0.777777778
[illegible]	0.777777778	1	0.777777778	0.777777778
[illegible]	0.777777778	1	1	0.777777778
[illegible]	1	0.777777778	0.777777778	1

Normalized Weighted Matrix

Criteria	C1	C2	C3	C4	Results	Rank
[illegible]	0.2	0.2	0.111111111	0.155555556	0.666667	3
[illegible]	0.155555556	0.2	0.155555556	0.155555556	0.666667	4
[illegible]	0.155555556	0.2	0.2	0.155555556	0.711111	1
[illegible]	0.2	0.155555556	0.155555556	0.2	0.711111	2

The selected MFD guarantees the following performance:

Display Area: 10.6” (269.2 mm) diagonal

Resolution: 1240 pixels (W) x 768 pixels (H)

Viewing Angle:$\pm 80^o$

Brightness: >500 CD/m2

B.8 HUD Operational Capability Assessment

Head Up Display (HUD)

Criteria	Resolution	Field of View	Approach Capability	Function Integration (EVS, SVS)

Decision Matrix

Weighting	0.2	0.2	0.4	0.5
Criteria	C1	C2	C3	C4
Aviguide	7	7	9	9
HGS	7	5	9	7
GHD	7	7	5	5
HUD2020	7	7	7	9

Normalized Matrix

Criteria	C1	C2	C3	C4
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

The selected HUD guarantees the following performance:

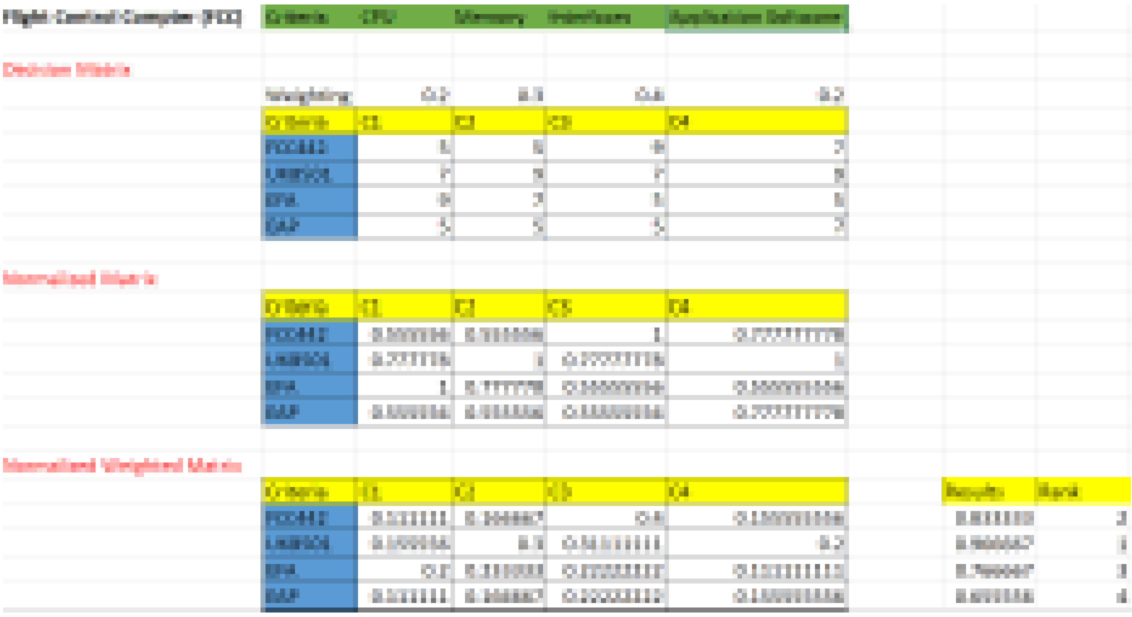
Resolution: 1400×1500

Field of view: 30°x22.5°

Integrated Functions: EVS, SVS

Approach Capability: capable of descending below official weather minima

B.9 FCC Operational Capability Assessment



Flight Control Computer (FCC)	Criteria	CPU	Memory	Interfaces	Application Software

Decision Matrix

Weighting	0.2	[illegible]	[illegible]	0.2
Criteria	C1	C2	C3	C4
FCC442	5	6	[illegible]	7
UKB501	7	9	7	9
[illegible]	9	7	5	5
EAP	5	5	5	7

Normalised Matrix

Criteria	C1	C2	C3	C4
FCC442	[illegible]	[illegible]	1	[illegible]
UKB501	[illegible]	1	[illegible]	1
[illegible]	1	[illegible]	[illegible]	[illegible]
EAP	[illegible]	[illegible]	[illegible]	[illegible]

Normalised Weighted Matrix

Criteria	C1	C2	C3	C4	Results	Rank
FCC442	[illegible]	[illegible]	0.4	[illegible]	[illegible]	2
UKB501	[illegible]	[illegible]	[illegible]	0.2	[illegible]	1
[illegible]	0.2	[illegible]	[illegible]	[illegible]	[illegible]	3
EAP	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4

The selected FCC guarantees the following performance:

CPU: PowerPC computing Processor

Memory: 64 MB

Interfaces:

- 8x OPEN/GND Discrete Input, 8x OPEN/GND Discrete Output
- 4 x MIL-STD-1553B Interface
- 13 x RS-422/RS-485 Interface (2400 - 230400 baud rate)
- 3 x CAN 2.0 A&B CANBus Interface (100Kbit/sec - 1Mbit/sec)
- 8 Rx and 4 Tx ARINC 429 Interface (12.5 Kbps - 100 Kbps)
- 4x software configurable Synchronous / Asynchronous RS-422 Interface (2400 - 230400 Baud Rate)
- 1 x 10/100 Ethernet Interface
- 1x 10/100/1000 Ethernet Interface

Software applications: UKB-501 consists of RTCA DO-178B Level B certifiable BSP and driver software for Green Hills Integrity 178B real-time operating system and RTCA-DO-254 Level B certifiable hardware

B.10 IAC Operational Capability Assessment

Integrate Avionics Cabinet (IAC)	Criteria	CPU	Memory	Interfaces	Application Software		
Decision Matrix							
	Weighting	0.2	0.3	0.4	0.2		
	Criteria	C1	C2	C3	C4		
	CMA5000	7	7	7	7		
	FV4000	9	9	9	9		
	PU3000	5	5	5	5		
Normalized Matrix							
	Criteria	C1	C2	C3	C4		
	CMA5000	0.777778	0.777778	0.7777777778	0.7777777778		
	FV4000	1	1	1	1		
	PU3000	0.555556	0.555556	0.5555555556	0.5555555556		
Normalized Weighted Matrix							
	Criteria	C1	C2	C3	C4	Results	Rank
	CMA5000	0.155556	0.233333	0.3111111111	0.1555555556	0.855556	2
	FV4000	0.2	0.3	0.4	0.2	1.1	1
	PU3000	0.111111	0.166667	0.2222222222	0.1111111111	0.611111	3

The selected IAC guarantees the following performance:

CPU: MPC7447 Power PC, 1.1GHz

Memory: 512M RAM, 256M Flash, 8M Boot Flash

Interface:

ARINC-429 In 42

ARIN-429 Out 22
Discrete In Open/Ground 77
Discrete In Open/28V 17
Discrete In 28V/Ground 1
Discrete Out Open/Ground 61
Discrete Out Open/28V 3
Discrete In RS-422 time mark 2
Analog In 2
Analog Out 11
Ethernet external links 4
Ethernet Internal links 4
MIL-STD-1553B 3
Composite Video In 4
Composite Video Out 3
RGB Video In 1
RGB Video Out 4
RS-232 5
RS-422 in 11
RS-422-Out 8
RS-485 1
HUD Interface 1

CAN BUS (I2C protocol)

Application software: Green Hills Integrity and RTCA/DO-178B

B.11 WXR Operational Capability Assessment

Weather Radar (WXR)	Criteria	Max Detection range	Azimuth Coverage	Elevation Coverage	Capabilities		capabilities
Decision Matrix							
	Weighting	0.2	0.2	0.3	0.4		
	Criteria	C1	C2	C3	C4		
	WXR2100	7	7	7	5		
	RTA4100	5	5	5	5		
	RDR7000	9	9	9	9		
	WXR840	7	7	7	7		
Normalized Matrix							
	Criteria	C1	C2	C3	C4		
	WXR2100	[illegible]	[illegible]	[illegible]	[illegible]		
	RTA4100	[illegible]	[illegible]	[illegible]	[illegible]		
	RDR7000	1	1	1	1		
	WXR840	[illegible]	[illegible]	[illegible]	[illegible]		
Normalized Weighted Matrix							
	Criteria	C1	C2	C3	C4	Results	Rank
	WXR2100	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	RTA4100	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	RDR7000	0.2	0.2	0.3	0.4	1.1	1
	WXR840	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

The selected WXR guarantees the following performance:

Max Detection range: 320 Nm weather and Ground map, 60 nm Turbulence, 5 nm windshear.

Azimuth coverage: +/- 60 degs - Weather and Ground Map, +/- 40 degs windshear

Elevation coverage: 0 to 60000 ft

Capabilities: Turbulence and windshear detection

B.12 TCAS Operational Capability Assessment

Traffic Alert Collision Aviodance System (TCAS), T/	Criteria	Bearing Accuracy	Range Capability	Operating Altitude	#Functions Integrated		Funcions: T
Decision Matrix							
	Weighting	0.2	0.2	0.2	0.4		
	Criteria	C1	C2	C3	C4		
	T3CAS	9	9	9	9		
	T2CAS	5	5	5	5		
	TSS4100	7	7	7	7		
Normalized Matrix							
	Criteria	C1	C2	C3	C4		
	T3CAS	1	1	1	1		
	T2CAS	[illegible]	[illegible]	[illegible]	[illegible]		
	TSS4100	[illegible]	[illegible]	[illegible]	[illegible]		
Normalized Weighted Matrix							
	Criteria	C1	C2	C3	C4	Results	Rank
	T3CAS	0.2	0.2	0.2	0.4	1	1
	T2CAS	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	TSS4100	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

The selected TCAS guarantees the following performance:

Bearing Accuracy: 2 degree

Range capability: 80 nm active, 100+ nm pssive

Operating Altitude: Sea Level to 55000 ft

Functions integrated: TCAS, TAWS, ADS-B, Mode S transponder

B.13 EFB Operational Capability Assessment

Electronic Flight Bag (EFB)	Criteria	Resolution	Viewing Angle	CPU & Memory	Functions		
Decision Matrix							
	Weighting	0.2	0.2	0.3	0.4		
	Criteria	C1	C2	C3	C4		
	TacView	5	7	7	7		
	CMA1612	5	7	5	7		
	CMA1310	9	9	9	9		
Normalized Matrix							
	Criteria	C1	C2	C3	C4		
	TacView	[illegible]	[illegible]	[illegible]	[illegible]		
	CMA1612	[illegible]	[illegible]	[illegible]	[illegible]		
	CMA1310	1	1	1	1		
Normalized Weighted Matrix							
	Criteria	C1	C2	C3	C4	Results	Rank
	TacView	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
	CMA1612	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	CMA1310	0.1	0.2	0.3	0.4	1.1	1

The selected EFB guarantees the following performance:

Resolution: 1920 x 1200 pixels;

Viewing angle: $\pm 80^{o}$

CPU & memory: Intel Quad Core (2.16GHz), 8GB of DDR3 RAM

Functions: Moving map

B.14 FDR Operational Capability Assessment

Flight Data Recorder (FDR) & CVR

Criteria	Recording Time	Impact shock	Penetration resistance	Deep See Pressure

Decision Matrix

Weighting	0.2	0.3	0.3	0.4
Criteria	C1	C2	C3	C4
CVFDR	9	9	7	7
LDR	9	7	5	5
IDARS	9	7	5	5
HFR5	9	9	9	9

Normalized Matrix

Criteria	C1	C2	C3	C4
CVFDR	1	1	0.7777777778	0.7777777778
LDR	1	0.7777777778	0.5555555556	0.5555555556
IDARS	1	0.7777777778	0.5555555556	0.5555555556
HFR5	1	1	1	1

Normalized Weighted Matrix

Criteria	C1	C2	C3	C4	Results	Rank
CVFDR	0.2	0.3	[illegible]	[illegible]	[illegible]	2
LDR	0.2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
IDARS	0.2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
HFR5	0.2	0.3	0.3	0.4	1.2	1

The selected FDR guarantees the following performance:

Recording time: 120 min CVR, 120 min data link recording, 25 hours FDR

Impact shock: 3,400 G, 6.5 ms,
Penetration resistance: 500 lbs. / 10 ft. / ¼-in. probe
Deep see pressure: 20,000 feet, for 24 hours

B.15 SATCOM Operational Capability Assessment

SATCOM	Criteria	Data Rate	Service Coverage	Functions	Capability (FANS & ACARS)		
Decision Matrix							
	Weighting	0.4	0.2	0.2	0.3		
	Criteria	C1	C2	C3	C4		
	SAT6100	9	7	9	9		
	IRT2110	7	7	5	5		
	CMA2200SB	9	7	7	7		
Normalized Matrix							
	Criteria	C1	C2	C3	C4		
	SAT6100	1	1	1	1		
	IRT2110	0.777777778	1	0.555555556	0.555555556		
	CMA2200SB	1	1	0.777777778	0.777777778		
Normalized Weighted Matrix							
	Criteria	C1	C2	C3	C4	Results	Rank
	SAT6100	0.4	0.2	0.2	0.3	1.1	1
	IRT2110	0.311111111	0.2	0.111111111	0.166666667	0.788889	3
	CMA2200SB	0.4	0.2	0.155555556	0.233333333	0.988889	2

The selected SATCOM guarantees the following performance:

Data rate: 650 Kbps to 1.3 Mbps

Service coverage: Global

Functions: SATCOM, Data link

Capability:
- Graphical weather information
- Engine monitoring and fault reporting
- Electronic Flight Bag
- In-flight entertainment updates
- Fax, email and instant messaging
- Corporate web access
- Video/stills/audio data transfers
- Telephony

B.16 VHF Operational Capability Assessment

VHF

Criteria	Preset channels	Channel change time	VDL Mode2 availablity	Channel Spacing (25, 50, 8.33 Capability)

Decision Matrix

Weighting	0.2	0.2	0.4	0.3
Criteria	C1	C2	C3	C4
SRT700	9	9	9	9
VHF4000	7	7	7	7
ESD1721	5	5	5	5

Normalized Matrix

Criteria	C1	C2	C3	C4
SRT700	1	1	1	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

Criteria	C1	C2	C3	C4	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3

The selected VHF guarantees the following performance:

Preset channels: 99

Channel change time: 1ms

VDL Mode2: available

Channel Spacing Capability: 8.33 KHz

Appendix C IMA Architecture Terminology

IMA avionics architecture include two main activities. First is the hardware platform consisting of IMA modules and their interconnection. The second is the systems layers including the functions which have to be realized by specifying signal flows, signal processing and sources. Combination of these two process is carried out in integration phase. In the following basic terms and concepts from the field of IMA are explained:

Aircraft Function: The capability of the aircraft that may be provided by the hardware and software of the systems on the aircraft. Functions include flight control, autopilot, braking, fuel management, flight instruments, etc. IMA has the potential to broaden the definition of avionics to include any aircraft function.

Application: Software and/or application-specific hardware with a defined set of interfaces that, when integrated with a platform, performs a function.

Component: A self-contained hardware part, software part, database, or combination thereof that is configuration controlled. A component does not provide an aircraft function by itself.

Core software: The operating system and support software that manage resources to provide an environment in which applications can execute. Core software is a necessary component of a platform and is typically comprised of one or more modules.

IMA System: Consists of an IMA platform(s) and a defined set of hosted applications.

Interoperable: The capability of several integrated modules to operate together to accomplish a specific goal or function. This requires defined interface boundaries between the modules and allows the use of other interoperable components. To describe this concept in physical terms, an IMA platform may include interoperable modules and components such as physical devices (processor, memory, electrical power, Input/output devices), and logical elements such as an operating system, and communication software.

Module: A component or a collection of components that may be accepted by themselves or in the context of IMA. A module may also comprise other modules. A module may be software, hardware, or a combination of both, which provides resources to the IMA-hosted applications. Modules may be distributed across the aircraft or may be co-located.

Partitioning: An architectural technique to provide the necessary separation and independence of functions or applications to ensure that only intended coupling occurs. The mechanism for providing the protection in an IMA platform are specified to a required level of integrity.

Platform: Module or group of modules, including core software that manages resources in a manner sufficient to support at least one application. IMA hardware resources and core software are designed and managed in a way to provide computational, communication, and interface capabilities for hosting at least one application. Platforms, by themselves, do not provide any aircraft functionality.

The platform establishes a computing environment, support services, and platform-related capabilities, such as health monitoring and fault management. The IMA platform may be accepted independently of hosted applications.

Resource: Any object (processor, memory, software, data, etc.) or component used by IMA platform or application. A resource may be shared by multiple applications or dedicated to a specific application. A resource may be physical (a hardware device) or logical (a piece of information).

Reusable: The design assurance data of previously accepted modules and applications may be used in a subsequent aircraft system design with reduced need for redesign or additional acceptance.