Artificial Intelligence for Trusted Autonomous Operations of Distributed Satellite Systems

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

Kathiravan Thangavel

M.Sc. Aerospace Engineering, Università degli Studi di Roma "La Sapienza"

B.E. Aeronautical Engineering, Anna University, Chennai

School of Engineering

College of Science, Technology, Engineering and Maths

RMIT University

April 2023

DECLARATION

I certify that except where due acknowledgement has been made, this research is that of the author alone; the content of this research submission is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and procedures and guidelines have been followed.

In addition, I certify that this submission contains no material previously submitted for award of any qualification at any other university or institution, unless approved for a joint-award with another institution, and acknowledge that no part of this work will, in the future, be used in a submission in my name, for any other qualification in any university or other tertiary institution without the prior approval of the University, and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that the copyright of any published works contained within this thesis resides with those works' copyright holder(s). I give permission for the digital version of my research submission to be made available on the web, via the University's digital research repository unless permission has been granted by the University to restrict access for a period of time.

Kathiravan Thangavel

26th April 2023

This thesis is dedicated to my mother for her love, endless support and encouragement.

Acknowledgements

To begin, I would like to convey my profound and heartfelt gratitude to my advisors, Professor Roberto Sabatini, Professor Pier Marzocca, Dr. Haytham Fayek, and Dr. Dario Spiller, for providing me with the opportunity to carry out this research project and providing me with invaluable guidance all the way through the process. Rob's enthusiasm, vision, and energy in the field of research were an enormous driving source of inspiration. I am extremely grateful to him for all the guidance he has provided me during my career. Working and learning under his tutelage was an incredible honour and privilege that I will never forget. The assistance and direction that Pier, Haytham, and Dario offered during my PhD program were of the utmost importance to its successful completion. Because of their assistance, I have gained the understanding, expertise, and technical abilities that, without a shadow of a doubt, will play a significant part in whatever future success I experience in my professional life. I would also like to express my gratitude for their friendship, compassion, and wonderful sense of humour, all of which served as a source of inspiration for me whenever I was going through a challenging situation. I would like to thank SmartSat CRC and The Andy Thomas Space Foundation for their support in carrying out this research work. SmartSat CRC activities are funded by the Australian Government's CRC Program. I would like to thank Adjunct Professor Nicola Sasanelli for his guidance and mentorship throughout the PhD candidature. I would like to thank Dr. Pablo A. Servidia for the collaboration as the presented work also benefited from the Autonomous Low-Earth Orbit Formation Flying Technology (ALOFT) project from the National Commission of Space Activities (CONAE) in Argentina. I would like to express my gratitude to all my colleagues and friends at RMIT University for their friendship and support in areas other than academic endeavours. I am deeply grateful to my mother for the love, care, and sacrifices she has made for me throughout my upbringing, education, and preparation for the future. I want to express my deep gratitude to my sisters for the support they have given me and their prayers for me to be successful.

Table of Contents

List of Figures

List of Tables

Glossary of Terms

1D	One-Dimensional
2D	Two-Dimensional
ACO	Ant Colony Optimisation
AI	Artificial Intelligence
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Networks
AOCS	Attitude and Orbit Control System
AODE	Averaged One-Dependence Estimators
AOI	Area of Interest
AOS	Acquisition of Signal
APD	Avalanche Photodiode
API	Application Programming Interface
ASI	Italian Space Agency
AT-InSAR	Along-Track Interferometric Synthetic Aperture Radar
BBN	Bayesian Belief Network
BHGNP	Ben Halls Gap National Park
BN	Bayesian Network
BPNN/FFNN	Back-Propagation/Feedforward
CART	Classification and Regression Tree
CHAID	Chi-squared Automatic Interaction Detection
CHMI^2	Cognitive Human-Machine Interfaces and Interactions
CM	Communication Module
CNN	Convolutional Neural Networks
COTS	Commercial-Off-The-Shelf
CPH	Cyber-Physical-Human
CPS	Cyber-Physical Systems
CPU	Central Processing Unit
cuDNN	CUDA Deep Neural Network
DBM	Deep Boltzmann Machine
DBN	Deep Belief Networks
DL	Downlink
DL	Deep Learning
DNN	Deep Neural Networks
DoD	Depth-of-Discharge
DSL	Distributed System Layer
DSM	Distributed Spacecraft Mission
DSS	Distributed Satellite Systems
EA	Elevation Angle
ECI	Earth-Centred Inertial
ECSS	European Cooperation for Space Standardization
EDFAs	Erbium-Doped Fiber Amplifiers
EFB	Extreme Fire Behaviour
EM	Exception Maximisation

EO	Earth Observation
ESA	European Space Agency
EWE	Extreme Wildfire Events
FDA	Flexible Discriminant Analysis
FEEP	Field Emission Electric Propulsion
FF	Formation Flying
FL	Flame Length
FLI	Fireline Intensities
FOV	Field of View
FPGA	Field Programmable Gate Arrays
FSS	Federated Satellite System
GBM	Gradient Boosting Machines
GBRT	Gradient Boosted Regression Trees
GD	Gradient Descent
GEO	Geostationary Orbits
GNSS	Global Navigation Satellite System
GPU	Graphics Processing Unit
HDF5	Hierarchical Data Format version 5
HMS	Human-Machine System
HypSEO	HyperSpectral Earth Observer
IDF	Individual Discipline Feasible
iDSS	intelligent DSS
IMB	International Maritime Bureau
IoT	Internet of Things
IR	Intermediate Representation
ISL	Inter Satellite Links
ISR	Intelligence, Surveillance, and Reconnaissance
IUU	Illegal, Unreported and Unregulated
kNN	k-Nearest Neighbour
LARS	Least-Angle Regression
LASSO	Least Absolute Shrinkage and Selection Operator
LDA	Linear Discriminant Analysis
LEO	Low Earth Orbits
LOS	Loss of Signal
LSTM	Long Short-Term Memory
LVQ	Learning Vector Quantization
LWL	Locally Weighted Learning
MA	Multidisciplinary Analysis
MDA	Maritime Domain Awareness
MDF	Multidisciplinary Feasible
MDO	Multidisciplinary Design Optimisation
MDS	Multidimensional Scaling
MiDA	Mixture Discriminant Analysis
MINLP	Mixed Integer Non-Linear Problem
ML	Machine Learning
MLI	Multi-Layer Insulation

MLP	Multi-Layer Perceptron
MM	Mission Management
MOE	Mean Orbital Parameters
NCS	Neural Compute Stick
NLGBS	Nonlinear Block Gauss-Seidel Method
NLP	Non-Linear Programming
NLPR	Non-Linear Problem Reduction
NN	Neural Network
OBDH	On-board Data Handling
PC	Personal Computer
PCA	Principal Component Analysis
PCR	Principal Component Regression
PL	Payload
PLSR	Partial Least Squares Regression
PPP	Precise Point Positioning
PR	Processor
PRISMA	PRecursore IperSpettrale della Missione Applicativa
PSO	Particle Swarm Optimisation
QDA	Quadratic Discriminant Analysis
RAAN	Right Ascension of the Ascending Node
RAND	Research and Development
RBFN	Radial Basis Function Network
RF	Radio Frequency
RGB	Red, Green and Blue
RL	Reinforcement Learning
RNN	Recurrent Neural Networks
ROE	Relative Orbital Elements
ROS	Rate of Spread
RSO	Resident Space Objects
RTK	Real-Time Kinematics
RTN	Radial, Transverse, Normal
SAND	Simultaneous Analysis and Design
SAOCOM	Satélite Argentino de Observación COn Microondas
SAR	Synthetic Aperture Radar
SBSS	Space Based Space Surveillance
SDK	Software Development Kit
SFF	Satellite Formation Flying
SGD	Stochastic Gradient Descent
SIASGE	Satellite System for Emergency Management
SLSQP	Sequential Least-Squares Quadratic Programming
SMRS	Space Modular Self-Reconfigurable Satellite
SOPFEU	Societe De Protection Des Forets
SSO	Sun-Synchronous Orbit
SVM	Support Vector Machines
SWAD	System Wide Access Duration
SWAP	System Wide Access Percentage

SWIR	Short-Wave Infrared
TASO	Trusted Autonomous Satellite Operations
TIR	Thermal Infrared
TOA	Top of Atmosphere
TPU	Tensor Processing Unit
TT&C	Telemetry, Tracking, and Control
UN	United Nations
USA	United States of America
USB	Universal Serial Bus
VNIR	Visible to Near Infrared
WGS	World Geodetic System
WWII	World War II
XDSM	extended Design Structure Matrix

Abstract

For decades, satellites in outerspace have been designed as monolithic systems, which are extremely integrated systems that are aimed to accomplish a set of objectives that match specific user needs. These systems are made up of specific space, control, and ground elements that may go unused or be deactivated once the mission is over. Since space missions have typically been seen as highly customized endeavours, engineers have always worked on developing systems that do not share data and information with other satellites. The space industry is increasingly considering technologies such as Distributed Satellite Systems (DSS), particularly when combined with monolithic satellite systems, where studies indicate that performance is considerably improved while costs are reduced. Recent advancements in Artificial Intelligence (AI) technologies reveal that autonomy is vital in this modern era of space applications. Autonomy is required for enhanced implementation and operation, which can be accomplished by integrating AI techniques to satisfy space mission objectives. These tactics have proved their ability to perform, adapt, and respond to external environment changes without human intervention. Autonomy is provided because it is a critical attribute for steering the new distributed activities that require collaboration and coordinated approaches, allowing new structural functions such as opportunistic coalitions, resource sharing, and in-orbit data services. Trusted Autonomous Satellite Operations (TASO) is required within the DSS infrastructure to accomplish this. This research focuses on developing and using AI technologies for the TASO in DSS, which endows intelligent DSS (iDSS). Specifically focused on the evolution of space and control (on-board) segments required to maximize the performance of iDSS operations through advancements in Cyber-Physical Systems (CPS) and autonomous system designs. The Earth Observation (EO) missions based on iDSS have been investigated and analysed. A generic iDSS design optimisation methodology for EO that provides persistent coverage of the Australian territory is developed from the investigation.

Chapter 1

Introduction

This chapter provides the context for the doctoral thesis work, including the research background, potential research gaps, research questions, and objectives. A methodology for carrying out the research is presented. The limitations of the thesis work, as well as the structure of the article, are also discussed.

1.1 Background

The previous two decades have been an exciting time for space missions, with major participants in the space domain making a concerted effort to create and build autonomous mission ideas and concepts that are more difficult than ever. Owing to the continual success of interplanetary and Earth-orbiting missions, space engineering has pushed the boundaries for constant development, conceptualizing increasingly ambitious missions on a daily basis. Conventional, monolithic, high-performance spacecraft are not the only category of satellite systems impacted by this drive for innovation and ambition; smaller satellites are gaining traction due to newly developed technologies and a synthesis of the current state of the art. Small satellites, nanosatellites, and CubeSats are seeing renewed and never-before-seen interest and utilisation due to the game-changing properties possessed by this class of space systems. The major agencies and enterprises share the effort to use smaller satellites in the global landscape.

Significant reductions in space and launch segment costs of entry-level spacecraft are possible thanks to efforts in technology miniaturisation, the appearance of radiation-hardened Commercial-Off-The-Shelf (COTS), and tighter system integration. Small satellites frequently employ high computational capabilities within low power consumption and small form factors due to faster development cycles of COTS components. Compared to larger missions, this allows advanced and computationally-intensive autonomy methodologies to be run on-board [1].

Future space mission concepts are filled with low-cost Distributed Satellite Systems (DSS) working together to achieve complex mission objectives. Real-time multi-spacecraft coordination, data processing, and prioritization will not only optimise mission science return by establishing observational parameters of interest or success, but it will also facilitate outer solar system missions and missions in extreme environments (e.g., Io, Venus, subsurface Europa) where communication with ground operations and ground-based analysis times are limited. Through unparalleled levels of autonomy, this capacity will enable hitherto inconceivable classes of missions [2-4]. Implementing these envisaged space missions will necessitate considerable advancements in the capabilities of the architectures that implement them. DSS comprises numerous spacecraft that work together to fulfil a common mission goal [5]. In some circumstances, the DSS combine to generate a sensory system that would be impossible to create on a monolithic platform [6, 7]. In other configurations, they use distributed measurements to extract data on the spatial and temporal consequences of phenomena far larger than a single spacecraft can observe [8, 9]. Significant interest has been shown in addressing the technology required to support developing applications as missions requiring DSS become more and more important. On-board data processing, inter-satellite networks (often referred to as Inter Satellite Links (ISL)), and autonomous orbit control are the main topics of this thesis. These three technologies depend on one another. For instance, the processors on-board must have sufficient processing capability to process the data required to make conclusions as well as any calculations that may be involved in time-sensitive or computationally intensive decision-making. Moreover, the DSS spacecraft can communicate and coordinate among themselves without immediate ground control due to the inter-satellite networks. This functionality is essential because the ground link has limited bandwidth and latency, especially for far-space applications [10]. Additionally, it provides the option to gradually build the system in orbit, allowing for the construction of various modules at various stages. The modular architecture theory serves as the foundation for DSS architecture. The study by the Research and Development (RAND) shows that [11]:

a) Distributed constellations may weigh less and cost less to launch.

b) Distributed satellites may perform better during deployment.

c) Distributed satellite constellations may be able to fail more gracefully.

d) Distributed satellite constellations may be more survivable in an attack.

DSS's primary goal is to deliver a more responsive and resilient solution to meet the expanding demands of the scientific community and also the defence sector by aiding in the measurement and prediction of Earth Observation (EO) missions [12], Space-Based Space Surveillance (SBSS) missions [13-16] and Astronomy and Astrophysics missions [9, 17]. DSS has several advantages (i) Simultaneous multipoint data collection, (ii) Increased availability, (iii) DSS can look at different things at once, (iv) Reduced downtime and graceful degradation. A DSS that includes these enabling technologies can provide four major benefits [4]:

a) *Distributed Coordination:* can share data and change what they prioritize.

b) *Autonomous Re-tasking:* can respond to environmental stimuli autonomously without requiring intervention from a ground operator.

c) *Increased Availability:* when only a single spacecraft can be reached, it can relay commands to the others.

d) *Workload Balancing:* can re-task satellites based on available computation, power, and communications resources.

This research aims to demonstrate the DSS Trusted Autonomous Satellite Operations (TASO) for Mission Management (MM), such as wildfire detections. The findings of such analyses could be helpful for future time-critical missions, i.e., disasters and rare events, and the following novel contributions have been made from an intelligent DSS (iDSS) perspective:

a) ***Mission Astrionics:*** Reactive elements, such as Artificial Intelligence (AI), is integrated with the DSS to achieve TASO for on-board data processing to provide real-time/near real-time alerts. To accomplish the same, a Deep Learning (DL) model is developed and demonstrated for detecting wildfires on-board the satellite using optical payload, i.e., hyperspectral imagery.

b) ***Service Astrionics:*** For the TASO, the intelligent DSS (iDSS) will reconfigure either based on the (i) detection of disaster event (wildfire), (ii) based on the requirements of the owner/operator for the requested duration, (iii) to evade from the Resident Space Objects (RSO), (iv) to avoid the collision between the satellites/modules.

It is essential to mention that the detection of wildfires and Maritime Domain Awareness (MDA) should be treated as an example test case and that the suggested methodology (or ones similar to it) can be successfully applied to other scenarios or activities, as has already been explored and shown in other publications [18].

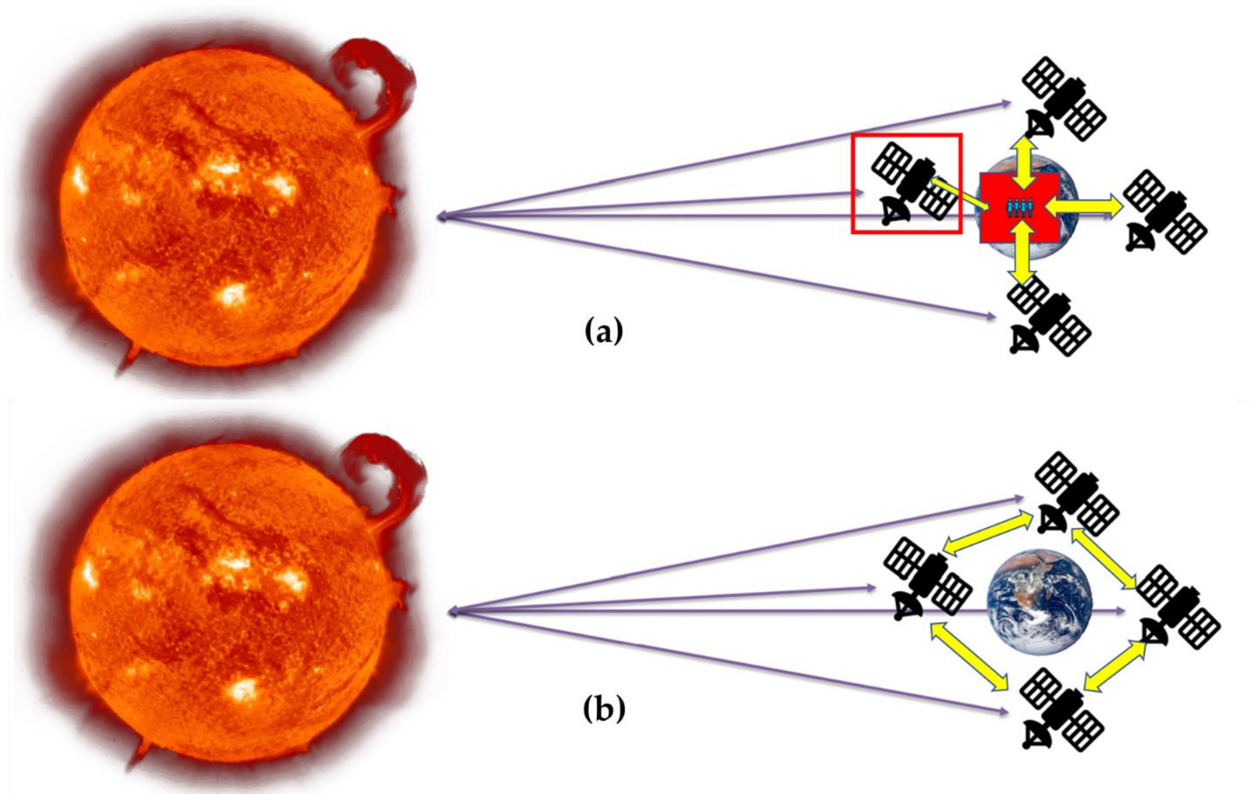


Figure 1: DSS operations (a) without ISL (b) with ISL.

Sharing information about the acquired data is made possible by iDSS, allowing maximum scientific output to be achieved through opportunistic research. The operational requirements can be lowered with iDSS autonomy, allowing for human-in-the-loop operations to be converted into human-on-the-loop activities. Humans will be responsible for overseeing the operations in some capacity. Despite the loss of one spacecraft, iDSS is able to continue working at normal levels, and its trusted autonomous reconfiguration capabilities allow it to redistribute workload without interference from the ground. Figure 1 illustrates the differences between the current DSS and iDSS operations. Figure 1 (a) illustrates how the data is transmitted to the human operators on the ground before being relayed to the remaining satellites, which is not ideal for time-sensitive applications like rare events and disasters. In Figure 1 (b), where the ISL allows for data sharing, and reactive elements allow on-board processing, allowing the system to respond quickly.

1.2 Motivation

Satellite systems provide a wide variety of services, which can be easily accessed from almost any global location. These systems have rapidly evolved over the last few decades and have become essential in various application domains, such as communications, navigation, EO and astronomy [19]. However, certain aspects of satellite technology, such as trusted autonomous operations, remain to be explored due to the increasing complexity of hardware/software components and associated safety, integrity and cyber-physical security concerns [20].

Present-day autonomous systems can execute intelligent functions (e.g., decisions and/or actions traditionally performed by humans) using various computer-based algorithms, such as AI. This requires the ability to gather real-time data from the external operational environment (i.e., sensing), to perform inference and/or decision-making functions, and to execute proper actions if and when required. Despite the significant progress made in hardware and software technologies, TASO is still largely a research topic and significant investments are needed to fully exploit the anticipated safety and efficiency and sustainability benefits that such operations would bring, possibly leading to the progressive removal of present-day socio-political barriers such as AI ethics, liability and public trust [21]. In many applications, fully autonomous satellite operations are either impractical or undesirable, mainly because a minor error can result in the loss of millions of dollars and, in some cases, lead to human casualties (point-to-point suborbital space transport, Earth-orbiting inhabited space stations, etc). Therefore, an acceptable level of trust is required for near-Earth operations, especially considering the steady increase of RSO in Low-Earth Orbits (LEO) and Geostationary Orbits (GEO) [4, 22]. Furthermore, to facilitate further progress in TASO research, it is essential to address the implications of trusted autonomy and AI in the evolution of Cyber-Physical Systems

(CPS) for space applications, including the co-evolution of system-level requirements (i.e., communication, control and computing) and human-autonomy interactions. Current research trends in this area show that Cyber-Physical-Human (CPH) architectures are evolving with the widespread adoption of Machine Learning (ML) and hybrid AI techniques (e.g., neuro-fuzzy inference engines) and becoming progressively more capable of modulating both the levels of autonomation and the human command/control functions towards achieving specific goals. In this context, the current generation's participation is in an evolutionary process, where humans are progressively transitioning to a high-level supervisory role [23].

Clearly, AI will play a significant role in easing the transition to TASO. A radical departure from conventional system design and development is required to meet the intelligence requirements of future trusted autonomous space system vehicles and intelligent operation in highly integrated and information-rich environments. Going forward, certification and explainability of these AI systems will be critical, particularly in outer space operations where liability is required for the damages these systems cause. As a result, there is a need to understand the associated technical and legal challenges with this system.

1.3 Research Gaps and Questions

There is currently no comprehensive classification of satellite systems in the body of published work, which is one of the most serious shortcomings in the existing DSS. Second, when looking at Australia, wildfires have become a significant problem over the course of the last few years. In addition, indigenous capabilities are lacking to provide disaster event management in real-time/ near real-time. There is the potential to develop iDSS solutions that can change their structure and function and reconfigure mission profiles in response to operational and environmental indicators.

The design of a state-of-the-art satellite system has not yet accounted for the contingency planning aspect of iDSS, which is an essential component of the system. This refers to the capability of the satellite to perform data processing on-board and detect a disaster event or monitor a particular Area of Interest (AOI), and then downlink only the actionable information to the receiver with enough time to dynamically determine which appropriate restorative or reconfiguration actions to take in order to ensure that the system can continue to perform its mission objective.

In addition, the optimal combination of physics-based methodologies and data-driven/AI inference techniques for different engineering application domains to deliver TASO has not been determined. This is a problem because physics is the foundation of engineering. This thesis focuses on the following research questions in order to address these gaps in the literature:

RQ1: How can Artificial Intelligence (AI) techniques be employed in DSS architectures for Earth Observation (EO) operations to enhance the performance of both service and mission astrionics systems?

RQ2: How can intelligent DSS (iDSS) be designed for Disaster Management and Maritime Domain Awareness (MDA)?

RQ3: How can we develop a generic iDSS design optimisation methodology for EO that provides persistent coverage of the Australian territory?

1.4 Research Aim and Objectives

This research aims to enhance the development of DSS systems intended to function in information-rich and networked environments. The following set of clear objectives has been established in order to accommodate the intelligence requirements for future autonomous aerospace vehicles.

- Conduct a thorough and in-depth review of the DSS current state-of-the-art to find new requirements for TASO.
- Identify AI inference techniques for wildfire detection and develop an iDSS for real-time/near real-time disaster management. Finally, identify and implement mission management and reconfiguration options to ensure an acceptable level of operational capability for wildfire management.
- Develop an iDSS mission for monitoring Australia's Maritime with autonomous orbit control.

- Develop a Multidisciplinary Design Optimisation (MDO) methodology for iDSS to ensure persistent coverage over Australia.

Exploring the role and capabilities of AI-based algorithms to increase the mission and system autonomy of iDSS missions significantly. To this end, AI-based methods and algorithms can be integrated into space missions with the intention of boosting the self-sufficient decision-making functionalities of the space segment. This can be accomplished by improving the space segment's capabilities in the following areas:

- Execution of tasks that were not defined during the development of the spacecraft,
- Optimisation of on-board resources and execution of specific tasks, such as on-board data processing,
- Emulation of the expert knowledge that is necessary for mission operations.

Therefore, it ultimately reduces operations costs for future iDSS operations through relatively small operations teams and far less frequent usage of massive deep space ground station network antennas.

1.4.1 Limitations

The research presented in this thesis centred on identifying and implementing algorithms for EO intelligent DSS operations. However, it is important to remember that the thesis was developed as part of the doctoral thesis and that the research was not conducted to determine which of the available algorithms is the best for performing TASO on a specific type of task; rather, the research aimed to demonstrate the practicability of TASO in iDSS.

This research project also used commercial hardware components, free EO data for academic research, and open data sets. This investigation aims to identify the emerging design features that characterize iDSS, develop iDSS for EO missions over the Australian territory and extract lessons of general applicability for establishing an MDO methodology. More investigation and comparisons will be necessary to ascertain whether the suggested iDSS architecture and algorithms are the best solutions for resolving the issue addressed in the case studies.

1.5 Research Methodology and Thesis Outline

Following the completion of a comprehensive review of the relevant prior research, an in-depth investigation into the DSS was carried out to determine its applicability in space operations. The research questions and objectives are framed to support the research work after identifying the key areas where DSS may be beneficial. The current space research priorities of SmartSat CRC and the Australian Space Agency and future advances in related areas (particularly emerging intelligent space system opportunities and EO Road map) are taken into consideration when framing the research questions and objectives. Initially, a DSS was developed to provide continuous coverage over Australia for wildfire management. The subsequent step was to enable real-time or near real-time management of wildfires using the AI techniques that were used to deliver data processing on-board the satellite, from which only actionable information that can be acted upon is downlinked. After it has been proven that on-board capability is feasible, iDSS operation can be accomplished by integrating these astrionics, i.e., hardware accelerators in the iDSS architecture. It has been proposed to endow TASO with an iDSS so that real-time Intelligence, Surveillance, and Reconnaissance (ISR) operations can be carried out to support maritime monitoring. The results have been presented at conferences and published in peer-reviewed scientific journals to accomplish the objectives and answer the questions being investigated. The detailed research methodology of the thesis work is shown in Figure 2. This project's research work is divided into phases that accomplish the objectives outlined in Section 1.4. An extensive literature review is conducted, and the second chapter of the thesis contains a review of the current state-of-the-art in DSS, as well as key advancements and contributions to knowledge in the field of iDSS for the aerospace industry.

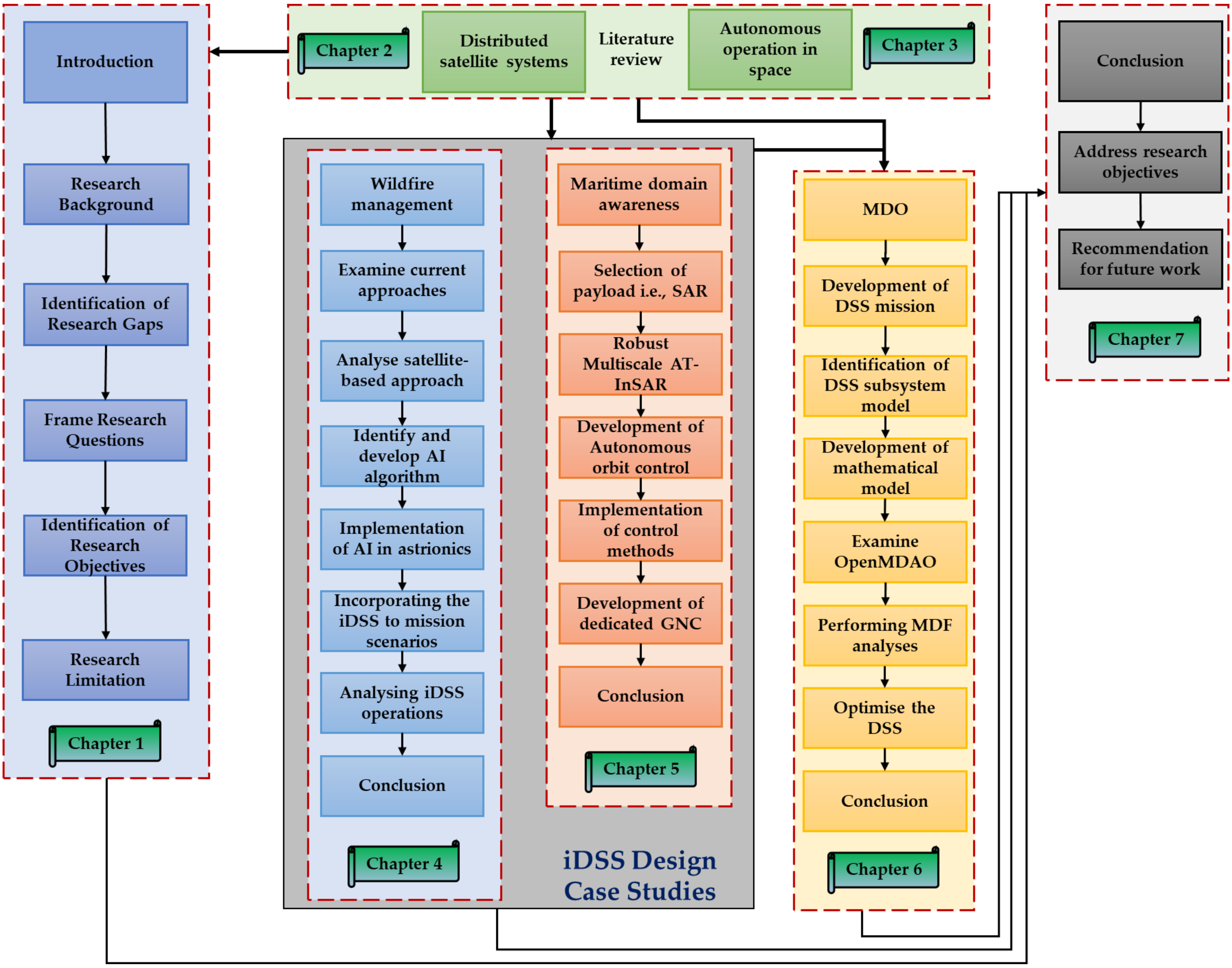


Figure 2: Research methodology.

Figure 3 depicts the thesis structure. In Chapter 3, the review's focus is broadened to include a thorough analysis of autonomous space operations. Chapter 4 discuss the real-time/ near real-time disaster management using iDSS, for the same Australian Bushfire has taken as case study. Chapter 5 is devoted to maritime management using iDSS. In Chapter 6, multidisciplinary iDSS design and optimisation is carried out to provide persistent coverage over Australia. Chapter 7 summarises the key findings and provides recommendations for future research.

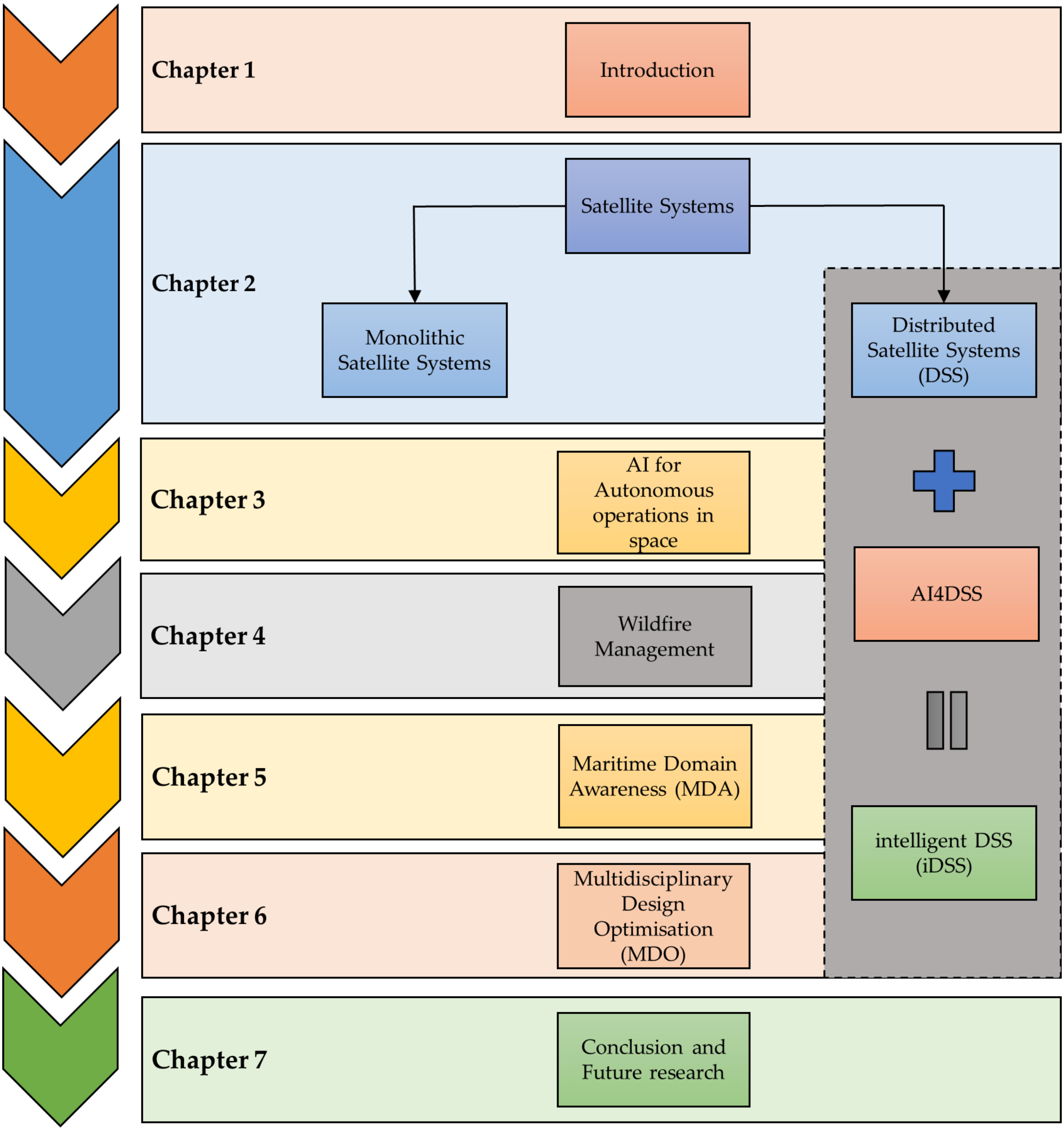


Figure 3: Thesis structure.

1.6 List of Publications

Published

Journals

1. **Thangavel. K,** Spiller. D, Sabatini. R, Marzocca.P and Esposito. M, *"Near Real-Time Wildfire Management Using Distributed Satellite System"*, in IEEE Geoscience and

Remote Sensing Letters, vol. 20, pp. 1-5, 2023, Art no. 5500705, doi: 10.1109/LGRS.2022.3229173 (IF: 5.343, Q1).

2. **Thangavel, K.;** Servidia, P.; Sabatini, R.; Marzocca, P.; Fayek, H.; Cerruti, S.H.; España, M.; Spiller, D. "*A Distributed Satellite System for Multibaseline AT-InSAR: Constellation of Formations for Maritime Domain Awareness Using Autonomous Orbit Control*". Aerospace 2023, 10, 176. https://doi.org/10.3390/aerospace10020176 (IF: 2.660, Q1).

3. **Thangavel, K.,** Spiller, D., Sabatini, R., Amici,S., Longepe,N., Servidia, P., Marzocca, P., Fayek, H., L. Ansalone. "*Trusted Autonomous Operations in Distributed Satellite Systems for Mission Management Using Optical Sensors*", MDPI Sensors (IF: 3.847, Q1

4. Miralles, P., **Thangavel, K**., Scannapieco, A. F., Jagadam, N., Baranwal, P., Faldu, B., ... & Stepanova, D. (2023). *"A critical review on the state-of-the-art and future prospects of Machine Learning for Earth Observation Operations".* Advances in Space Research (IF: 2.611, Q1).

5. Khaja Faisal Hussain, **Thangavel, K.**, Gardi, A., Sabatini, R. "*Passive Electro-Optical Tracking of Resident Space Objects for Distributed Satellite Systems Autonomous Navigation*", Remote Sensing (IF: 5.349, Q1)

6. **Thangavel, K.,** Spiller, D., Sabatini, R., Amici, S., Sasidharan, S.T., Fayek, H. and Marzocca, P., 2023. "*Autonomous Satellite Wildfire Detection Using Hyperspectral Imagery and Neural Networks: A Case Study on Australian Wildfire*", Remote Sensing, 15(3), p.720 (IF: 5.349, Q1).

7. Ranasinghe, K., Sabatini, R., Gardi, A., Bijjahalli, S., Kapoor, R., Fahey, T., & **Thangavel, K.** (2022, January), "*Advances in Integrated System Health Management for mission-essential and safety-critical aerospace applications*" Progress in Aerospace Sciences, 128, 100758. https://doi.org/10.1016/j.paerosci.2021.100758 (IF: 8.653, Q1)

Conference

8. **Thangavel, K.,** Servidia, P., Sabatini., R., Marzocca, P., Fayek, H., Spiller, D "*Distributed Satellite Systems for Maritime Domain Awareness*", AIAC 2023, Australia.

9. Khaja Faisal Hussain, **Thangavel, K.**, Gardi, A., Sabatini, R. "*Autonomous Optical Sensing for Space-Based Space Surveillance*", IEEE AeroConf 2023, USA.

10. Khaja Faisal Hussain, **Thangavel, K.**, Gardi, A., Sabatini, R. "*Autonomous tracking of Resident Space Objects using multiple ground-based Electro-Optical sensors*", SpaceOps 2023, UAE.

11. **Thangavel, K.,** D. Spiller, R. Sabatini, P. Servidia, P.Marzocca, H. Fayek, K.Faisal, A. Gardi., "*Trusted Autonomous Distributed Satellite System Operations for Earth Observation*", SpaceOps 2023, UAE.

12. Thomas, G., **Thangavel,** K., Anne-Sophie,M. *"New Legal Challenges in International Space Law: Artificial Intelligence and Liability"*, SpaceOps 2023, UAE

13. D. Spiller, **Thangavel, K,** S. T. Sasidharan, S. Amici, L. Ansalone and R. Sabatini, *"Wildfire segmentation analysis from edge computing for on-board real-time alerts using hyperspectral imagery,"* 2022 IEEE International Conference on Metrology for Extended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE), Rome, Italy, 2022, pp. 725-730, doi: 10.1109/MetroXRAINE54828.2022.9967553.

14. **Thangavel, K.**, Spiller, D., Sabatini, R., & Marzocca, P. *"On-board Data Processing of Earth Observation Data Using 1-D CNN"*, SmartSat CRC Conference 2022.

15. **Thangavel, K**, J. J. Plotnek, A. Gardi and R. Sabatini, *"Understanding and investigating adversary threats and countermeasures in the context of space cybersecurity,"* 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), Portsmouth, VA, USA, 2022, pp. 1-10, doi: 10.1109/DASC55683.2022.9925759.

16. Meyrick, E., Pickard, A., Rahlloff, T., Bonnart,S., Carlo, A., **Thangavel, K.** *"Ground Station as a Service: A Space Cybersecurity Analysis"* 72nd International Astronautical Congress, IAC 2020, UAE.

17. Wischert D, Baranwal P, Bonnart S, Álvarez MC, Colpari R, Daryabari M, Desai S, Dhoju S, Fajardo G, Faldu B, González EL,…, **Thangavel, K**. *"Conceptual design of a mars constellation for global communication services using small satellites"* 72nd International Astronautical Congress, IAC 2020, UAE.

Chapter 2

Review of Satellite Systems

This chapter provides a review of spaceflight systems, with a particular emphasis on Earth orbit systems, where the DSS is expanded and reviewed further. A detailed discussion of DSS unified classification is presented. The DSS hardware and software architecture is also presented.

Thousands of active satellites are currently orbiting Earth [24]. The satellite's size, orbital parameters, and design depend on its intended purpose. The classification of spaceflight systems adopted in this article is presented in Figure 4. Broadly, spaceflight systems can be grouped into three categories: (1) Space exploration systems [4]; (2) Earth orbital/sub-orbital transport system [25]; (3) Earth orbit satellite systems. Earth satellite systems can be further divided into the following categories: (i) Monolithic satellite systems, and (ii) Distributed satellite systems which are discussed broadly in the following sections.

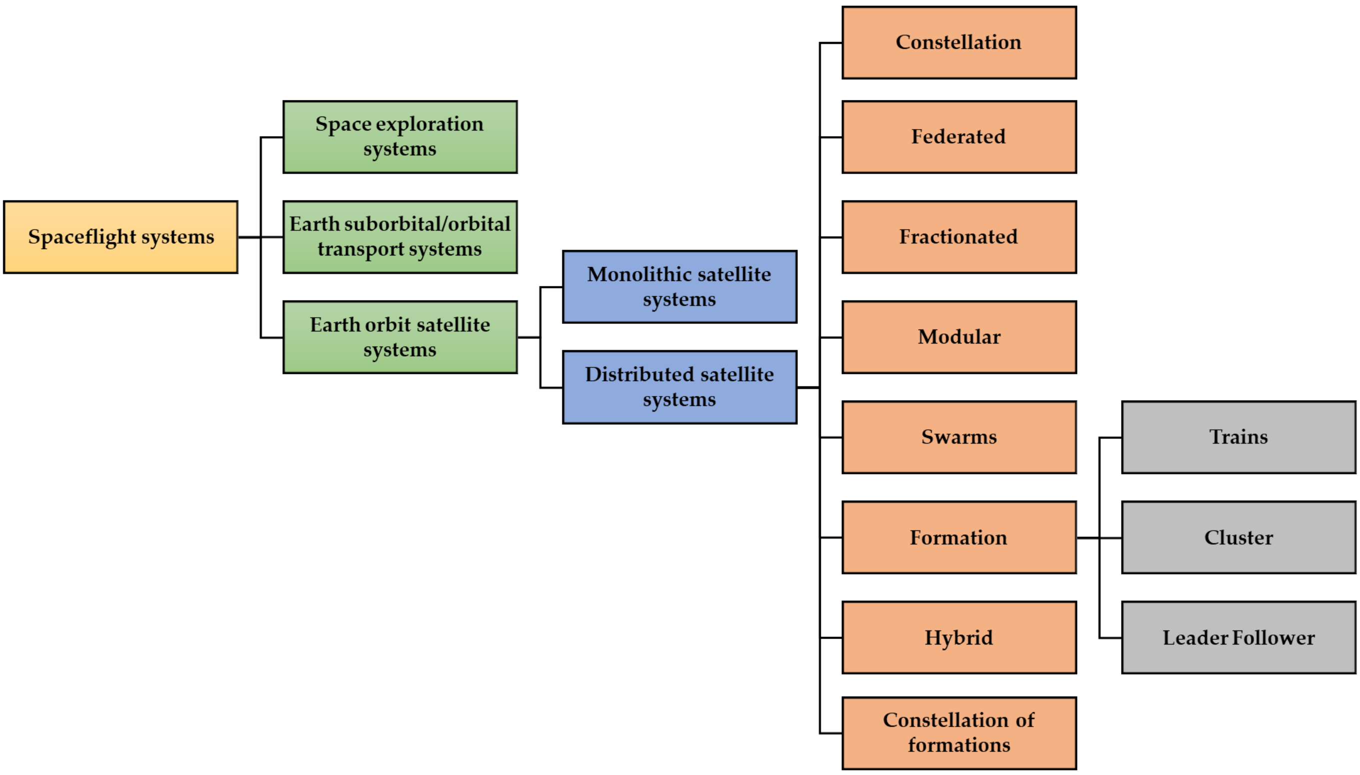


Figure 4: Classification of spaceflight systems.

2.1 Monolithic Satellite Systems

If a satellite system has modules or subsystems and is physically independent of other space assets, it is classified as a monolithic system. Adding redundant components increases their reliability while also increasing the system's overall weight, making it more expensive. Monolithic systems are still a large fraction of spacecraft being deployed in missions such as deep space exploration, technology demonstration, universities and research centres [11, 26]. A typical monolithic satellite system has the following modules PR: Processor, PL: Payload, DL: Downlink, CM: Communications Module, BUS, which carries all the modules as depicted in Figure 5. Monolithic satellite systems are comprehensively detailed in the works [27, 28].

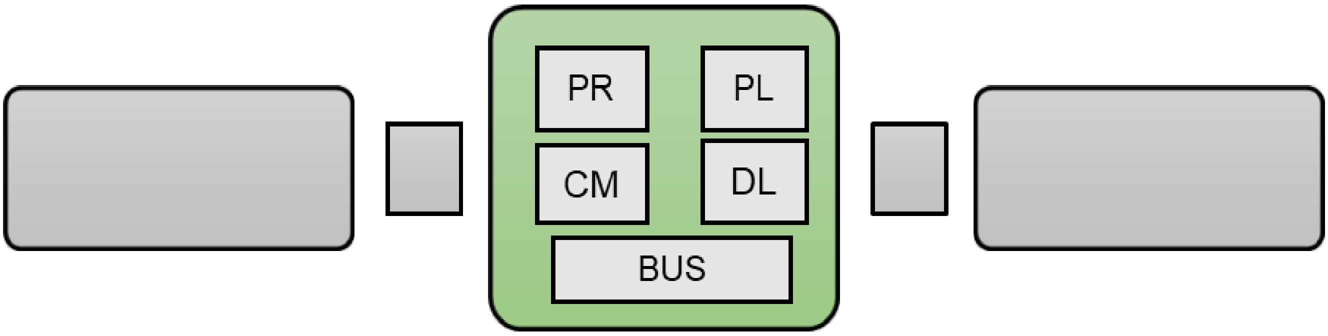


Figure 5: Monolithic satellite system. Adapted from [28].

2.2 Distributed Satellite Systems

A DSS consists of multiple spacecraft working together to achieve one or more common objectives. A DSS is a type of satellite architecture in which the functional capabilities are shared among many space assets that communicate via wireless networks [29]. The DSS concept is gradually changing the physical connectivity of various components in a satellite system into wireless connections, typically using optical communication methods, i.e., Inter Satellite Links (ISL). DSS is a mission architecture that shifts away from monolithic systems and more towards multiple spacecraft/modules of elements that communicate, interact, and cooperate with one another. They communicate via ISL, resulting in new systemic properties and/or emerging functions. Dividing a spacecraft over many launches reduces risk, ensuring that the whole system is not lost when a launch fails. It also offers the flexibility to gradually construct the system in orbit, which allows the development of different modules/satellites at distinct stages. DSS are mission designs comprised of several spacecraft modules that cooperate, communicate, and interact with one another, resulting in the emergence of novel system attributes and/or functions [30]. The following concepts of modularity are required in order to have a better understanding of the DSS concept.

2.2.1 Modular Architecture

Modularity is a feature of systems that quantifies the degree to which a system's functionalities can be subdivided into distinct modules or clusters which interact more with each other [31, 32]. Damage to one module can cascade to subsequent modules in a highly interconnected system with minimal modularity, enhancing the risk of a system-wide failure [33]. On the other hand, a disturbance to one component may be best controlled in a system along with a high level of modularity. Modularity is often explored as a spectrum of several levels and forms of a system that exist as a continuum within the system and not a binary property [34, 35]. Further, continuous modularity can be intuitively and methodically represented and quantified for some satellite systems that are now being introduced for a subset of elements in a network system [32, 36]. However, it can be challenging for other engineered systems to deal with the continuous spectrum in system architecture decisions for several reasons. One reason for this is that it would transform the decision problem into an optimisation problem. This uses a general continuous spectrum that becomes computationally intractable and may not reconcile with the engineering design required for such decisions. Furthermore, interpreting modularity as a continuum does not fit hierarchical and layered structures, which results in a discontinuity in the level of modularity when a new layer is added. A hybrid method is mostly utilised to handle this issue since it preserves the spectral aspect of modularity while discretising it into many stages, each reflecting a distinct class of modularity. Modularity can be conceived as continuous inside each level (or, if necessary, further discretised), whereas changes in the stage of modularity are viewed as discontinuous shifts [37].

2.2.2 Modularity Spectrum and Decision Operators

In accordance with the broad concept of modularity [31], the framework consists of five modularity stages identified from M_0 to M_4, as shown in Figure 6. The spectrum is discretised into five key modularity stages, which allows for computational feasibility of the framework. This category comprises fully integral architectures (M_0), integral yet decomposable architectures (M_1), modular yet monolithic architectures (M_2), static distributed architectures (M_3), and dynamic distributed architectures (M_4). A set of value operators to quantify the net operator (M^+ Operator), which modifies the modularity levels between two neighbouring stages on the spectrum. When used in conjunction with M^+ Operators, the spectrum can help designers choose appropriate parameters and put together a system-specific computational tool using a number of pre-existing tools and approaches [28, 31, 35-37]. To comprehend how distributed satellites are distinct from monolithic satellites, one must understand the concept of Modularity. The transition from M_0 to M_1 is referred to as *Decomposition*, from M_1 to M_2 as *Splitting*, from M_2 to M_3 as *Fractionation* and from M_3 to M_4 as *Resource Sharing*. In satellite architecture, M_0, M_1, and M_2 are considered monolithic systems, whereas M_3 and M_4 represent distributed architecture systems.

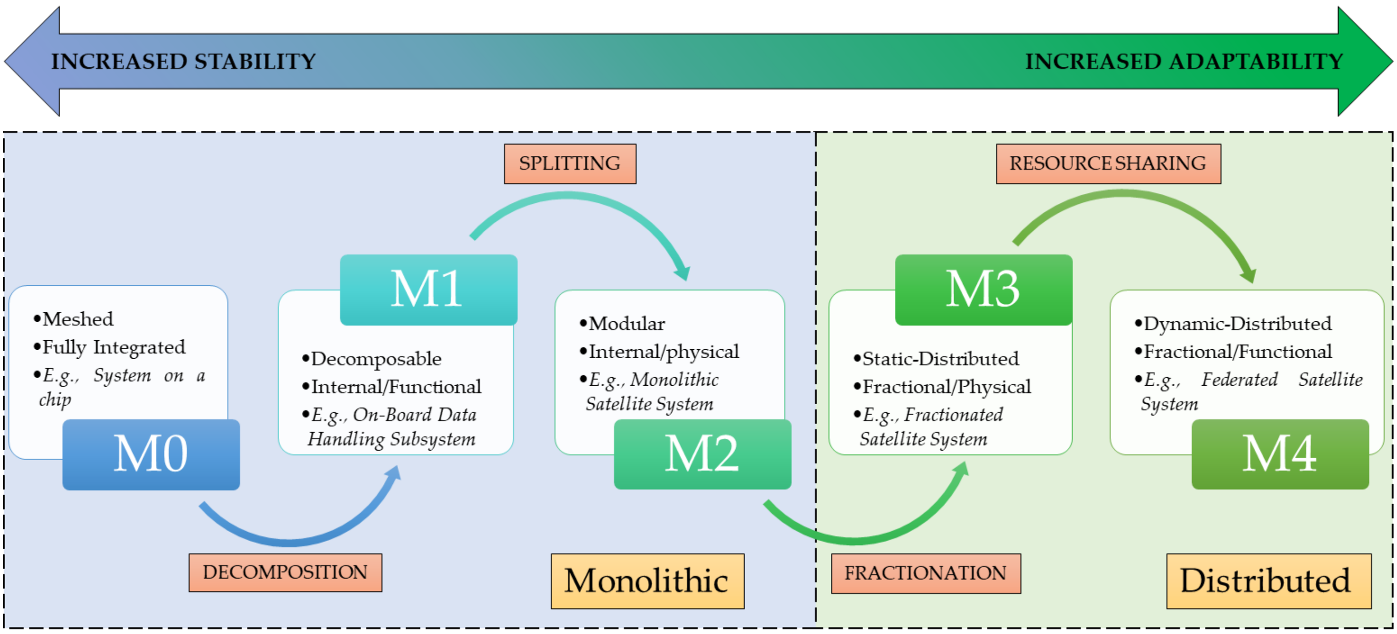


Figure 6: A five-stage modularity with distributed architecture spectrum and M^+ operations. Adapted from [37].

While M_0, M_1, and M_2 cover all instances of modularisation for monolithic systems (systems with only one physical unit), M_3 and M_4 cover systems with multiple units (distributed systems) and the possibility of communication between them [38]. M^+ decision operators express the modularity of any architecture by adding a decision layer to a model; as a result, the conceptual framework has been transformed into a computational tool. This determines the best level of modularity for a certain system's functionality in a given environment profile. The decision entails both the modularity phase and the design implementation inside that phase. By including a set of operators (M^+ operators) for calculating the transition value from one stage of modularity (M_x) to its next immediate phase (M_{x+1}). By computing the probability distribution of the difference in value between two consecutive phases, the suggested decision-making operators evaluate the performance of the system before and after operation [38]. This will allow conclusions to be drawn based on an average value difference as well as the level of risk tolerance. For most engineering systems, M_1 is the lowest modularity, so the splitting operation, the first decision operation, suggests the changeover from M_1 to M_2 through the development and use of proper standard interfaces. Fractionation operation by shifting one or more of its subsystems to other fractions takes a system from M_2 to M_3. Although M^+ evaluation specifics a procedural algorithm which is dependent on particular systems and its parameters, which acts as a decision-making evaluation engine [37].

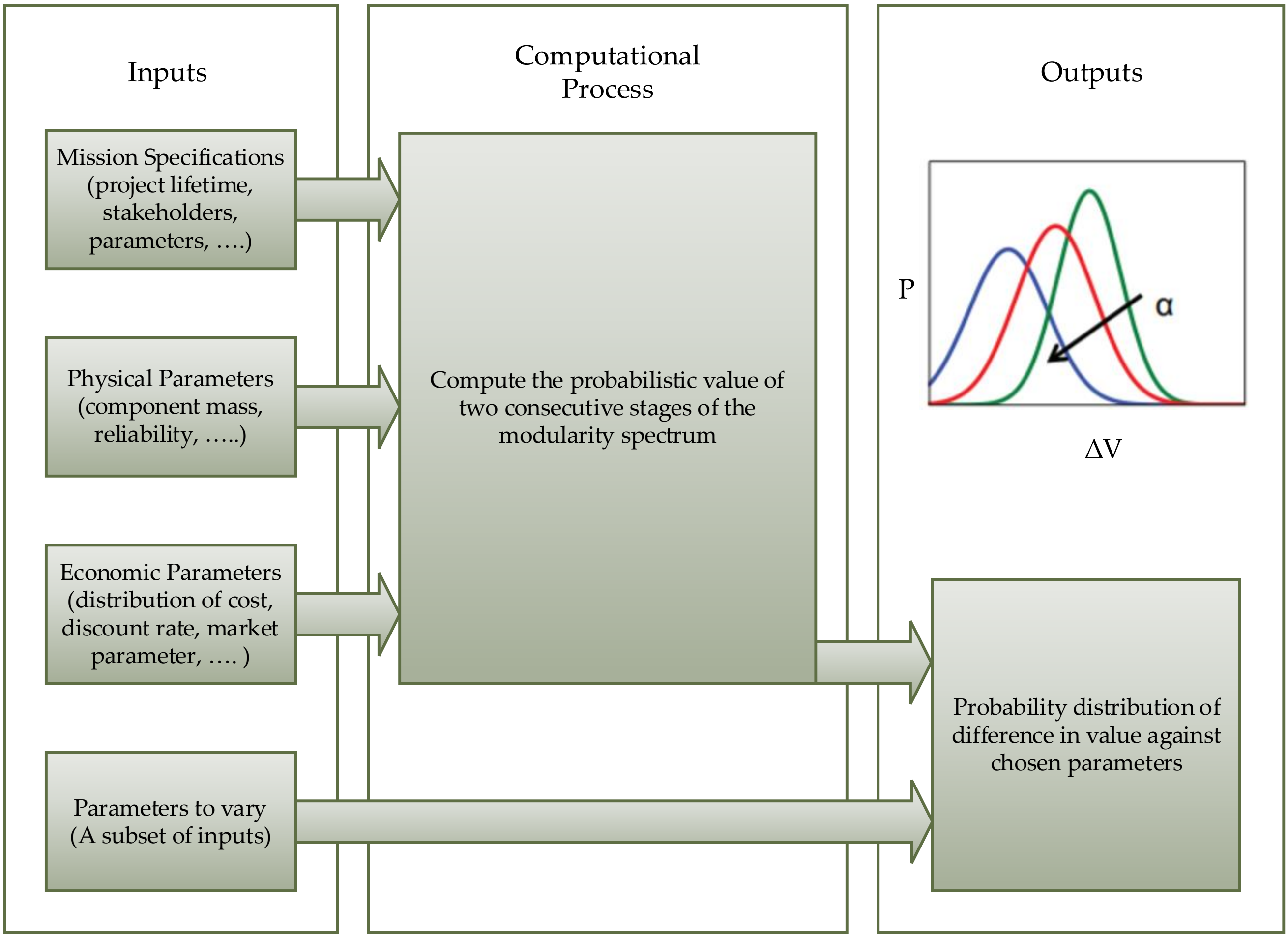


Figure 7: Quantifying the value of the M^+ operation. Adapted from [37].

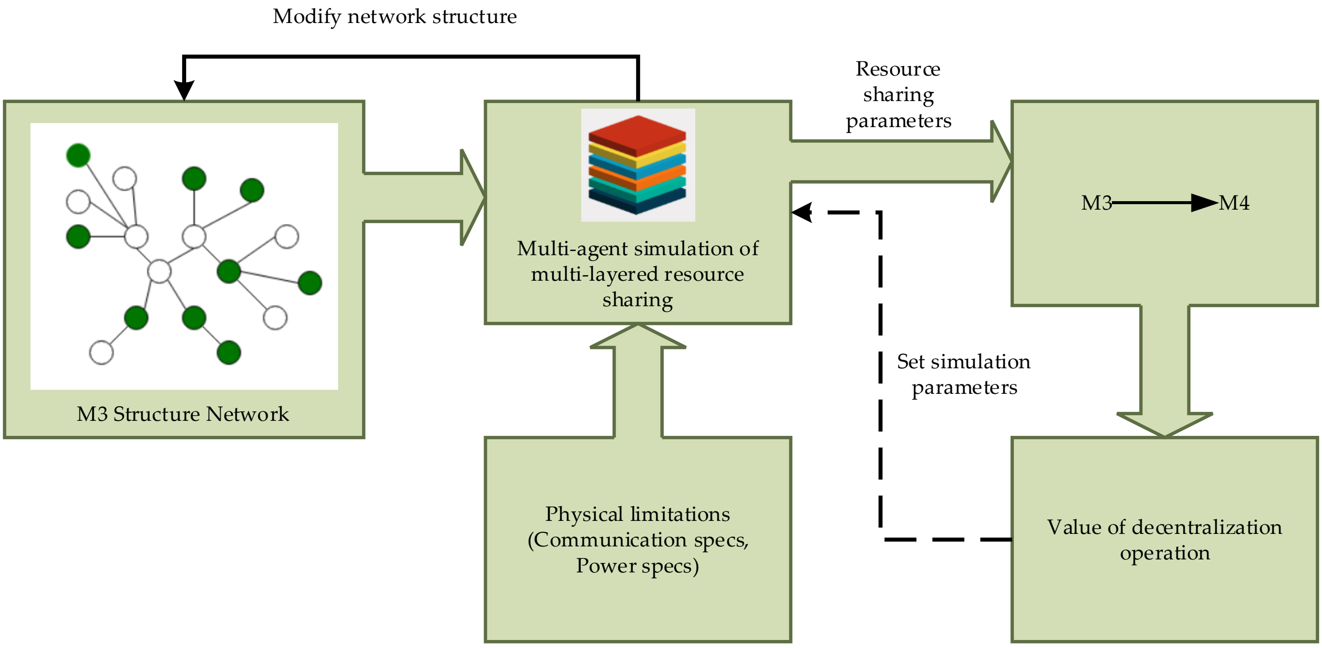


Figure 8: Calculating the value of the decentralisation operation (M3 to M4). Adapted from [37].

The M^+ value is measured by comparing the system's value prior and post its operations. Such assessment involves knowledge of the system and its settings [28, 31, 32, 35, 37, 38].

Figure 7 shows the input and output characteristics for evaluating M^+ Operators. At each level of modularity, the system's value is determined using one of the common system assessment methods (e.g., discounted cash flow analysis, scenario analysis) while taking the subsequent criteria into account [37, 38]:

a) **Technical Parameters:** For instance, the probability density of a failure, the time required for an upgrade to become available, the highest number of modules allowable, and the maximum transmission bandwidth permitted.
b) **Economical Parameters:** For instance, the number of modules in need at a given time, the cost of launching and operating a module, and the rate at which distinct module types generate value.
c) **Life Cycle Parameters:** Total time required for operation, budget, as well as maximum time required for initial deployment.

A high-level sketch for calculating the decentralisation values from M_3 to M_4 is depicted in Figure 8. Because of the underlying network structure, designers must use multi-agent techniques that blend system dynamics and evolution with autonomous behaviour [39].

2.3 DSS Classification

DSS are categorised based on the type of mission and function they perform. Activities required to meet local objectives (i.e., those specific to each module) or small bits of a global objective's functioning (i.e., particular to the infrastructure) may be included in modules performing activities in a distributed infrastructure, whether in monolithic systems or distributed spacecraft. As a result, the function type is measured in terms of how dispersed the mission's goals are, ranging from no collaboration between modules (i.e., local functionality) to a fully functional symbiosis (i.e., distributed functionality).

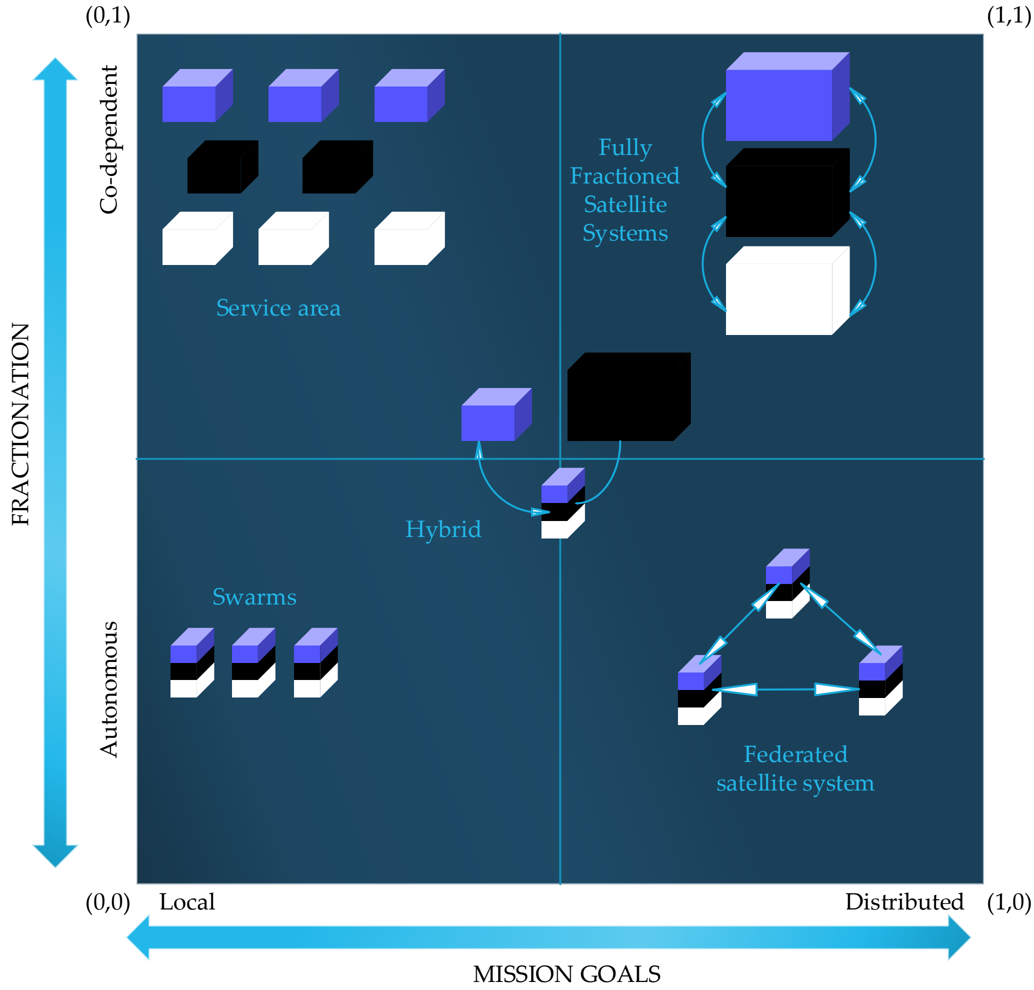


Figure 9: Distributed Satellite system classification. Adapted from [40].

As a result, different distributed missions are characterised according to their degree of distribution in terms of the system's capabilities or goals and resource interdependence between modules. A bi-dimensional space can be formed using the analysis of these two domains, as shown in Figure 9, with values in the range [0,1]. The x-axis shows the degree of mission goal distribution, which ranges from missions in which satellite modules work together to advance a single global function to goals in which each satellite module develops its own local activity. The y-axis shows the degree of fractionation among scenarios where modules are totally reliant on one another and cases where nodes are completely resource self-sufficient [40, 41]; both axes are independent. The following are the classification of DSS.

- Constellation
- Fractionated

- Federated
- Modular
- Swarms
- Formation
- Constellation of formations
- Hybrid Missions

2.3.1 Constellation

A satellite constellation is a collection of human-made spacecraft that operate as a unified system. A satellite constellation, as opposed to a single satellite, can provide continuous global or near-global coverage, as shown in Figure 10 (a). This means that at any given time, at least one of the satellites in the constellation will be visible somewhere on Earth. Satellites are often positioned inside sets of orbital planes that are complementary to one another and connected to ground stations spread across the world. It's also possible that they communicate with one another via satellites. A satellite constellation is a system of artificial units that are identical to one another or of a similar sort, all of which share the same purpose and control. These groups communicate with ground stations placed all over the world and, at times, even communicate with one another. They are designed to function together as a system and complement one another in some way. The works of literature provide descriptions of some of the satellite constellations [42-46].

2.3.2 Fractioned

Fractionated satellite is a system in which a spacecraft is divided into smaller units or fractions collaborated to achieve a common mission objective. The satellite consists of co-dependent modules that require system resources to be exchanged to function, as shown in Figure 10 (b) [47]. Two extremes can be thought of for this category based on task achievement. In the first instance, the satellite's functionalities are implemented by the satellite fractions, which need services such as data processing, power, communication link, etc., to complete the functions calling for dedicated fractions to provide these services. Fractionated systems have mission objectives that are specific to each of their fractions/modules. Though there is minimal cooperation between them, each fraction is still highly dependent on the infrastructure of the system. Secondly, fully fractionated satellite systems have modules that collaborate on accomplishing the mission's global objective. There is considerable resource dependency in this scenario and functionalities of the modules [40, 48-53].

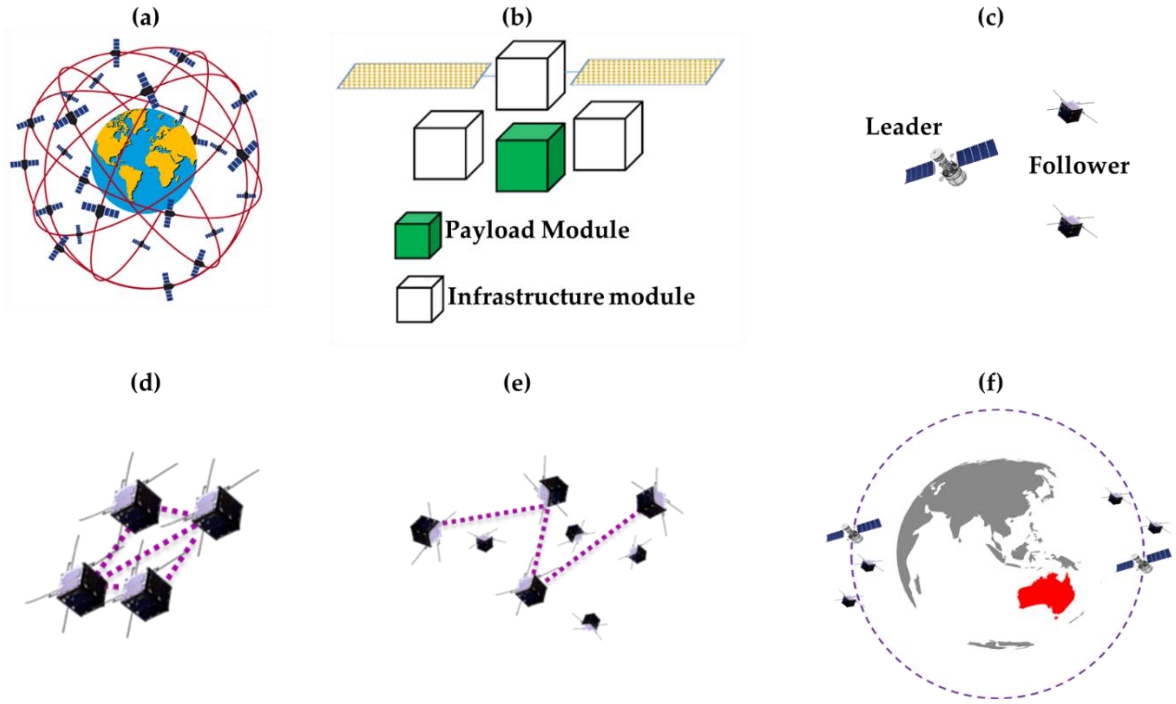


Figure 10: DSS types (a) constellation (b) fractionated (c) formation (d) modular (e) swarms (f) constellation of formations.

2.3.3 Federated

In a federated system, a group of satellites work together to provide a specific service, but each satellite operates independently, with its own mission and communication capabilities. A Federated Satellite System (FSS) is a network of satellites that coordinate by exploiting the potential of their resources, with each satellite having all the infrastructure needed (i.e., not a fraction) to operate, and so being completely self-contained. Independent satellites are built and placed in orbit for specific objectives, allowing them to employ their resources and capabilities for an opportunistic distributed mission [54]. Federated and Fractionated satellites share some features, combining some of their capabilities and resources for a global mission [54-57]. Because the transferred resources are always underutilised in a module's primary mission, the nodes are complete and form heterogeneous systems, allowing for a new category of distributed mission to be categorized, as shown in Figure 11 [40].

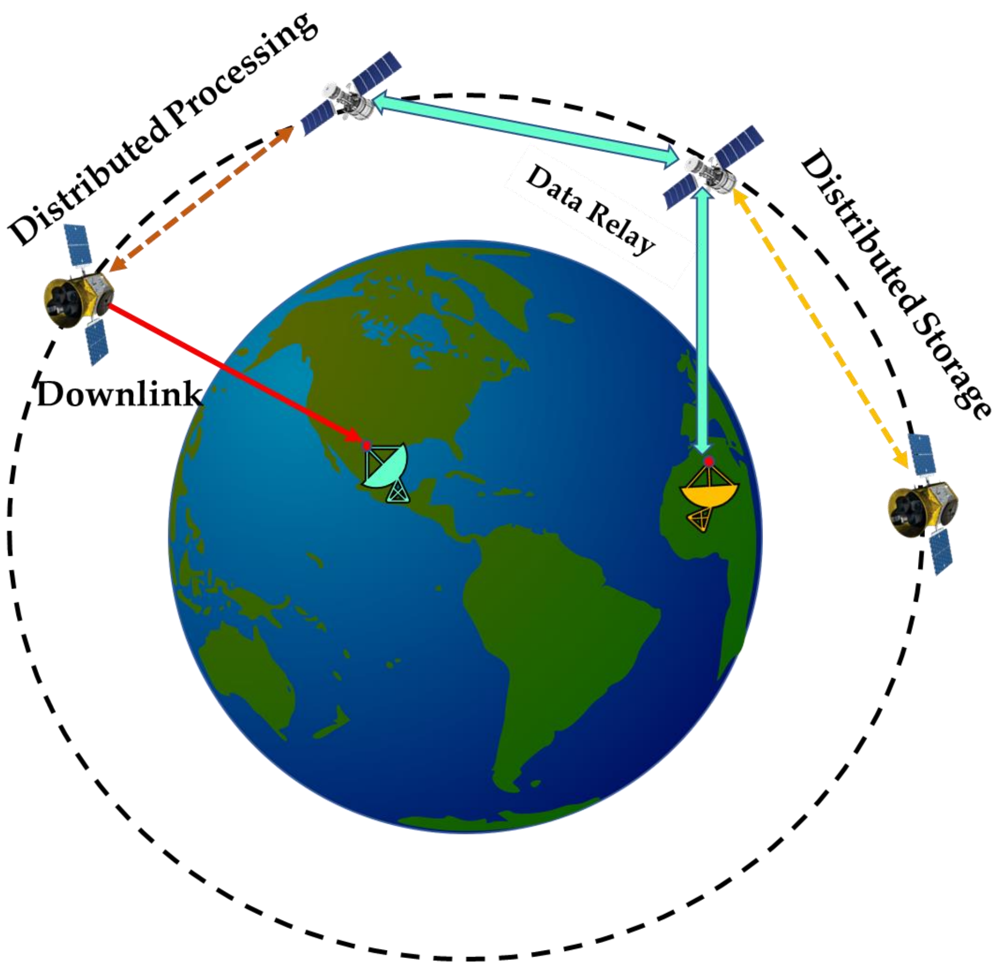


Figure 11: Federated satellite system.

2.3.4 Modular

Modular concepts are relatively new DSS classifications in which the satellites/modules are disintegrated, as shown in Figure 10 (d). Based on the CubeSat, Jiping et al. presented a new type of DSS with a reconfigurable construction and customizable function, dubbed Space Modular Self-Reconfigurable Satellite (SMSRS) design concept as shown in Figure 12, which shows the SMSRS configuration and the deployment from the folded state to work state. The following are some of the features of SMSRS: (1) Modularity, (2) Scalability, (3) Structural Reconfigurability, (4) Risk Resistance, and (5) Functional Adjustability are all critical characteristics [58].

i)

ii)

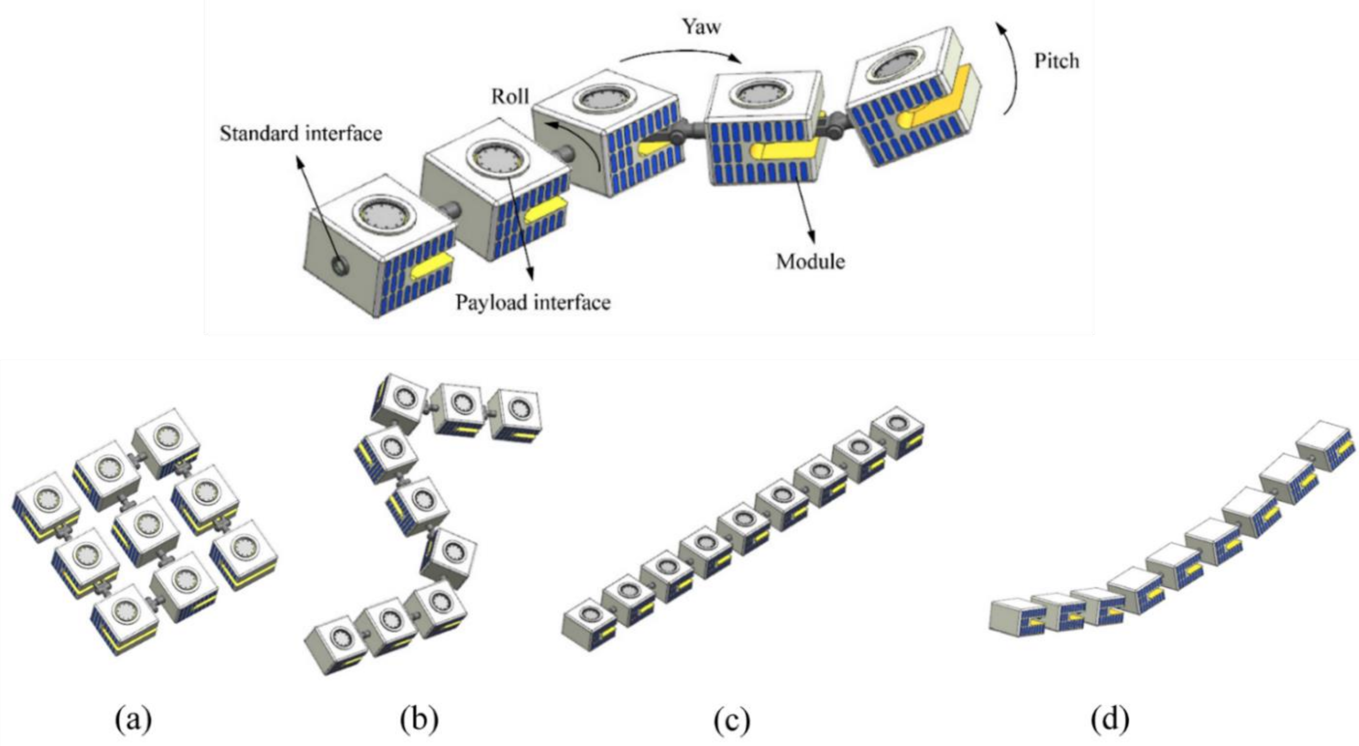


Figure 12: i) Model of SMSR, ii) Configurations of SMSRS (a) Folded state, (b) Unfolding, (c) Unfold state, (d) Work state. Adapted from [58].

Optical cameras, SAR, communication payloads, and other payloads are among the payloads carried by SMSRS. SMSRS, while carrying several payloads, arranges and reorganises these payloads in a variety of space orientations through structural reconfiguration, allowing it to carry out a variety of space missions. SMSRS is commonly used in the following scenarios [58]:

a) When SMSRS transports numerous optical cameras, joint motions may cause these cameras' spatial orientations to alter. By stitching together their field of view, these cameras might accomplish a larger imaging area, as shown in Figure 13 (a), or reconnaissance of numerous targets, as shown in Figure 13 (b).
b) As illustrated in Figure 13 (c), SMSRS can achieve multi-area communications to the ground by carrying several communication payloads and adjusting them to different orientations.

SMSRS application scenarios are not restricted to these, and there are broader expansions available. Simultaneously, a considerable amount of space debris is produced, endangering the survival of satellites in orbit. SMSRS's multi-functional feature, which allows individual satellites to execute many functions adaptively, can cut down on satellite launches. It eases the strain on space traffic management and minimises the amount of space debris produced [58].

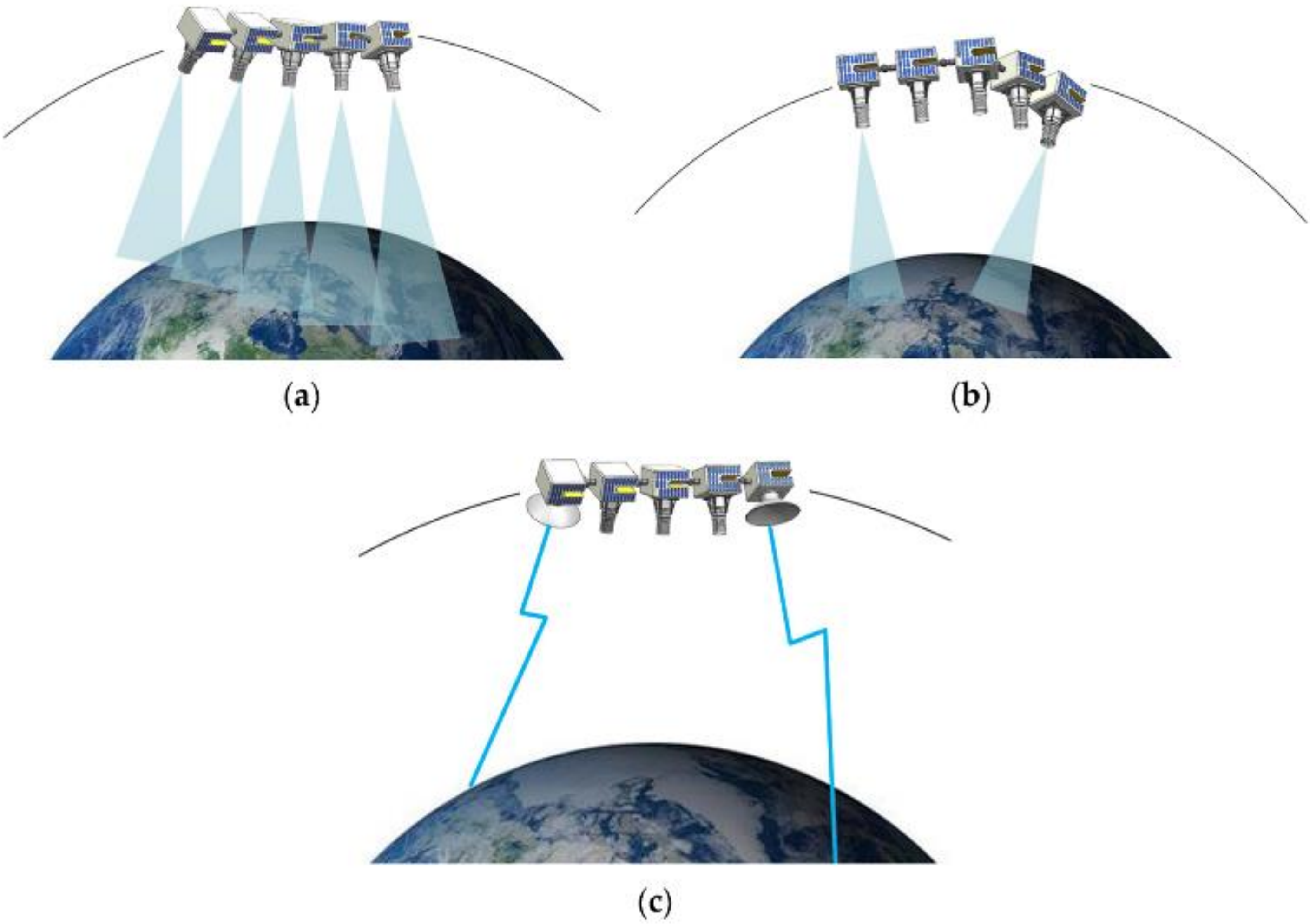


Figure 13: Application scenarios for SMSRS (a) SMSRS carries numerous optical cameras and stitches together fields of view. Multi-camera SMSRS surveillance of multiple targets (c) SMSRS provides multi-area communication to the earth [58].

2.3.5 Swarms

Swarm intelligence studies how natural (and artificial) multi-agent systems cooperate via decentralised control and self-organization, as shown in Figure 10 (e). Bloom [59] coined the term while researching complex adaptive systems, and it is made up of several principles (distributed parallel processing, superorganism, group selection, apoptosis). A typical swarm system has specific characteristics, such as a large number of homogenous agents (either identical or belonging to several typologies) that interact with one another via fundamental rules that exploit only local information. Information is exchanged either directly with another agent or indirectly through the environment. Stigmergy is the name given to this indirect coordinating mechanism [60]. The system's overall behaviour finally organises the group. This type of individual behaviour is commonly stated in probabilistic terms based on local neighbourhood perception. This ensures that the system can be scalable, parallelised, and fault resistant. It also includes consideration for any Swarm Intelligence system. It is distributed (executed by each agent in the system) and integrates randomization through each node's decision process. This is the reason, why the system is not stuck in "local compressed states" [61]. This allows a swarm divided into multiple isolated subgroups to have a single module eager to leave the group and keep the interaction process alive. In reality, swarms are very adaptable while also being extremely resilient (the system continues to work even if certain components fail) and completely decentralised and unsupervised. It works whether they are being used to describe natural or man-made agents. Satellite swarms are distributed missions in which the infrastructure modules are autonomous satellites conducting their own functions without the interchange or collaboration of resources (such as data). A distributed satellite of this kind comprises homogenous modules with the same or similar capabilities [62]. By increasing the number of modules dedicated to a certain task (i.e., adding redundancy), the set of constellation-conforming modules increases the system's usability, benefiting the system's robustness. For example, a deteriorating sensor in one of the modules of an EO mission does not prevent the operators from obtaining images. However, the amount of resources transferred (i.e. power, computational resources) is almost minimal in this situation. This type of distributed spacecraft can still communicate with one another to preserve flight formation or to relay critical trajectory information (e.g. to avoid collisions) [41, 62-64]. Nonetheless, their functions are limited to local, and their activities are done autonomously, as shown in Figure 9, without transferring any resources altogether [40, 41].

2.3.6 Formation

The coordination of multiple satellites to achieve the objective/goal is known as a formation. The flight of multiple objects in formation is known as formation flying, as shown in Figure 10 (c). In an effort to match the user's requirements, different configurations of formation flying missions are available. Small differences in the orbital parameters of the deputy satellite in relation to the nominal parameters of the chief satellite produce each configuration. Satellite formation flying has many architectures based on configuration, operation mode, and other parameters. References for more information can be found in the literature [65, 66]. In any case, the following sections cover the most critical aspects of the classification.

Cluster formation: A cluster configuration occurs when a set of satellites are organised in a close formation and positioned in orbits that keep them near together. Satellites in a cluster normally travel close together; however, this is not always the case in a trailing formation [52, 65].

Trailing formation: The satellites share the same orbit and follow one another with constant mean anomaly differences, keeping a predetermined relative angular separation from the Earth's centre. Notice that in terms of the mean anomaly the relative phase between satellites on a trailing formation is always constant, regardless of the eccentricity. The relative angular spacing in elliptic orbits, on the other hand, changes depending on the satellite's location. As a result, while the primary satellite is at perigee, these angles must be determined [65].

Leader-Follower formation: When describing two spacecraft, the term 'Leader-Follower' has the clearest meaning when one (the follower) is forced to fly in formation with the leader. In some works of literature, the term 'Leader-Follower' can also refer to a group of spacecrafts led by a single hierarchical leader. The following is the list of alternative terms for describing a leader-follower formation hierarchy. (i) Chief-Deputy, (ii) Master-Slave, (iii) Mother-Daughter, (iv) Primary-Secondary, (v) Hub/Combiner-Telescopes/Mirrors [65-68].

2.3.7 Constellation of Formations

A constellation of formations is a set of formations, where each formation has flight coordination between neighbouring satellites. In the constellation view, each formation can be described by the centre of mass of each formation, flying far away from each other but with a common mission goal, as shown in Figure 10 (f). Within the formation, there is typically relative navigation, guidance and control to acquire and reconfigure the relative orbits. On the other hand, the constellation objective is stated in terms of the desired orbit of certain selected satellites (sometimes called chief) or some weighted position average as the centre of mass of each formation in the constellation. Notice that each satellite must obey two objectives: to keep the formation relative geometry and to belong to the constellation [69].

2.3.8 Hybrid Missions

Hybrid Mission architecture categories are theoretical extremes, i.e., a mix of distributed systems generates this type, which is more complicated in most real circumstances and tends to be located in the centre of Figure 9. A fractionated satellite is a spacecraft unit capable of building a constellation with the other satellites (fractionated satellite swarm) or cooperating with other units in more heterogeneous and complex situations (federation of fractionated satellites). It is worth mentioning that mission designs can change quadrants in some situations, depending on the mission goals enforced by the ground segment [40, 41]. Hybrid mission objectives may alter because of a technical issue (unit maintenance, repair, replacement, research potential) or for commercial reasons (exploitation of modules, sporadic provision of services). Federated satellite systems with modules that can function individually or in formation in flight are excellent examples of this dynamism [40]. Table 1 provides a detailed description of different types of DSS architecture. The level of the operational independence of a satellite or a fraction of distributed spacecraft is characterised as *Operational/Functional Independence*. Individual spacecraft or fractions of a distributed spacecraft's *Homogeneity* is defined as the degree of similarity between them. [10, 11, 30, 41].

Jacqueline et al. [2] studied a variety of DSS attributes and classified them according to the taxonomy shown in Figure 14, and defined all of the concepts used in this taxonomy.

Table 1: Types of distributed mission architectures. Adapted from [30].

DSS architecture type		Mission goals	Cooperation	Homogeneity	Operational/ Functional Independence
Constellation		Mission goal shared (Iridium, GPS)	Cooperation is required to support the mission goals	In general, homogeneous components, some differences possible (GPS generations)	Autonomous
Formation	**Trains**	Mostly Independent, but could be shared	Cooperation from optional to required	Heterogeneous components	Autonomous
	Clusters	Mission goal shared	Cooperation is required to support mission goals	Homogeneous components	From autonomous to completely co-dependent
	Leader-Follower	Mission goal shared	Cooperation from optional to required	Heterogeneous components	From autonomous to completely co-dependent
Swarms		Mission goals shared	Cooperation required to support mission goals	From homogeneous to heterogeneous components	From autonomous to completely co-dependent
Fractionated		Shared mission goals	From optional (service areas) to required (distributed critical spacecraft functions)	Heterogeneous components	From autonomous to completely co-dependent
Federated		Independent mission goals	Ad-hoc, Optional	Heterogeneous components	Autonomous
Modular		Mission goal shared	Cooperation is required to support mission goals	From homogeneous to heterogeneous components	From autonomous to completely co-dependent
Hybrid		Mostly Independent, but could be shared	Ad-hoc, Optional	Heterogeneous components	From autonomous to completely co-dependent
Constellation of formations		Mostly shared but could be independent	Cooperation is required to support mission goals	From homogeneous to heterogeneous components	From autonomous to completely co-dependent

2.4 DSS System Architecture

A methodology for the bottom-up design of a distributed architecture is widely used, where elements of each layer are built up to reach the desired distributed architecture. The basic units arise from the bottom layer's objects and elements. At the top, there is a launch plan that shows which vehicle will launch each module [53]. DSS hardware and software architecture is discussed in the following sections.

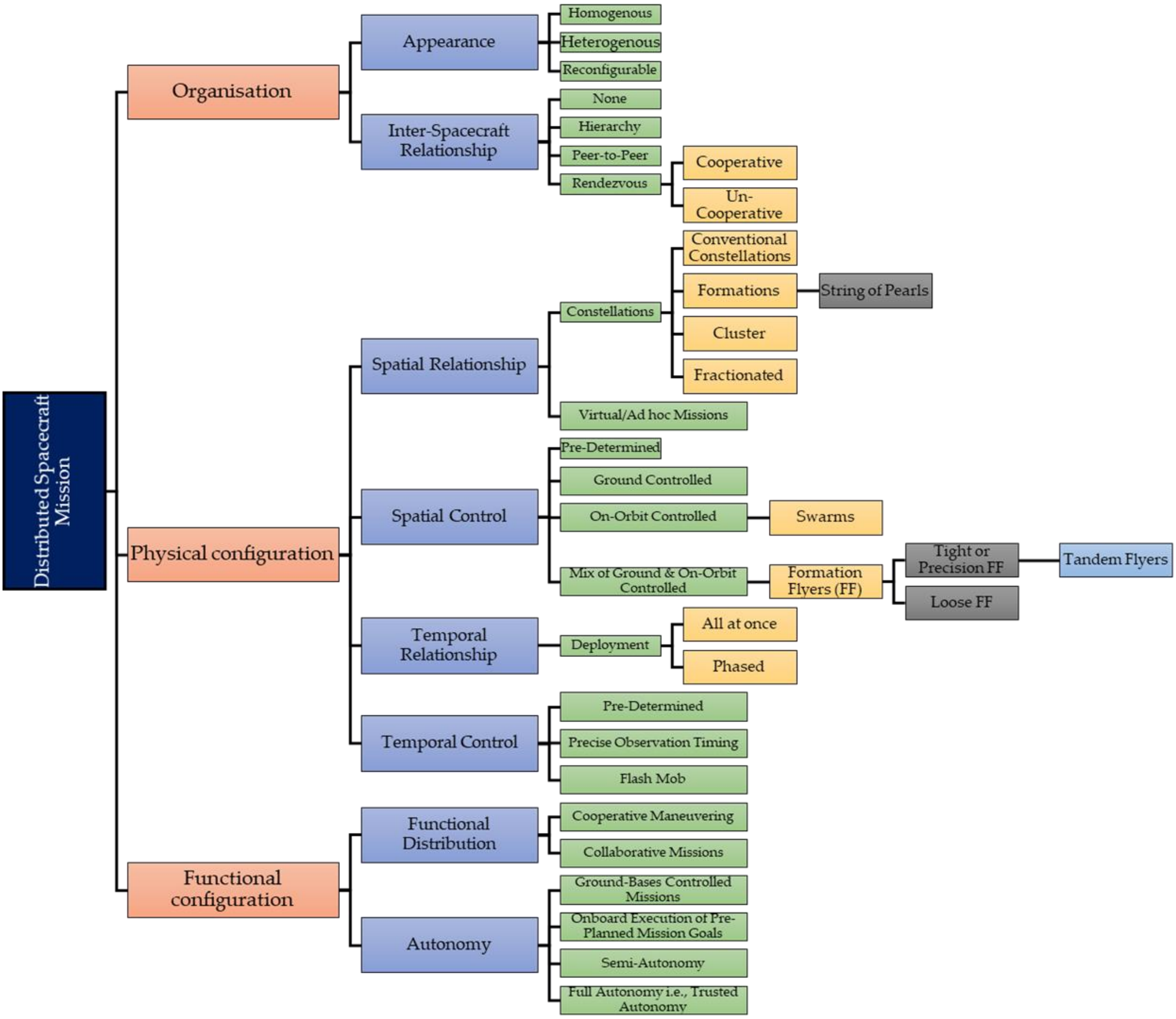


Figure 14: DSM Terminology Adapted from [2].

2.4.1 Hardware Architecture

Modules can be defined as constituents of a distributed system, such as payload modules, which include mission-specific instruments, functionalities, and infrastructure (or resource) modules that support the mission-specific payload modules [11, 26, 41]. DSS architecture provides a plug-and-play system due to the physical independence between the modules. Furthermore, it increases the value of the DSS. An example of this architecture is shown in Figure 15, with PR: Processor, DL: Downlink, PL: Payloads are fractionated, distributed and connected through ISL [28].

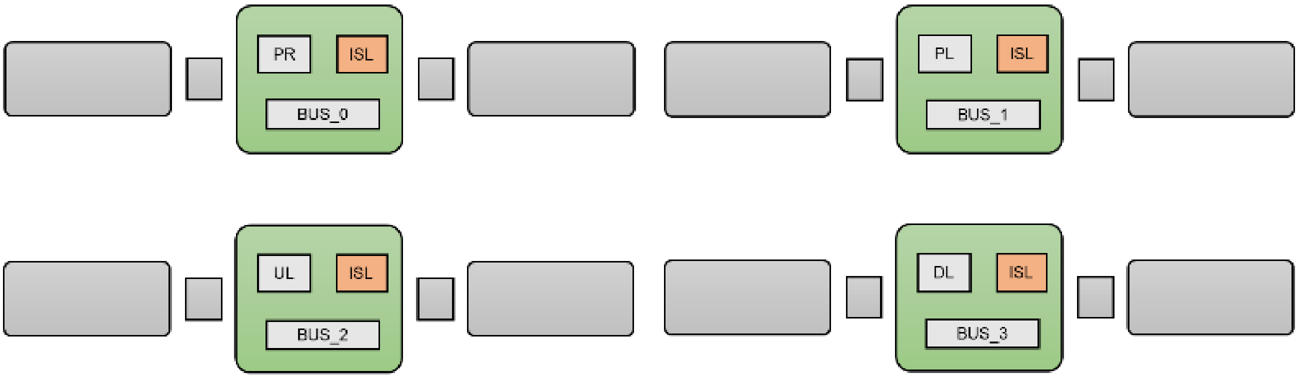


Figure 15: DSS architecture. Adapted from [28].

A DSS Hardware illustration is shown in Figure 16, which is a dove system consisting of a flock of satellites used mainly for EO operations. Cloud-based mission control is used for mission planning and scheduling in this system. Every payload plan and change in the task is updated via the network of ground stations operated by planet labs [15, 16].

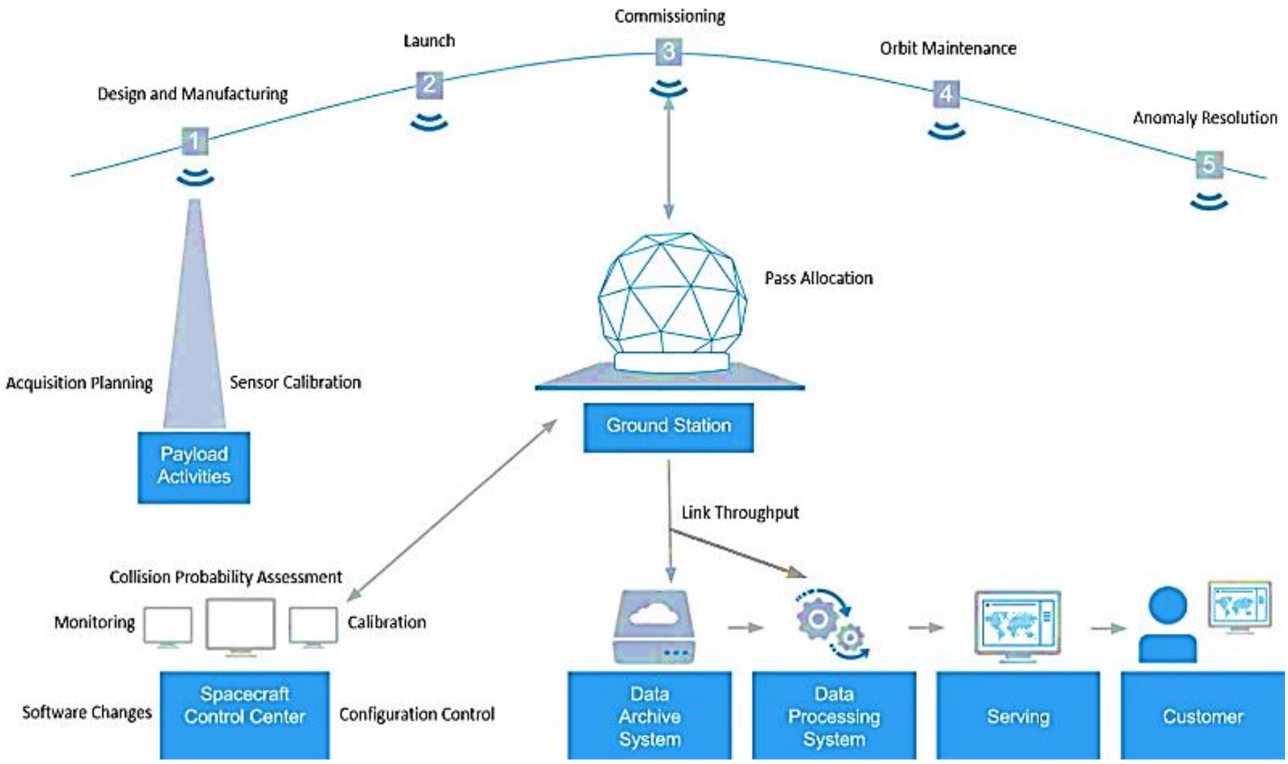


Figure 16: DSS Hardware architecture (Dove System) [15].

2.4.2 Software Architecture

Traditionally, spacecraft is controlled from a main on-board computer (possibly with redundant backups). Typically, modern spacecraft employs several dedicated on-board computing capabilities in charge of particular tasks and/or subsystems. Depending on the features of the mission, suitable software architectures are designed. Software architecture for DSS seeks to operate a system autonomously, where the components interact to:

i. Distribute tasks between modules /components of satellites,
ii. Allocate infrastructure resources,
iii. Perform task scheduling in a distributed manner as per requirements.

A structural view of software architecture is shown in Figure 17. The representative modules/components are not homogenous, indicating they have different payloads, computational capabilities, subsystems, and availability times, i.e., system encapsulation. The system is made up of autonomy management entities (i.e., task planners) that interact autonomously to operate a spacecraft. A transparent communication channel between global and local entities is provided by the Distributed System Layer (DSL) [40].

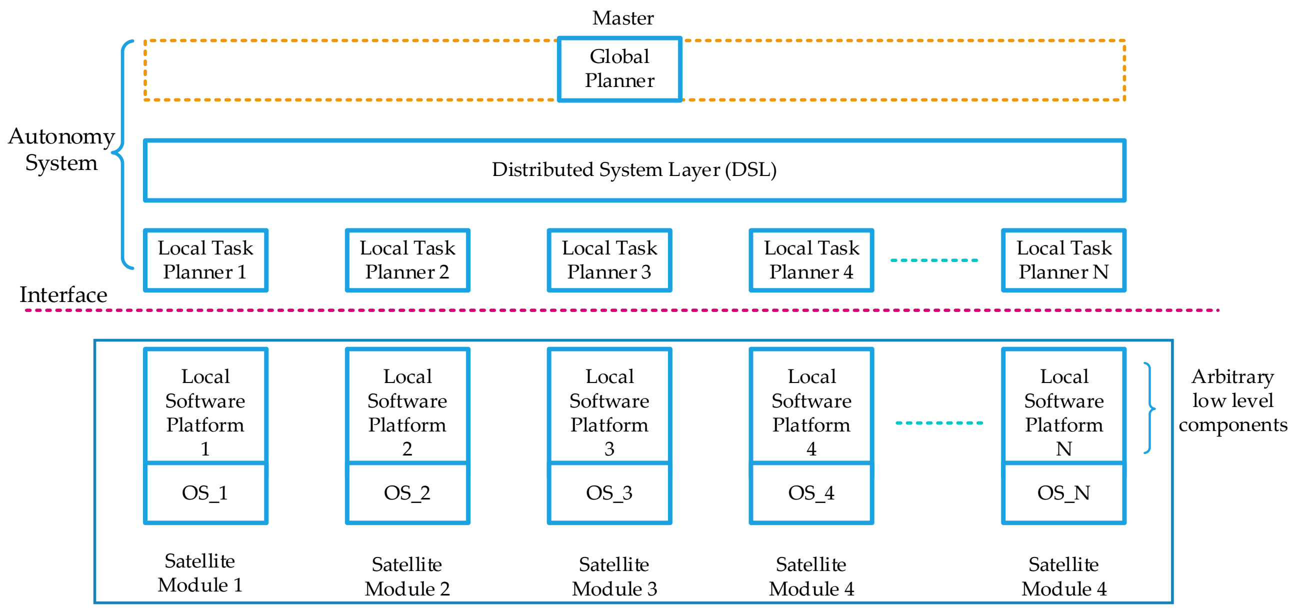


Figure 17: Distributed software architecture. Adapted from [40].

The entire architecture is controlled using two control levels, the Global control level, which is mainly relative to the software infrastructure domain, and the Local control level, which is relative to each module domain. The software architecture incorporates a master-slave hierarchical relation. In recent times, software architectures have been designed to suit changing environments. In addition, the structural description of the architecture also consists of dynamic management policies. The hierarchy and data encapsulation are detailed in the structure, while the policies define the system's functional behaviour. The system's functional view locally manages activities hidden from the autonomy system, which includes activities/tasks done by local software platforms—for example, satellite formation, functionalities/tasks extrinsic to the infrastructure, maintenance tasks, etc. The Global tasks are scheduled by the autonomy system, i.e., the main activities/tasks that are executed, a priori, by any module in the infrastructure. The "Policy" is the architecture's functional behaviour/model, which creates an interchange of information among the Global and Local control levels. It provides a method to execute distributed assignments of global objectives for each module and period consisting of a compendium of algorithms. Considering the distributed autonomous software architecture within a dynamic context, dynamic management policies are adopted, bringing about changes in the mission.

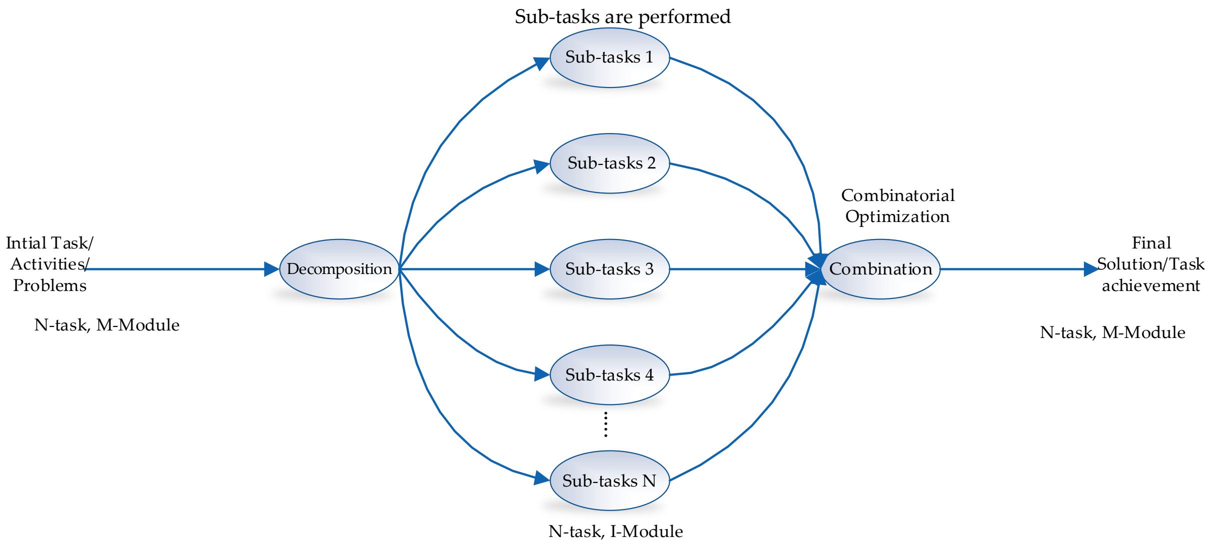


Figure 18: Task execution in a distributed system. Adapted from [40].

The Local-Global approach of software architecture is a mixed management policy. This is intended to provide an arbitrary number of heterogeneous modules to an adaptive planning solution for an autonomous distributed spacecraft (i.e., payloads, computational capabilities, different platforms, hardware, ISL bandwidth, etc.). It had grown accustomed to the software infrastructure modules [40]. This balances the master module processed information. By decomposition, the "multiple-tasks multiple-modules” task/problem is changed into “multiple-tasks single-module”, as depicted in Figure 18.

2.5 Conclusion

Space systems mostly encounter a great deal of uncertainties in the space environment. This makes their design specifications difficult and, in many instances, intractable since a huge number of feasible design options must be evaluated against a multitude of uncertain situations. Several sources of uncertainty occur for space systems in both space environment and ground-based setup, including technological developments, fluctuations in demand, failure to launch, availability of funding, and changes in stakeholder requirements. This results in increased cost and complexity, especially for conventional monolithic designs. Alternative designs for these systems should be considered to reduce both space and ground-based uncertainty. This chapter presents a unified classification of DSS and discusses the various categories that make up this categorisation. The modularity spectrum mentioned earlier should be used to evaluate these designs. It is apparent that distributed systems will play an essential role in the modern space era. As a result, appropriate strategies and use cases for exploring and exploiting DSS architectures should be outlined.

Chapter 3

AI for Autonomous Operation in Space

In a system's context, autonomy can be defined as the ability to make informed, reasonable, self-reliant, and self-determined decisions. A system should be able to sense, think, and act inside its surrounding environment in order to be deemed autonomous. It necessitates the capacity to detect its surroundings as well as some awareness of one's own powers and how they affect one's environment and internal condition. An autonomous system makes inferences and conclusions about its own goals and takes action to achieve them [70]. Additionally, an autonomous system must react to non-nominal conditions by adjusting its system of operations to fulfil its goal while being secure. The degree of autonomy a system achieves is determined by the degree of off-nominality it can handle and the level of abstraction of its objectives [71].

A closed-loop ("sense-think-act") system, as shown in Figure 19, describes an autonomous (machine) device or function for different layered architectures as follows.

- Sensors ("sense") provide the computer with knowledge about its surroundings, i.e., data.
- Control software is used to process the data ("think").
- Conduct an operation ("act") without further human interference based on its analysis.

As a result, autonomy is described as a system's ability to function without direct human interference, though it is a spectrum with several levels and grey areas. Some autonomous systems in aerospace carry out predetermined acts that do not alter in response to the environment (automatic). Other systems (automated) initiate or modify their behaviour or output in response to environmental feedback, while more advanced systems (autonomous) combine environmental feedback with the system's own interpretation of its current situation. Increased autonomy is often viewed as increased "intelligence" or even "artificial intelligence" for a specific mission, and it is usually equated with greater adaptation to the environment. Over the years, it has been agreed that architectures for autonomy should be included in a planning layer, a task sequencing layer, and a reactive layer. Deliberative, executive, and functional layers are all terms used to describe these layers. These layers are defined by their separation from the equipment and the time it takes for them to respond. Response-time constraints usually imply limitations on the capability to deliberate and the time horizon that can be considered. The functional layer has fast turnaround requirements since this must maintain pace with the hardware, and each component normally only considers one task or series of tasks. The executive layer manages a collection of tasks at the moment, and it only needs to reply fast enough to meet task action potentials and terminations. Finally, the deliberative layer considers numerous tasks, multiple possibilities, and future repercussions. It just needs to reply quickly enough to offer the executive extra job sets or plans when necessary. On the other hand, the layers do not just indicate a boost in capabilities; trade-offs do exist. The functional layer has access to detailed data about the hardware and frequently performs complex numerical calculations to decide responses or provide data to the layers above. The executive layer usually includes contingency management and control skills that the deliberative layer lacks. Each layer executes a variation of the sense-think-act cycle in an autonomous system. The overarching system of autonomy has a sense-act-think cycle as well. Sensing entails gathering data from lower levels or hardware and transferring it to a representation that the software can understand. Thinking entails weighing sensory data, spacecraft information and desired outcomes before deciding what should be done. Finally, acting entails putting the decisions made throughout the cognitive cycle into action [72]. The European Cooperation for Space Standardization (ECSS) has defined four degrees of autonomous capability, with level E4 being the most autonomous. Only level E4 compatible technologies can be deemed genuinely autonomous, according to the criteria in Table 2, whereas levels E1 to E3 relate to human-operated or automated systems. Unlike autonomous systems, automated systems can only deal with their designers' predicted scenarios. It will react to these conditions using so-called on-board control procedures, which are pre-programmed sequences of operations (i.e., events). The different levels of autonomy in mission execution are shown in Table 2. In order to operate the entire mission autonomously, there is a need for autonomy in mission data management and mission fault management. ECSS defines these capability levels as in Table 3 and Table 4 [73].

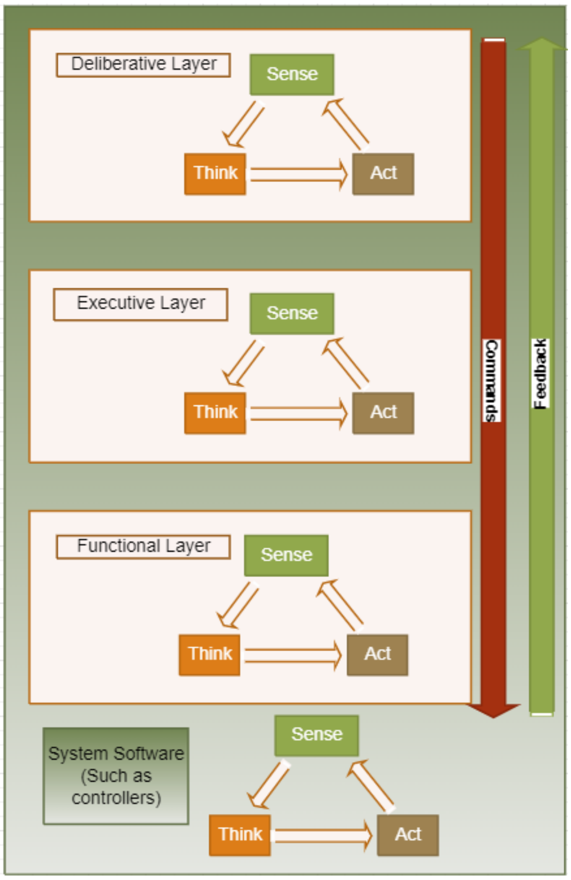


Figure 19: The Sense-Think-Act cycle Autonomous systems layered architecture [72].

Table 2: Levels of autonomy for mission execution as stated by the ECSS [73].

Level	Description	Functions
E1	Mission execution underground control with limited on-board capability for safety issues	Real-time control from the ground for nominal operations Execution of time-tagged commands for safety issues
E2	Execution of pre-planned, ground defined, mission operations on-board	Capability to store time-based commands in an on-board scheduler
E3	Execution of adaptive mission operations on-board	Event-based autonomous operations Execution of on-board operations control procedures
E4	Execution of goal-oriented mission operations on-board	Goal-oriented mission re-planning

Table 3: Levels of autonomy for Mission fault management as stated by the ECSS [73].

Level	Description	Functions
F1	Establish a safe space segment configuration following an on-board failure	Identify anomalies and report to ground segment Reconfigure on-board systems to isolate failed equipment or functions Place space segment in a safe state
F2	Re-establish nominal mission operations following an on-board failure	As F1, plus reconfigure to a nominal operation configuration Resume execution of nominal operations Resume generation of mission products

Table 4: Levels of autonomy for mission data management stated by the ECSS [73].

Level	Description	Functions
D1	Storage on-board of essential mission data following a ground outage or a failure situation	Storage and retrieval of event reports Storage management
D2	Storage on-board of all mission data, i.e., the space segment, is independent from the availability of the ground segment	As D1 plus storage and retrieval of all mission data

Satellite operations can occur in Earth orbits or deep space missions, such as planetary exploration. Both activities require the use of robotics and autonomy. Most spacecraft operations' control functions and procedures are transmitted for immediate execution by telecommand or, more commonly, in precisely organised sequences at specified times. Almost all remote sensing satellites, for example, gather images and downlink to Earth at predetermined geographic areas while retaining the correct attitude using on-board sensors and reaction wheels. On the other hand, Astronauts operate robotic systems in space, such as, the Canadian arm Remote Manipulator. Few autonomous aerospace systems make decisions without human intervention in order to attain high-level objectives. Freed et al. [74] offer a Verification and Validation (V&V) methodology for autonomous space systems that aims to increase trust in the stability of complicated software. This combines runtime analyses and model control using software design architectures to enable traceable modular verification activities and automated code generation while delivering automatic formal V&V verification tasks. Freed's intelligent automation system guarantees that software is conceived, produced and verified by domain experts-engineers and scientists for space activities, which is another crucial part of creating confidence in autonomous software [74].

As defined by Proud et al. [75] and Novaes [76], variable autonomy develops the concept of selecting desirable levels of autonomy while constructing a space system. This allows the autonomous system or the human user to alter the level of autonomy as needed by the situation. Autonomous space systems provide excellent sensing and are therefore necessary if human usage and exploration of space are to expand in terms of both reach and complexity. Trusted autonomous spacecraft systems will allow such activity to continue with confidence. For crucial space systems, several scenarios can be predicted. Some of these are already in the development and demonstration stages. For example, autonomous on-board data processing, on-orbit satellite servicing/repairing, analysis and decision-making for remote sensing for both defence and civil applications, as well as future human space habitation, which could include both space tourism and deep space colonisation, are all plausible scenarios [71, 77-79]. Figure 20 shows an evolutionary roadmap of space system capabilities with a growing degree of autonomy over time. The four categories are defined as follows, 1. Teleoperation (operated from Earth), 2. Automatic Operation (pre-programmed self-control), 3. Semi-autonomous Operation (start with a predefined command sequence, where the machine adapts to the external environment), and 4. Fully Autonomous Operation (autonomous decision-making (goal-oriented)) [80]. Autonomy can be incorporated into various segments of the satellite infrastructure. With recent trends, TASO for space applications is becoming more popular. AI applications in the control and space segments can potentially increase the value of both ground and space operations.

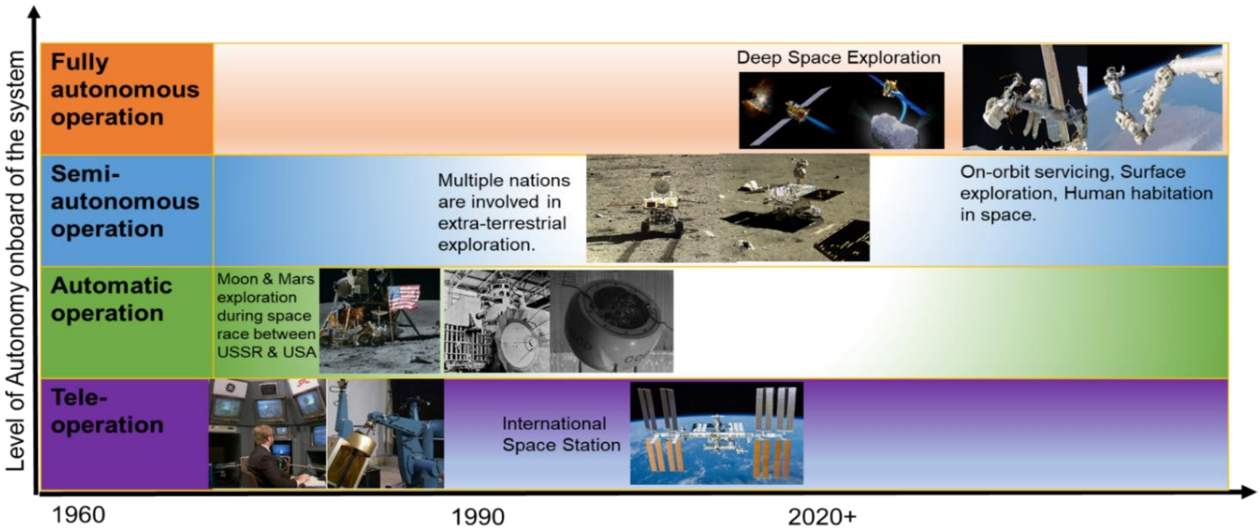


Figure 20: Evolution of Autonomy in space systems. Adapted from [80].

3.1 Human-Machine Autonomy

The autonomy of a machine is significant because it influences the number of tasks it can complete because of the increasing demand and regularity of human-machine interaction. There are several levels of autonomy, varying from teleoperation to complete autonomy. Beer [81] proposes a structure for categorising levels of autonomy and guidelines for choosing and maximising the appropriate level of human-machine interaction centred on the machine's intended purpose. Human interaction is required at all levels except the final stage of autonomy. The categorisation is shown in Table 5.

Table 5: Levels of Automation. Adapted from [81].

Automation Level	Level Name	Description
1	Manual	Human performs all mission aspects
2	Tele-operation	Machine assists in task execution
3	Assisted Tele-operation	Machine assists in sensing and task execution and intervenes when needed
4	Advisory execution	Human formulates mission, and machine executes the task
5	Skilled Execution	Machine assists in sensing and planning and executes the task
6	Shared Control with Human Initiative	Autonomous Machine operations with human oversight
7	Shared Control with Machine Initiative	Autonomous Machine operations with human assistance
8	Supervisory Control	Autonomous Machine operations with a human directive
9	Executive Control	Autonomous Machine operations with human override
10	Full Autonomy, i.e., Trusted Autonomy	Autonomous Machine operations without human intervention

3.2 Cognitive Human-Machine Systems

A human-machine system incorporates the functions of an individual human operator (or group of operators) and a machine as an interface. This can also be viewed as a system of a single entity interacting with the external environment. An autonomous system or function is, by definition, out of human control to some extent. Humans can, however, exert some control during the design and development stage at the point of task initiation and during service, for example, by interrupting the system's operation. The need for human supervision or the degree of autonomy that can be tolerated is related to the complexity of the environment wherein the system operates as well as the complexity of the role it executes. There is no widespread model for ideal human-machine interaction with autonomous systems. In general, the higher the complexity, the greater the need for direct human control and the lower the tolerance for autonomy, particularly for safety-critical tasks and environments in which a system failure may injure or kill people or cause property damage. When an autonomous system is used in an unpredictable, uncontrolled environment, there is a high risk of failure with unforeseen consequences. Nonetheless, recent technological advancements in complex control software, such as AI techniques, aim to increase the degree of autonomy, tolerating more complex tasks in complex environments. Humans can control machine systems in a variety of ways:

1. Direct control: Requires continuous interaction by a human operator to control the system's functions directly or indirectly, making it non-autonomous.

2. Shared control: The human operator controls specific tasks directly, while the computer controls others under the operator's supervision. The aim of shared control is to:
   a. Combine human control's advantages (global situational awareness and decision) and computer control (high-speed, high-accuracy actions).
   b. Partially overcome human control limitations (attention period and field of vision limitations, tension, and fatigue) and machine control limitations (limited situational awareness and decision-making capacity, sensing uncertainties).
3. Supervisory control: A device executes tasks autonomously while a human operator supervises and provides guidance, intervenes, and reclaims control as required.

Supervisory control is often used in civilian applications because direct or shared control of a machine system is not feasible due to communication delays between the operator's commands and the system's corresponding operation, such as in systems working in outer space or deep-sea areas.

3.2.1 Human-on-The-Loop

Predictive control is challenging in most real-world environments because the operating environment is complex, unpredictable, and dynamic in nature. On the other hand, human supervisory control allows operators to exert some control through "human-on-the-loop" monitoring and intervention. There may be several loops in which the operator may intervene, each with different outcomes, such as a low-level control loop for particular roles (control level) and/or a high-level control loop for broader objectives (planning level). In any case, successful human-on-the-loop monitoring and intervention necessitate constant situational awareness as well as sufficient time to intervene (i.e., override, deactivate, or take other actions) and a way to interfere, such as a permanent contact connection (for remotely operating systems) and/or direct physical controls that allow the user to regain control or deactivate the machine. Unfortunately, even though the human-on-the-loop model meets all the above requirements, it is not a silver bullet for maintaining successful control over autonomous systems due to well-known human-machine interaction issues. These include:

1. Over-trust in the system, or automation bias, occurs when humans put too much faith in the operation of an autonomous machine.
2. A lack of external environmental awareness on the part of the operator (insufficient knowledge of the state of the system at the time of intervention, as explained below)
3. The ethical buffer, in which the human operator delegates moral obligation and accountability to the system, is viewed as a valid authority.

3.2.2 Cognitive Human-Machine Interfaces and Interactions

Cognitive Human-Machine Interfaces and Interactions (CHMI^2) is a new method to human factors engineering in aerospace that incorporates adaptive functionalities into the design of the operator's command, control, and display capabilities [82, 83]. A CHMI^2 system evaluates human cognitive states built on critical psycho-physiological observables being measured [83]. The cognitive states have been utilized to estimate and improve the operator's performance in the accomplishment of tasks to improve the efficiency and effectiveness of the overall human-machine teaming. Moreover, the result in the literature [83] indicates that higher levels of CHMI^2 supported automation are beneficial for space applications.

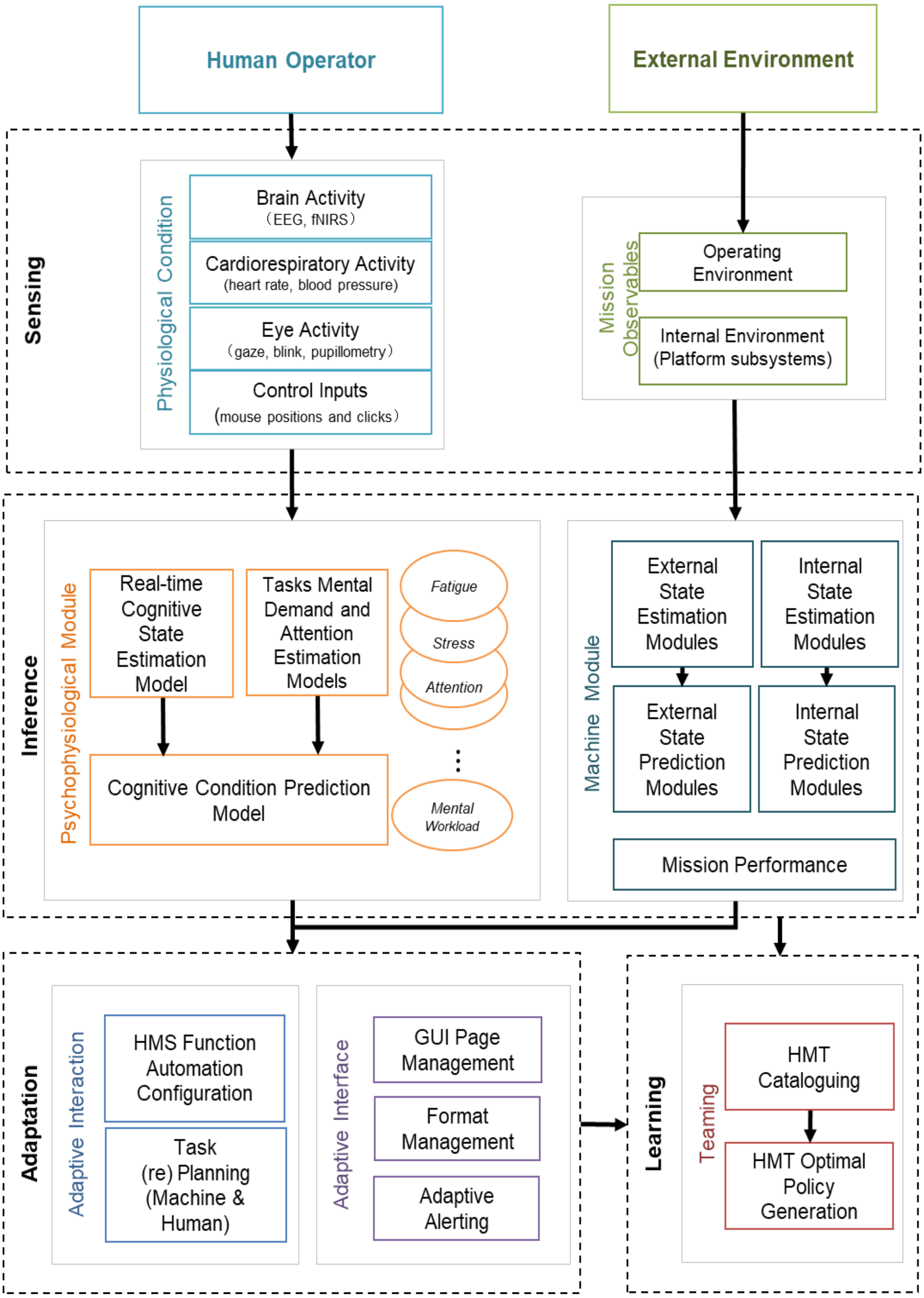


Figure 21: CHMI² Framework [13].

This shows that the presentation of CHMI² functionalities in potential space applications can considerably decrease response time, improve the operational effectiveness of spacecraft operation, and improve the overall protection and effectiveness of operations [83-85]. CHMI² supports human-machine teaming, whereby a system senses and adapts to the mission environment and the cognitive state of the operator. The CHMI² concept enables TASO in mission-critical and safety-critical applications [13]. The definition of $CHMI^2$ builds on and capitalises on significant developments in aerospace avionics human factors [86, 87], which are detailed in Refs. [86, 87]. The $CHMI^2$ framework's primary feature is an expansion of a device monitoring approach that assesses a Human-Machine System's (HMS) entire integrity by including both cognitive (human) and automated (machine) components. It is planned to characterise the operator's actions that resulted in a particular mission outcome by detecting specific features that can deduce cognitive states (for better or worse). This closed-loop input helps to improve HMS's trustworthiness in essential areas like the initial design process. It supports cognitive system engineering activities, such as the creation of system automation methodology based on operator policies and online adaption of the HMS based on the cognitive state of the person and the operating environment, during the early design process. The $CHMI^2$ system is depicted in Figure 21, and the reader is referred to Ref. [85] for further details. Before World War II (WWII), the human-machine connection was based on "humans adapting to machines, "whereas after WWII, it was based on "machines adapting to humans". It has progressed into $CHMI^2$, which is the communication between humans and non-AI computing systems since the dawn of the computer era. Computing products (such as automated machines) are typically used as a tool to aid humans in monitoring and executing tasks. The evolution of $CHMI^2$ over time is shown in Figure 22. Similar to what has happened in the computer era, AI technology has enabled a new sort of $CHMI^2$ collaborative interaction that would eventually lead to a paradigmatic shift in $CHMI^2$ application areas in the AI era, resulting in new design thinking and approaches to AI system development. Figure 23 expands on the approach, and the reader is referred to [88] for further details.

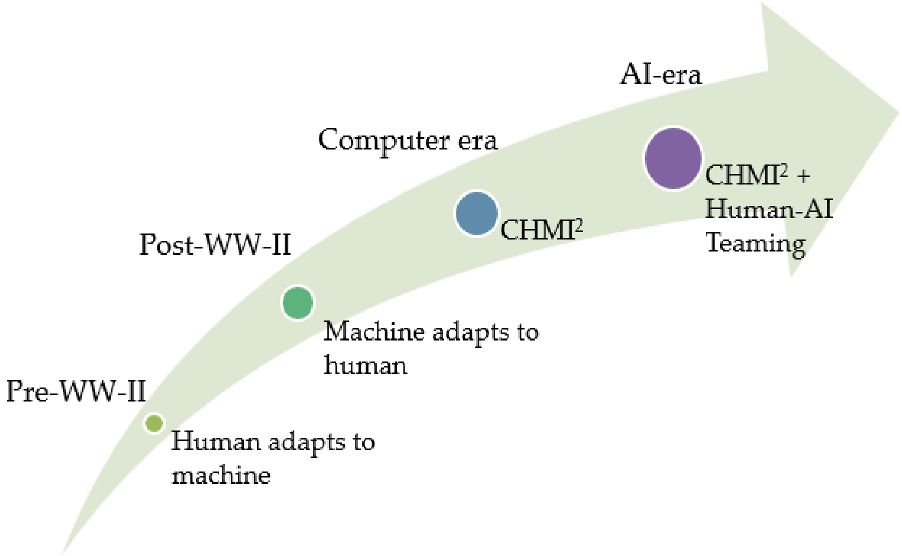


Figure 22: The development of the human-machine relationship across time. Adapted from [88].

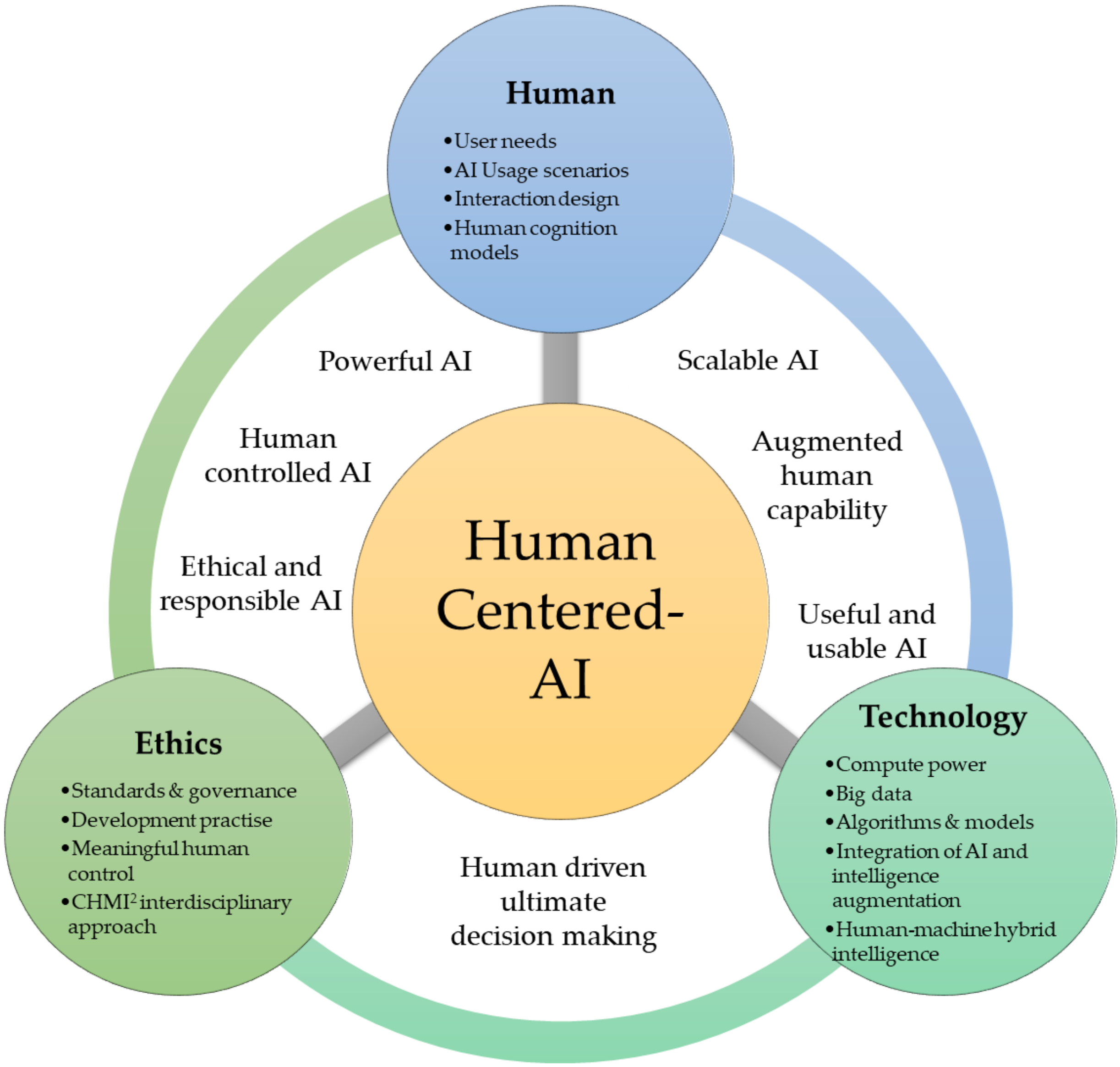


Figure 23: Framework for Human-Centered AI [88].

3.3 Artificial Intelligence for Space Operations

The term "intelligent space system" refers to space systems that operate independently using intelligent methods. To achieve autonomy, AI approaches are used without the need of human interaction in this system. AI may examine previous work to make sure everything is completed correctly. Furthermore, including collaborative robots ("cobots") into the production process decreases the requirement for human workers in clean rooms. It improves the consistency of production processes that are prone to errors. AI, unlike humans, does not require rest or sleep in order to digest large amounts of data quickly. The basic objective of the techniques utilised can also be used to classify AI [70, 79, 89]. The following are the four layers of autonomous systems,

1. A foundational layer that covers traditional methodologies like statistics and econometrics, as well as complexity theory and game theory.
2. A behavioural layer that comprises operational procedures including automated processes, machine translation, and collaborative and adaptive systems, among others.
3. A sensory layer that provides language, audio, and visual information to the model.
4. The "intelligence" is provided by a cognitive layer incorporating ML, reasoning, and information representation.

These definitions are helpful in thinking about the purpose of the strategies being used. A combination of these would be used for the most advanced AI systems. AI would significantly affect human and robotic space exploration missions in several different ways. As time progresses, AI will complement space exploration activities in a variety of ways, as seen in Figure 24.

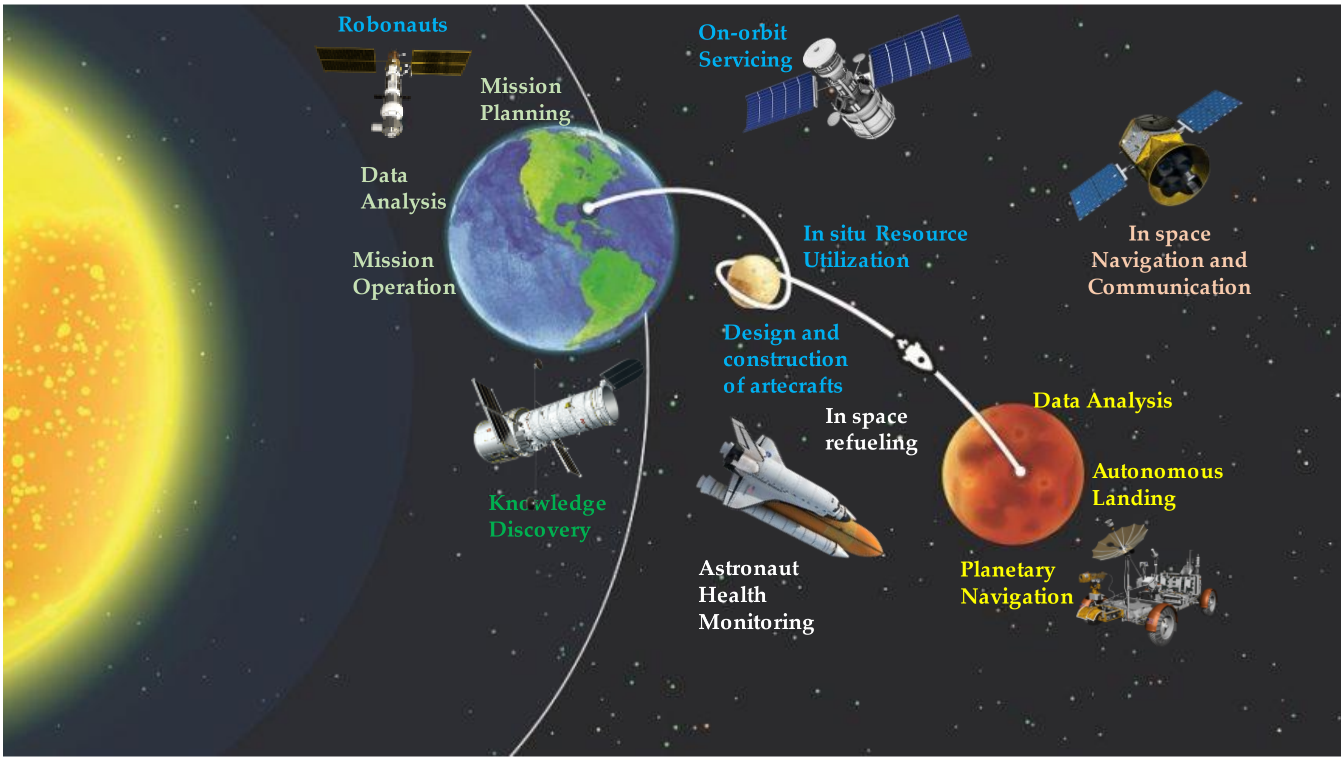


Figure 24: AI-augmented space exploration.

Table 6: Summary of AI techniques in space.

Spacecraft Operation	Techniques	References
Remote sensing	On-board data processing Image processing for precision agriculture and agroindustry Hyperspectral image classification	[18, 90-105]
Communication	Satellite communication Inter satellite communication	[106-111]
Automated control and navigation	Satellite attitude control Autonomous proximity operations and docking of spacecraft. AI-based control systems	[112-120]
Satellite Health Monitoring	Automatic anomaly detection techniques Intelligent health monitoring systems Spacecraft structural health monitoring	[121-127]
Deep space and Multi-Planetary Exploration	Exoplanet detection Interplanetary trajectory design Deep space communication	[106, 128-134]
Satellite Mission Planning	Autonomous planning and scheduling Trajectory optimisation of the space launch vehicle Spacecraft trajectory optimisation	[29, 120, 135-141]
Space Traffic Management	Collision avoidance Separation assurance Space Based Space Surveillance Space domain awareness	[23, 142-149]

Table 6 gives the summary of AI approaches in spacecraft operation. Some of the main applications of intelligent systems in the near-earth region and multi-planetary exploration in outer space are,

- AI for remote sensing data analytics.
- Satellite Trajectory Planning and Collision Avoidance using AI.
- Satellite Health Monitoring using AI.
- Satellite Communication using AI.
- AI for Deep Space and Multi-Planetary Exploration

3.3.1 AI for Remote Sensing Data Analytics

Every minute of the day, satellites produce thousands, if not millions, of imagery and several gigabytes of data daily. Weather and ambient pictures and photographs down to inches of the globe are all captured in these images. The autonomous capturing of Earth's photos poses a number of issues and possibilities where AI can assist. Without AI, humans are primarily responsible for interpreting, comprehending, and analysing imagery [89]. By the time a human arrives around to evaluate an image, the satellite may have moved back to the same place, requiring more refinement of the image analysis. AI-enabled recognition gives the researcher much power when it comes to image analysis and reviewing the images produced by satellites. On the other extreme, AI can analyse images as they can be captured and identify whether they have any abnormalities [94-96]. The use of AI to collect Earth's images also eliminates the need for a lot of communication to and from Earth to analyse images and decide whether or not a new one should be acquired. AI saves computing power by minimising communication, lowering battery use, and accelerating the image collection process [95].

3.3.2 Satellite Mission Planning and Collision Avoidance Using AI

Satellites are complex pieces of infrastructure that must be operated in order to function. Many issues could develop, ranging from equipment malfunctions to crashes with other satellites/debris. AI is used to maintain track of satellite health and ensure it continues functioning properly. AI can keep track of sensors and equipment in real-time, sending out alarms and, in some situations, taking corrective action. SpaceX, for example, utilises AI inside its systems to keep its satellites avoid colliding with other space objects [45, 150]. AI is also utilised to control satellite navigation and other spacecraft. AI can analyse the patterns of many other spacecraft, planets, and space junk. Once AI has discovered the patterns, the spacecraft's path can be changed to avoid collisions [121, 123, 125, 126, 151, 152].

3.3.3 Satellite Health Monitoring Using AI

Satellites have intricate pieces of equipment in their subsystems that are required for operation. Malfunctions in this equipment have the potential to lead to several on-orbit failures, such as attitude control malfunctions, battery and solar-array failures [153]. These failures cost the satellite industry billions of dollars. To ensure reliable and safe operations, AI can monitor the health of all satellite subsystems. Satellite operations involve human fault identification during routine inspections using on-board logbooks and minimal surveillance. This arrangement is insufficient for sophisticated missions incorporating intelligent satellite systems like DSS. As a result, the satellite system's reactivity and functionality, particularly in fault detection, are substantially improved. Concerning the other essential subsystems, an autonomous AI-based satellite health monitoring and management system might be entrusted with monitoring and predicting their health. Automatic monitoring of all satellite subsystems eliminates the need for human inspection, and any detected defect or imminent malfunction promptly alerts the ground station, redistributing satellite system resources to mitigate its impact. After the warnings, an operator can intervene and take control. Offline analytical techniques could be used to obtain further information and resolve the detected issue. So that only non-nominal situations necessitate operator intervention, the "human-on-the-loop" concept is promoted. Increased on-board autonomy would allow for more complicated satellite application missions and reduce human operator workload [154].

3.3.4 Satellite Communication Using AI

In addition to keeping spacecraft operational, it can be challenging to communicate between Earth and space. Interference with other signals and the environment depends on the state of the atmosphere. A satellite may have a lot of communication difficulties to overcome as a result of uncertainties in the environment. AI is now being utilised to control satellite communication in order to circumvent any transmission issues. These AI-enabled technologies can determine how much power and frequencies are needed to send data back to Earth or other satellites. The satellite does this regularly with an on-board AI to allow signals to pass through as it travels through space [155-159]. Beam-hopping, anti-jamming, detecting ionospheric scintillation, network traffic forecasting, channel modelling, telemetry mining, interference management, remote sensing, behaviour modelling, space air-ground integrating, and energy management are just a few of the applications where AI has shown promise. AI should be used to produce more effective, reliable, consistent, and high-quality communication systems in the future [109].

3.3.5 AI for Deep Space and Multi-Planetary Exploration

Even satellites on other planets or in interplanetary space, like the Curiosity rover currently on the red planet, use AI to operate. The Mars rover is using AI to assist it in navigating the planet. The computer may make several modifications to the rover's trajectory every minute. The Mars rover's technology is quite similar to that used by self-driving automobiles. The key difference is that the rover should cross more challenging terrain without having to worry about other vehicles or pedestrians. The rover's computer vision systems analyse the difficult terrain as it traverses. If an issue with the terrain is detected, the autonomous system adjusts the rover's navigation or modifies its trajectory to avoid it [77, 129-131, 160].

3.4 AI Techniques

In contrast to natural intelligence, AI is the study of intelligence as manifested in computer systems and observed in people and other lifeforms [161]. To be considered intelligent, a computer system must be capable of making reasonable judgments based on experiences and observations of the world (or a simplified model of it) and a set of objectives to be met. In space and satellite technology, AI holds a lot of promise. Spacecraft systems are complex and costly pieces of technology to assemble. There are repetitive and complex activities inside the manufacturing facilities of spacecraft that must be carried out with extreme precision and typically in clean rooms with limited exposure to potential contamination. Robotics and AI-enabled technologies are utilised to help with the production process and take over some of the activities humans now do to lessen their workload. By applying suitable AI techniques, satellite systems can make real-time decisions without explicit instructions. A plethora of research coupled with many tests is underway to implement AI-based technology in space systems, with various projects beings carried out [77, 79]. Some of the most commonly used AI methods are shown in Figure 24.

3.4.1 Based on Task Achievement

AI can be classified into two distinct types, strong AI and weak AI, based on the given task. Strong or general AI is concerned with the replication or outperformance of human brains, including sensitivity, consciousness, mind, and feelings. On the other hand, weak or applied AI focuses on completing a single task or resolving a specific problem. Because most research in the space domain is limited to weak AI, this research focuses only on applied AI.

3.4.2 Metaheuristics

Most conventional optimisation methodologies use a deterministic rule to switch from a single point in the decision hyperspace to another. The main disadvantage of this method is the possibility of converging on a local optimum. Since stochastic algorithms are designed to find the best global solution to problems with multiple local minima (usually nonconvex problems), they typically overcome this issue. There are two kinds of stochastic algorithms, namely heuristic and metaheuristic, though the distinction is minor. Stochastic optimisation is sometimes the second-best way to get a solution. Conventional techniques such as linear programming and specialised approaches that take full advantage of problem understanding should be investigated first. On the other hand, classical and specialised methods are often naive, whereas heuristic and metaheuristic paradigms can be utilised for various conditions. In addition, the primary value of heuristic and metaheuristic paradigms is their robustness. In this context, robustness refers to an algorithm's ability to solve a wide range of problems, and even multiple sorts of problems, with only slight changes to account for each problem's specific properties. A stochastic algorithm typically requires the length of the problem-solution vectors, certain information about their encoding, and an evaluation function, with the remaining programme unchanged. A heuristic algorithm is a strategy that uses a rule (or a set of rules) to find (or try to find) appropriate solutions at a low cost of computing. Theoretically, a heuristic provides (eventually) a decent answer with relatively little effort, but this does not ensure optimality. Heuristics are a straightforward way of showing which of many options appears to be the best [162, 163]. The so-called metaheuristic algorithms are an extension of heuristic algorithms. Meta signifies "beyond" or "higher level," and metaheuristics outperform simple heuristics. Heuristics use problem-specific information to identify a "good enough" solution to a given problem, but metaheuristics, such as design patterns, are broad algorithmic notions that may be applied to various problems. Importantly, all metaheuristic algorithms use randomization and a trade-off between local and global search. Because there are no widely accepted definitions of heuristics and metaheuristics in the research, many people describe them interchangeably. On the other hand, recent trends have labelled stochastic algorithms with randomization and local search metaheuristics as stochastic algorithms with randomization and local search metaheuristics. Transitioning from a local to a global search using randomization is a good idea. As a result, practically every metaheuristic algorithm strives to be appropriate for global optimisation [164]. The following features are shared by almost all metaheuristic algorithms [165]:

- They are nature-inspired, relying on physics, biology, or etiology principles,
- They use stochastic components (incorporating random variables),
- They do not use the objective function’s gradient or Hessian matrix. And
- They have multiple parameters that must be adapted to the nature of the problem.

Metaheuristic optimisation algorithms can solve complex problems over several iterations. Because of their inherent versatility and simplicity, metaheuristic algorithms have recently attracted a lot of attention. Metaheuristics can be broadly classified into four different types; the first one is ancient-inspired, mainly based on the Giza pyramid construction. Mutation Reproduction, Recombination, and selection are examples of evolutionary processes that have influenced evolutionary algorithms. These algorithms are based on the survival fitness of candidates in a population (i.e., a set of solutions) for a specific environment. Population-based metaheuristics aim to construct a solution that combines components of good solutions. Trajectory-based metaheuristics are based on the idea of developing a solution and iteratively refining it (moves). The reader is referred to as Refs. [162-170] to get a complete understanding of these concepts. A population-based metaheuristics approach, i.e., nature-inspired, as indicated in Ref. [171] are distinguished by,

- Their search uses a population of points (potential solutions).
- Relying on direct fitness data rather than function derivatives or other similar details
- Using probabilistic, rather than deterministic, transition rules.

Population-based algorithms adopt a similar approach, regardless of the applied paradigm and follow from the algorithm below,

1. Initialise the population.
2. Fitness is calculated for each individual in the population.
3. Produce a new population based on rules that strictly depend on each individual's fitness.
4. Repeat steps 2–4 until a condition is met.

Two of the most popular metaheuristic approaches are described in detail in the following sections.

3.4.2.1 Ant-colony optimisation

Ant Colony Optimisation (ACO) is a well-known bio-inspired method for solving combinatorial optimisation problems. [172]. For ACO algorithms, real ant colonies act as a reference of inspiration. Ants foraging behaviour has an impact on ACO. At the centre of this action is ant communication via chemical pheromone trails, which helps them to locate quick paths among their nest and food sources. Real ant colonies have this feature, which is employed in ACO algorithms to address discrete optimisation issues. In ant colony optimisation algorithms, for instance, an artificial ant is a relatively simple agent that searches for good solutions to a stated optimisation issue. To employ an ant colony approach, the optimisation issue must be transformed into the problem of finding the shortest path on a weighted graph. The ant colony optimisation algorithm is demonstrated using pseudocode. To identify the optimal option, an artificial ant was developed. As the initial stage in solving a problem, each ant creates a solution. In the second stage, the trails discovered by several ants are compared. The path value, or pheromone, is updated in the third stage [173-175].

Initialise the system parameters
  while *termination condition not met,* ***do***
    Construct Solutions
    Apply Path Search
    Update Pheromones
  repeat
end procedure

When all of the ants have completed their solution, the trails are usually altered by raising or lowering the level of trails correlating to moves that were part of "good" or "poor" solutions, respectively. A global pheromone updating rule is as follows:

$$\tau_{xy} \leftarrow (1-\rho)\tau_{xy} + \sum_{k}^{m} \Delta\tau_{xy}^{k} \tag{1}$$

where τ_{xy} represents the amount of pheromone deposited for a state change xy, ρ represents the pheromone coefficient, m represents the number of ants, and $\Delta\tau_{xy}^{k}$ represents the amount of pheromone deposited by k^{th} ant, which is usually given for a Travelling Salesman Problem (TSP) problem (with moves corresponding to arcs of the graph) by

$$\Delta\tau_{xy}^{k} = \begin{cases} \dfrac{Q}{L_k}, & if\ ant\ k\ uses\ curve\ xy\ in\ its\ tour \\ 0, & otherwise \end{cases} \tag{2}$$

where Q is a constant and L_k represents the cost of k^{th} ant's tour.

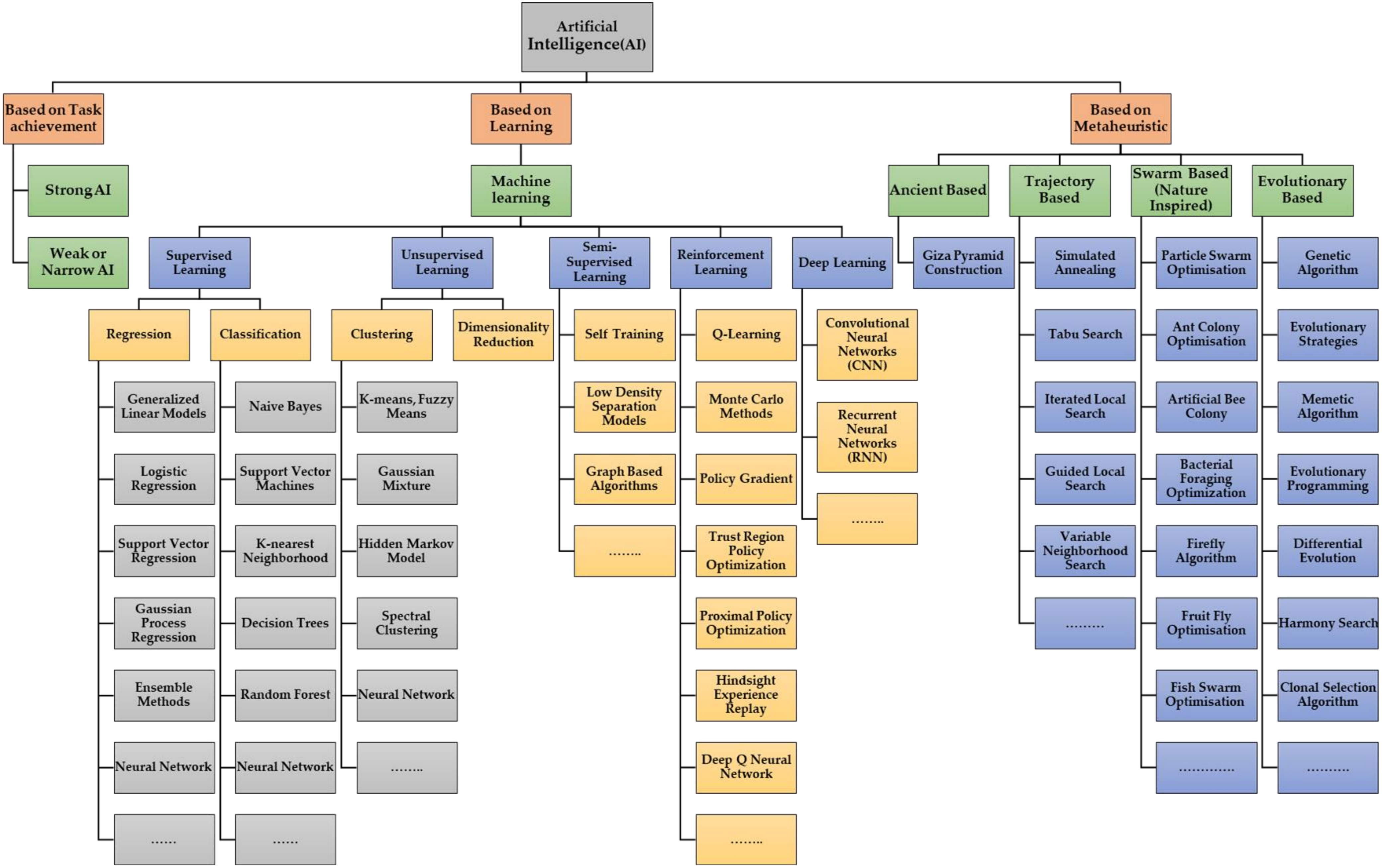


Figure 25: AI techniques in aerospace applications.

3.4.2.2 Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is one of the most extensively utilised metaheuristic algorithms, as evidenced by a series of research [176-183]. PSO is a swarm intelligence method that is based on numerical forms and requires a straightforward implementation. PSO's interaction with nature and societal issues is its most intriguing feature. The algorithm was developed to imitate flocking birds or swimming fish. In PSO, members of a population are candidate solutions to the problem, and the cost function determines the solution's quality. As is the case with most optimisation algorithms, limiting the search domain size improves calculation times.

1. The global version of the PSO converges quickly, but when the problem is extremely challenging, i.e., the cost function being optimised is not convex, it may become stuck in local minima (results can differ based on population initialization or exploration ability).
2. The swarm's exploration capabilities will improve with the local version of the PSO, but the computational convergence duration will be greater than with the global approach.
3. The population-based optimiser is randomly initialised with a collection of potential solutions (particles) and then iteratively searches for an optimal solution by moving the particles inside the problem space. The swarm is made up of p particles that represent different problem solutions in the search space. Each particle's position proceeds as follows:

$$x_{k+1}^{i} = x_{k}^{i} + v_{k+1}^{i} \tag{3}$$

where x is the location of the i^{th} particle at time k increments, and v is the velocity represented by

$$v_{k+1}^{i} = v_{k}^{i} + c_1 \cdot r_1 \cdot \left(p_{k}^{i} - x_{k}^{i}\right) + c_2 \cdot r_2 \cdot \left(p_{k}^{g} - x_{k}^{i}\right) \tag{4}$$

where r_1 and r_2 are uniformly distributed random numbers in the range $[0, 1]$, and c_1 and c_2 are parameters equal to 2 [182] and represent the cognitive and social scaling parameters.v_k does not refer to a velocity in the conventional sense, i.e., $v_k \neq$

$\frac{dx_k}{dt}$, but rather to the rate at which the location per generation shifts. f_{best}^{i} and f_{best}^{g} as best fitness estimate for the i^{th} particle and global solution correspondingly. The pseudocode for PSO is given below.

1. Initialization

(a) Set constant k_{max} , c_1 and c_2

(b) Initialization of particle positions in the problem space x_0^i for p particles

(c) Initialization of particle velocities in the problem space v_0^i for p particles

(d) Set $k = 1$

2. Optimisation

(a) Evaluate the function value f_k^i

(b) If $f_k^i \leq f_{best}^i$ then $f_{best}^i = f_k^i$, $p_k^i = x_k^i$

(c) If $f_k^i \leq f_{best}^g$ then $f_{best}^g = f_k^i$, $p_k^g = x_k^i$

(d) If the stopping criterion is satisfied, go to step (c)

(e) Particle velocities are updated.

(f) Particle positions are updated.

(g) Time is updated $k = k + 1$

(h) Go to step 2(a)

3. Termination

Explicit integration of the Ordinary Differential Equations (ODEs) for attitude and vehicle dynamics using the Dormand-Prince Runge-Kutta technique is possible thanks to the PSO algorithm's excellent computational parallelisation dependability and resilience [184] can be used, while implicit integration methods [185] may be more effective.

3.4.3 Based on Learning

ML approaches is a subset of AI techniques that are based on learning which is a type of data analysis that allows for the creation of analytical models to be automated. It is a branch of AI based on the idea that computers can learn through data, identify patterns, and make judgments with small or no human intervention. An ML process is shown in Figure 26. A model that an application can query is trained based on a data or knowledge base. Regardless of suitable conditioning, data selection, or overfitting, the model improves with a larger database and longer durations of training. If the model can learn in the field, the application can add data to the knowledge base during runtime and train the model with it.

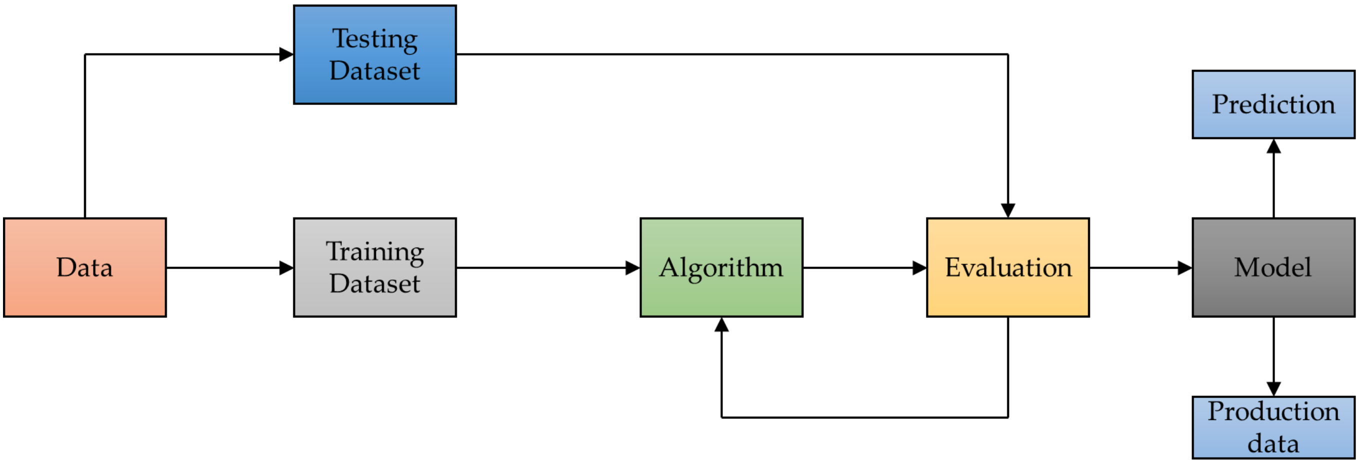


Figure 26: Generalized machine learning process.

A brief review of ML methods and an exposure to commonly used domain terms are provided in the preceding s sections. Different techniques for classifying ML methods have been taken in the literature—the most prevalent taxonomy in which techniques are classified according to the type of learning system used.

3.4.3.1 Supervised Learning

The algorithm is supplied with labelled training data in the form of labels that are included in the desired result in supervised learning. During the training phase, a model is constructed that specifies the link between the training data points' features or characteristics and the labels that belong to them. The model would then be put to the test to see if it would generalise to new information points or 'incidents'. Before being deployed to service, trained models were fine-tuned based on the assessment findings to create a model that extrapolates well with the new data. In most supervised approaches, the learning method keeps track of the difference between the model prediction and the label and uses it as an error term to drive model updating [21, 84].

3.4.3.2 Unsupervised Learning

The training data provided to the algorithm is unlabelled in unsupervised training, and the relationship model is created solely on the data attributes. Unsupervised learning methods include clustering, dimensionality reduction approaches, and association rule-learning methods [21, 84].

3.4.3.3 Semi-supervised Learning

Semi-supervised learning is an ML technique that involves training using a small amount of labelled data and a large amount of unlabelled data. Semi-supervised learning is a sort of learning that falls somewhere between unsupervised (in which there is no labelled training data) and supervised learning (with labelled training data) [84].

3.4.3.4 Reinforcement learning

In reinforcement learning, a model has been trained to learn a behavioural policy through many simulations iteratively, called the training set. Through trial and error, the agent learns how to attain a goal in an uncertain and potentially complex environment. In reinforcement learning, AI is presented with a game-like scenario. The machine uses the method of trial and error to find a solution to the problem at hand. AI gains either rewards or penalties for the acts it takes to persuade the system to perform the actions the programmer desires. Its aim is to increase the overall award. Reinforcement learning is a type of supervised learning in which an agent learns to do the best set of actions or rules to achieve a user-defined reward function, and the architecture is shown in Figure 27. To find the optimal system settings autonomously, sophisticated learning and decision-making must be used by this intelligent space system, and the encouraging solution is to use ML [186].

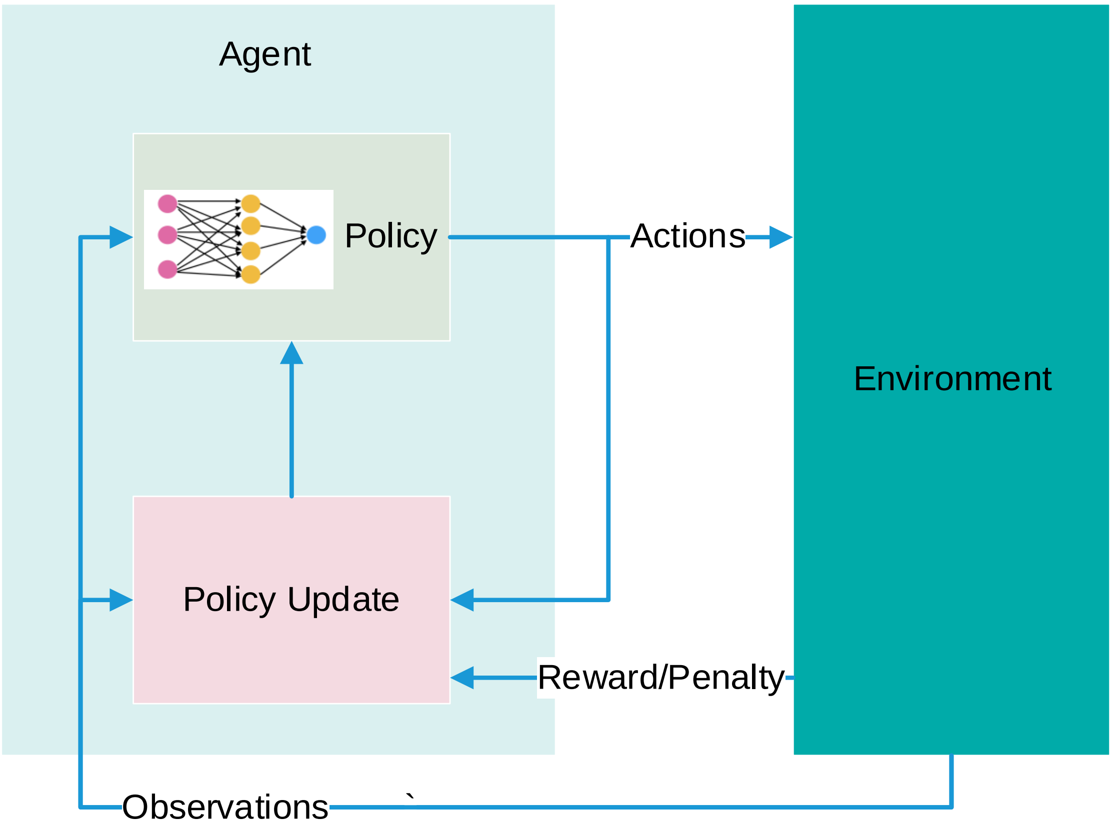


Figure 27: Reinforcement Learning Architecture [84].

3.4.3.5 Deep Learning

Deep learning is a subfield of ML that focuses on algorithms called Artificial Neural Networks (ANN). These networks are modelled after the structure and function of the human brain. The processing units are ANNs that are composed of inputs and outputs. ANN is a kind of ML technology that is inspired by biology and works in the same way as the brain (loosely). A brief overview of ANN is provided to highlight the distinctions between networks as well as the scenarios for which they can be used. The perceptron's architecture is depicted in Figure 28 (a). The input signals to the perceptron are scaled and added using a sequence of weights (typically randomly initialised prior to the learning process). The weighted total is passed via an activation function (typically non-linear) to generate output. A number of structures can benefit from the learning process, also known as iterative weight updates. Backpropagation is one of the most extensively used methodologies in supervised learning systems, as seen in Figure 28 (b). The difference between the output and the desired output is used as an error term to repeatedly change the weights. Although a single perceptron has limited applications, cascaded perceptron combinations could be used to learn complex connections between system/sub-inputs and outputs. The input-output model is really a weight matrix that is iteratively trained and initialised. Figure 28 depicts a layered feed-forward perceptron network. When comparing two neural networks, there are often many points of difference [21, 84]. These are based on the following,

- **Network Topology:** This comprises the number of nodes in a network and their structure. The topology and the constituent nodes also influence the procedures of learning and recall. In addition to the simple perceptron mentioned earlier, other node forms in Figure 27 include [84]:
  - ***Recurrent cells:*** Take input and feedback from previous node outputs as well as from neighbouring nodes. The memory capacity allows for the estimation of the temporal state and time-series forecasts.
  - ***Memory Cells:*** Memory cells in Long Short-Term Memory (LSTM) networks are similar to recurring cells in LSTM networks, but they have three gates: input, output, and forget gates that interrupt, permit, or discard data propagation in the network to solve the explosive gradient problem that plagues traditional recurring neural networks.
  - ***Update/Reset gate:*** These cells are identical to memory cells but with update and reset gates, as opposed to memory cells' input, output, and forget gates.
  - ***Convolution kernels and pooling:*** These cells are the most essential components of co-evolutionary neural networks, and they are primarily employed in image processing and computer vision applications. They work by parsing; rather than using all pixels simultaneously, they employ discrete parts of the input image at each epoch.
- **Learning algorithm:** This algorithm is used to update the network's weights. The most extensively used learning algorithms are:
  - ***Hebbian learning:*** This rule of weight update is based on the rule of Hebb, which is usually applicable to most unsupervised learning algorithms. When both neurons have strongly correlated outputs, the synapse (weighted connection)

between the two neurons is enhanced. For a synapse linking neurons i and j, the weight update rule is mathematically modelled as:

$$w_{ij}(n+1) = w_{ij}(n) + \eta x_i(n) x_j(n) \quad (5)$$

Where η is the coefficient of the learning rate, we can see that an increase in synapse weight is proportional to the product of each neuron's output.

- ***Competitive learning:*** This is another concept of unsupervised weight updating with the principle of 'winner takes all':

$$w_{kj} = \begin{cases} \eta\left(x_j - w_{kj}\right), if\ neuron\ k\ wins \\ 0, \qquad otherwise \end{cases} \quad (6)$$

- ***Error correction learning:*** This is a supervised method of learning where the output goal value is known, and the network iteratively changes weights to converge to the desired output. The rule for updating the weight is as follows:

$$\Delta w_{kj} = \eta e_k x_j \quad (7)$$

Where $e_k = d_k - y_k$the error term or discrepancy between the d_k mark and y_k network output. This method is the origin of the famous algorithm for gradient descent, widely used in supervised algorithms for learning. In fact, most backpropagation learning optimisers available today are based on this principle. Notable examples include AdaDelta, Adaptive Moment Estimation (Adam), RMSProp, Adagrad, Momentum, Gradient Descent (GD) and its variants Batch GD, and Stochastic Gradient Descent (SGD), Mini-Batch Gradient Descent. Among those, Adam is the optimiser to use to train a neural network in less time and with more efficiency.

$$\Delta w_{kj} = \alpha\left(r - \theta_j\right) e_{ij} \quad (8)$$

- **Recall:** This refers to retrieving the data stored in the network after being qualified. Feedforward linear combiners (of n neurons) followed by a non-linear thresholding function are the most common recall technique:

$$y_j = f(\sum_{i=1}^{n} x_i\, w_{ij}) \quad (9)$$

Whereas the recall equation is of the type, for a network with feedback loops:

$$x_j(t+1) = (1-\alpha)x_j(t) + \beta \sum_{i=1}^{n} f\big(x_i(t)\big)w_{ij} + a_{ki} \tag{10}$$

Table 7 describes typical ANN architectures that are classified based on mechanisms of learning and recall.

Table 7: ANN types are differentiated by learning and recall processes [84].

Learning/Recall	*Feedback*	*Feedforward*
Unsupervised	• Adaptive Resonance Theory • Bi-directional Associative Memory • Boltzmann Machines • Hopfield Networks • Principle Component Networks	• Fuzzy Min-Max Classifier • Kohonen's Self-Organizing Feature Map • Linear Associative memory
Supervised	• Gated Recurrent Unit • Long Short-Term Memory Network • Recurrent Neural Network	• Adaline • Convolutional Neural Network • Multi-layer perceptron • Neocognitron • Perceptron • Radial Basis Function Network • Reinforcement learning

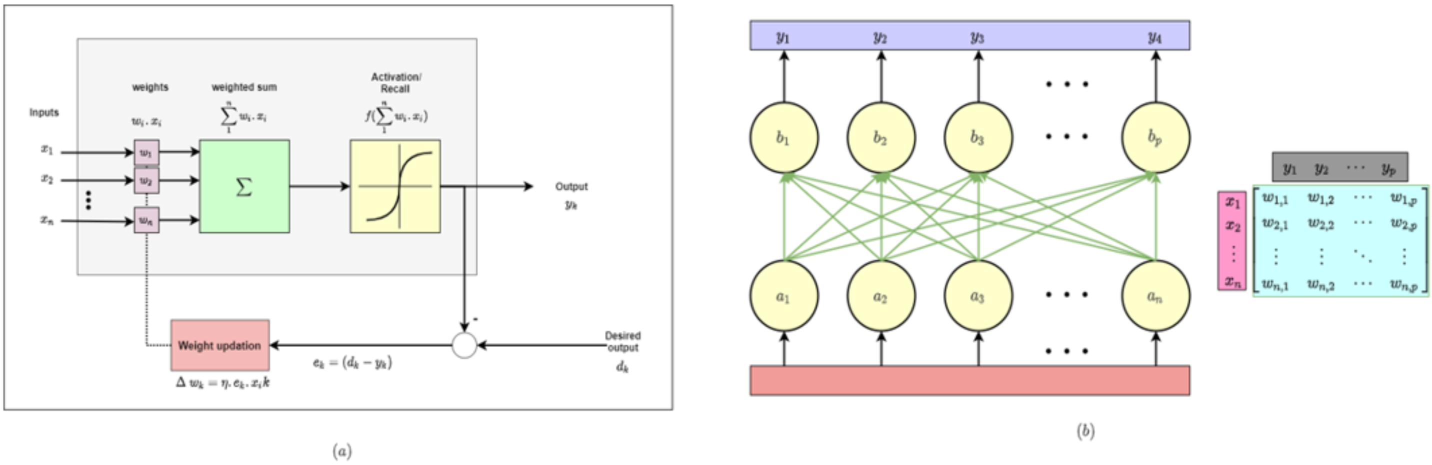


Figure 28: (a)The Biological Neuron Perceptron Model. In an iterative feedback loop, weight adaptation is incorporated; (b) The perceptron multilayer network and the weight matrix map the outputs to the inputs [84].

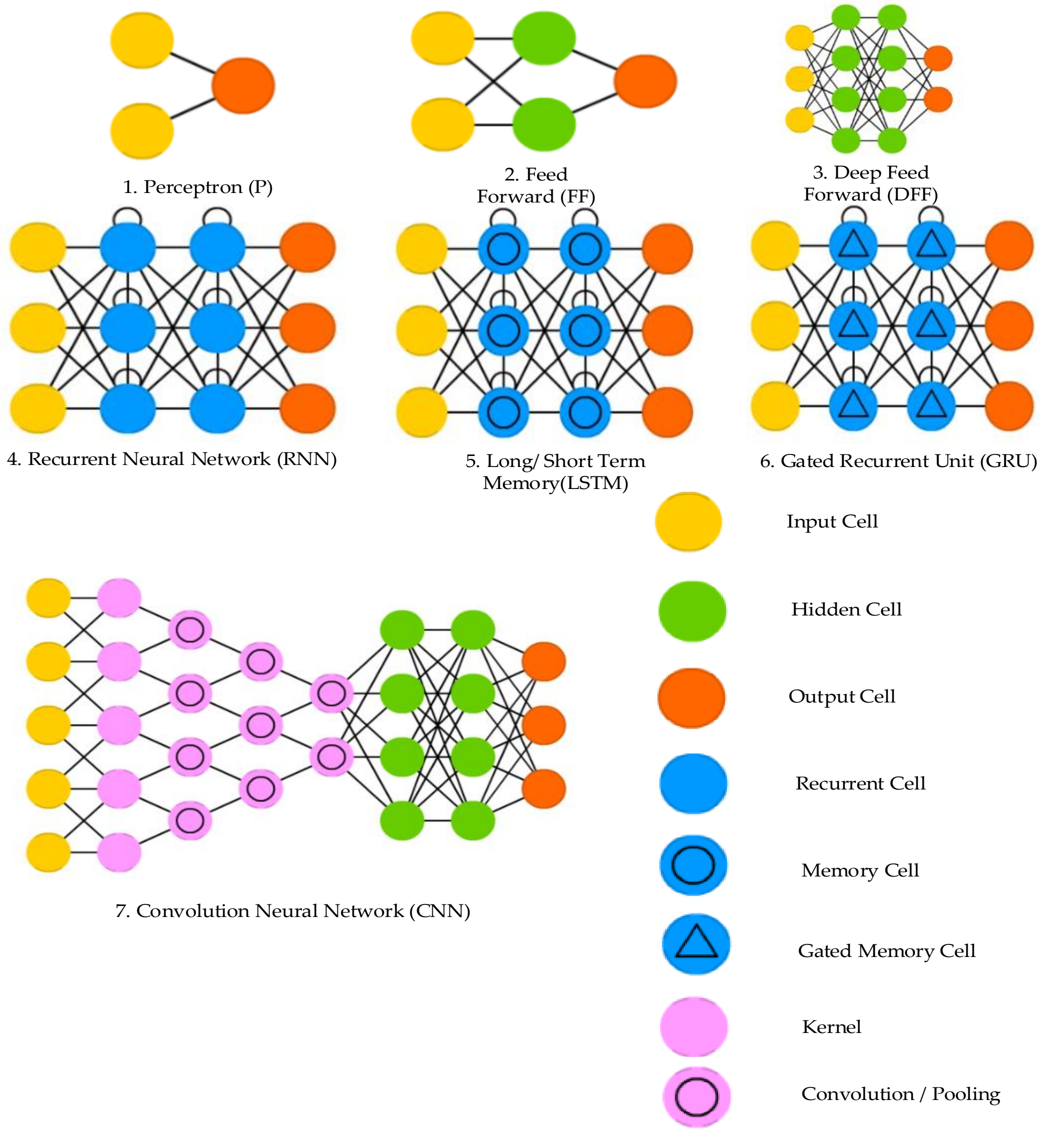


Figure 29: Neural Network Category [84].

Convolution Neural Networks

Convolutional Neural Networks (CNN) are regularised versions of multilayer perceptrons. Such perceptrons are typically fully connected networks in which each neuron in one layer is connected to all neurons in the next layer [99]. CNN is generally used for segmentation, classification, image processing, and other auto-correlated data processing. They are also utilised for speech recognition. Convolution is the process of applying a filter to an input signal as it is being played back. When looking for specific elements in a picture, it may be more productive to look at little sections of the image rather than the entire image at once. Among the most common applications of CNNs is image classification, such as discriminating between satellite images that feature roadways and those that do not. The use of CNNs for other standard functions, such as image segmentation and signal processing, is also a good fit for them. Each layer of a CNN model learns a collection of convolutional kernels throughout the training operation, which is essentially what happens during the training phase. During the deployment of the model, the trained kernels extract spatial information from the image and use these features to make inferences. Each convolutional layer is made up of a collection of filters known as convolutional kernels, which work together to create the final result. Filtering is accomplished by applying a subset of the input pixel values to a matrix of integers that has the same dimensions as the kernel [94, 97, 187, 188]. The operation of the convolutional kernel is depicted in Figure 30.

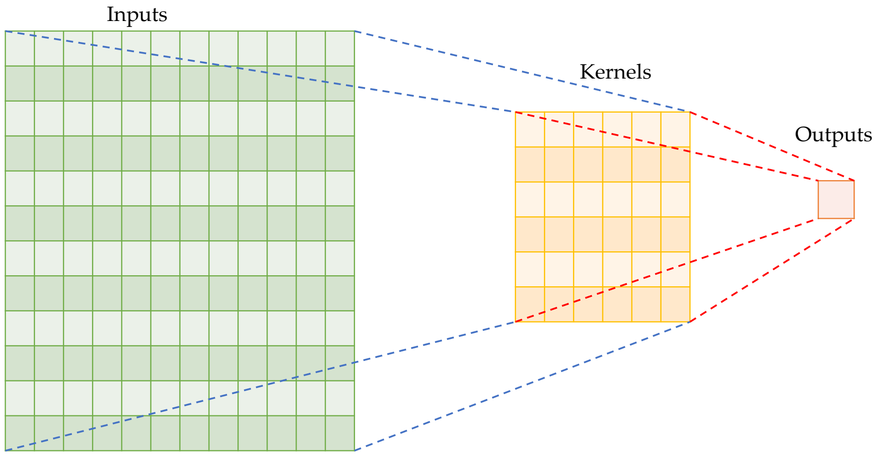
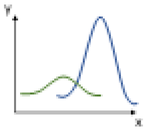
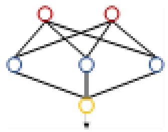
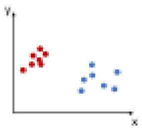
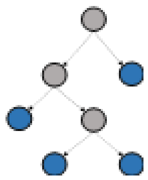
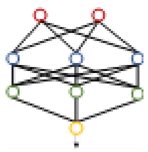
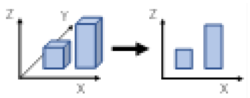


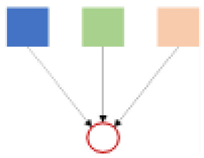
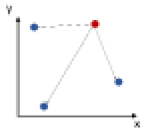
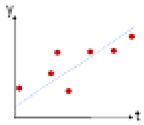
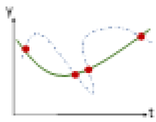
Figure 30: Operation of the convolutional kernel.

Recurrent Neural Networks

Recurrent neural networks (RNNs) keep track of previous outputs at each epoch by integrating feedback loops. RNNs are better at learning temporal relationships in data sequences than CNNs, which are meant to learn spatial patterns [84]. Thus far, the CNNs are classic feed-forward networks in which activations travel from the input to output layers at a predetermined rate. The network output is distinct from the outputs of previous timesteps at any given timestep. The detailed classification is illustrated in Figure 29. There is no sharp divide between these subtypes, even though the concepts seem to vary. The field of data-driven AI has a wealth of valuable and adaptable tools that can be used in various applications with minimal enhancements. Table 8 presents a range of representative instruments and their respective high-level conceptual diagrams.

Table 8: A brief overview of the most widely used AI techniques [21].

Technique	Conceptual Illustration	Description
Association Rule Learning Algorithms		Association rule learning algorithms derive the rules that most accurately describe the observed relationships between variables in the dataset. Valuable and essential associations in large multi-dimensional datasets can be discovered through the formation of these rules. The methods in this class of algorithms include: (1) Fuzzy inference; (2) Adaptive Neuro-Fuzzy Inference System (ANFIS)
Bayesian Algorithms		Bayesian methods tackle regression and classification problems by explicitly applying Bayes Theorem. The methods in this class of algorithms include: (1) Multinomial Naive Bayes; (2) Averaged One-Dependence Estimators (AODE); (3) Bayesian Network (BN); (4) Naive Bayes; (5) Gaussian Naive Bayes; (6) Bayesian Belief Network (BBN)
Classical Artificial Neural Networks		Bio-inspiration from the structure and functioning of naturally occurring neural networks has been a significant factor in the development of Artificial Neural Network models. Essentially, they can be described as a type of pattern matching algorithm widely used for classification and regression problems. The methods in this class of algorithms include: (1) Multi-Layer Perceptron (MLP); (2) Radial Basis Function Network (RBFN); (3) Back-Propagation/Feedforward (BPNN/FFNN).
Clustering Algorithms		Clustering is the process of grouping a collection of objects such that objects in the same category (called a cluster) are more related (on the basis of a single or multiple metrics) to each other than to those in other groups. The methods in this class of algorithms include: (1) k-Medians; (2) k-means; (3) Hierarchical Clustering; (4) Expectation Maximisation (EM)
Decision-trees		Trained decision-tree models use multiple input variables to predict target variable values. The source dataset, which constitutes the root node of the tree, is divided into subsets containing the successor children. A set of splitting rules are built based on classification features. The methods in this class of algorithms include (1) Conditional Decision Trees; (2) Iterative Dichotomiser 3 (ID3); (3) Chi-squared Automatic Interaction Detection (CHAID); (4) M5; (5) Classification and Regression Tree (CART); (6) C4.5 and C5.0 (different versions of a powerful approach); (7) Decision Stump
Deep Neural Networks		Deep Neural Networks are an extension of Artificial Neural Networks that exploit the availability of abundant computational resources. They are characterized by a large number of hidden layers in order to deal with highly non-linear problems. The methods in this class of algorithms include: (1) Stacked Auto-Encoders; (2) Deep Boltzmann Machine (DBM); (3) Deep Belief Networks (DBN).
Dimensionality reduction		Dimensionality reduction methods essentially exploit the inherent structure in input datasets to extract the most influential variables. This proves helpful when visualizing high-dimensional data or simplifying data that can subsequently be used in a supervised learning method. The methods in this class of algorithms include: (1) Principal Component Analysis (PCA); (2) Partial Least Squares Regression (PLSR); (3) Principal Component Regression (PCR); (4) Multidimensional Scaling (MDS); (5) Flexible

		Discriminant Analysis (FDA); (6) Linear Discriminant Analysis (LDA); (7) Quadratic Discriminant Analysis (QDA); (8) Mixture Discriminant Analysis (MDiA)
Ensemble methods		Ensemble methods are models built using several weaker models that are separately trained and whose predictions are merged to boost the accuracy of the overall prediction. The methods in this class of algorithms include: (1) AdaBoost; (2) Gradient Boosting Machines (GBM); (3) Boosting; (4) Gradient Boosted Regression Trees (GBRT); (5) Stacked Generalization (blending); (6) Random Forest; (7) Bootstrapped Aggregation (Bagging)
Instance-based algorithms		Related to Clustering Algorithms. Each instance of input data is compared against a database using a similarity measure to find an optimal match and classify it into groups. The methods in this class of algorithms include: (1) k-Nearest Neighbour (kNN); (2) Learning Vector Quantization (LVQ); (3) Locally Weighted Learning (LWL)
Regression		The relationship between variables is modelled through a curve-fit which is refined iteratively using error measurements in the model predictions. The methods in this class of algorithms include: (1) Ordinary Least Squares Regression; (2) Linear and Nonlinear regression
Regularization		An extension to all ML methods wherein models are penalized on their complexity to support generalization. The methods in this class of algorithms include: (1) Ridge Regression; (2) Least-Angle Regression (LARS); (3) Elastic Net; (4) Least Absolute Shrinkage and Selection Operator (LASSO)

While AI has been used successfully in space, it is still constrained to offline data processing but has not yet been utilised fully "on edge" within spacecraft.

Table 9 shows some algorithms and applications that could be developed and evaluated for future "AI on-board" missions. For more information, the reader should consult Ref [189-191].

Table 9: A brief overview of AI on-board missions. Adapted from [189, 191].

Missions	Applications
Debris removal, Docking, and In-orbit servicing:	i. Feature extraction. ii. Identification against 3D mesh model. iii. Obstacle avoidance.
EO missions (to be scaled to mission size, criticality, and duration)	i. Band co-registration for push-broom multispectral and hyperspectral images. ii. Change detection in time series of Earth Observation images, various resolutions. iii. Cloud detection algorithms (F-mask or Sen2Cor, however, the whole Sen2Cor is quite big, maybe some essential parts of it). iv. Fire/flares detection. v. Image compression (jpeg/CCSDS), (preferably Earth Observation-like picture).

	vi. Increase resolution of all Sentinel-2 bands to 10m/pixel. vii. Monitoring of forest distribution. viii. Monitoring of ice at poles. ix. Open sea objects detection and monitoring. x. Reconstruction involving multiple Images alignment using SURF equivalent, like BRISK or ORB (SURF is patented) and RANSAC. xi. Super-resolution (increase resolution using series of images) through compressive sensing methods, like over-determined equations. xii. Supervised NN Image Classification of Multi-Spectral Images Based on Statistical Machine Learning (TBD if learning speed should be measured as the benchmark as well). xiii. Template matching (scale and rotation invariant) in Earth Observation-type image (e.g., from Sentinel-2). xiv. Vessel detection/identification, integration, and data fusion with AIS receivers - identification of piracy.
Generic Imaging Instrument calibration	i. Active / adaptive optics: wave front analysis + actuation. ii. Auto-exposure. iii. Flat field dynamic correction. iv. Focal plane adjustment and calibration. v. Geometric calibration. vi. Top of Atmosphere calibration.
On-board platform imagers processing	i. Identification of fast-moving meteoroids/disturbance/radiation. ii. Star tracing and multiple sensor data fusion. iii. Orbital propagation.
Planetary Exploration (Autonomous Landing, Robotic)	i. Camera/LIDAR fusion processing. ii. Identification of craters, boulders, obstacle avoidance, automatic path discovery.
Reconfigurable platforms/on-board telemetry analysis, FDIR	i. Adapt platforms to change in requirements or new standards. ii. Autonomous failure prognostic and detection. iii. Autonomous Safe mode management. iv. AI-based FDIR.
Satellite guidance applications	i. Autonomous AOCS management for constellations. ii. Autonomous collision avoidance. iii. Autonomous navigation. iv. Autonomous pointing and/or acquisition (AOCS-in-the-loop). v. Payload-in-the-loop visual-based navigation. vi. SDR / Beamforming / Adaptive Coding and Modulation. vii. Smart FDIR / failure prediction / smart HKTM.
New missions that are possible credits to AI	i. Reconfigurable science (several missions with the same Hardware/ Instrument). ii. Servicing / Non-cooperative approach and rendezvous. iii. Debris detection and removal. iv. On-board feature extraction/mapping. Raw data downlink only On-Demand basis or Added-Value basis. v. Rapid alert: fire, flood, earthquake detection.

3.5 Conclusion

Recent developments in the research of human-machine interaction were discussed in this chapter. Subsequently, by using the human-machine cooperative relationship, it may be possible to optimise the benefits while limiting the potential safety risks of utilising AI technology. For decades, the CHMI² community has used a human-centred approach. The current generation is transitioning from human-centred design to human-centred AI, which is not a novel idea. While a technology-centric approach has dominated the development of AI technology, academics have studied a range of human-centred ways to address the particular difficulties highlighted by AI technology. A solid and comprehensive iDSS solution for space operation is only anticipated through the tailored integration of AI-based approaches. Depending on variables like available sensor data, failure modes/mechanisms, and overall system behaviour, the various methodologies for evaluating the performance of each mission element will change. Nonetheless, it is evident that these methodologies will increasingly rely on AI/ML techniques to facilitate TASO in an environment that is continually changing.

Chapter 4

Disaster Management

This chapter discusses the applicability of iDSS for disaster management, with a particular focus on the management of wildfires. As a case study Australian bushfire that occurred in 2019 is considered as a case study. In addition to that, the development of an AI-based trusted autonomous system for on-board data processing to endow TASO is presented in this chapter.

4.1 Wildfire

Climate change and other environmental issues associated with human activities have recently received much attention in the scientific literature [240]. Such issues include extreme weather events [241], droughts [242], sandstorms [243], rising sea levels [244], tornados [245], volcanic eruptions [246] and wildfires [247]. Wildfires decimate global and regional ecosystems and cause a lot of damage to structures, injuries, and deaths [248, 249]. Due to this, it is becoming increasingly important to find fires and keep track of their type, size, and effects over large areas [250]. To avoid or lessen these effects, early fire detection and fire risk mapping are used [251]. In the past, wildfires were mostly found by people monitoring wide areas from fire observation towers and using simple devices like the Osborne fire finder [252]. Nevertheless, such methods were not very accurate, and their effectiveness could be affected by human fatigue accumulated during long observation periods. On the other hand, alternative sensors designed to detect gasses, flame, smoke and heat emissions usually need extended measurement times for molecules to approach the sensors. Also, since the range of these sensors is small, wide areas can only be covered using a large number of sensors [253]. Rapid advancements in object recognition, DL, and remote sensing have given us new ways to find and track wildfires. New materials and microelectronics have also made it easier for sensors to find active wildfire [254, 255]. There are three primary classifications of extensively used technologies that can identify or observe active fire or smoke conditions in real or near-real time, namely terrestrial, aerial, and satellite systems. These technologies are typically incorporated with visible, infrared, multispectral, or hyperspectral sensors; once the data have been collected, they can be processed by applicable AI algorithms, usually a ML methodology. These techniques rely on either extracting hand-crafted features or on robust AI methods in order to detect wildfires in their earliest stages and to simulate how smoke and fires behave [254, 256, 257]. The different types of fire detection methods are shown in Figure 31.

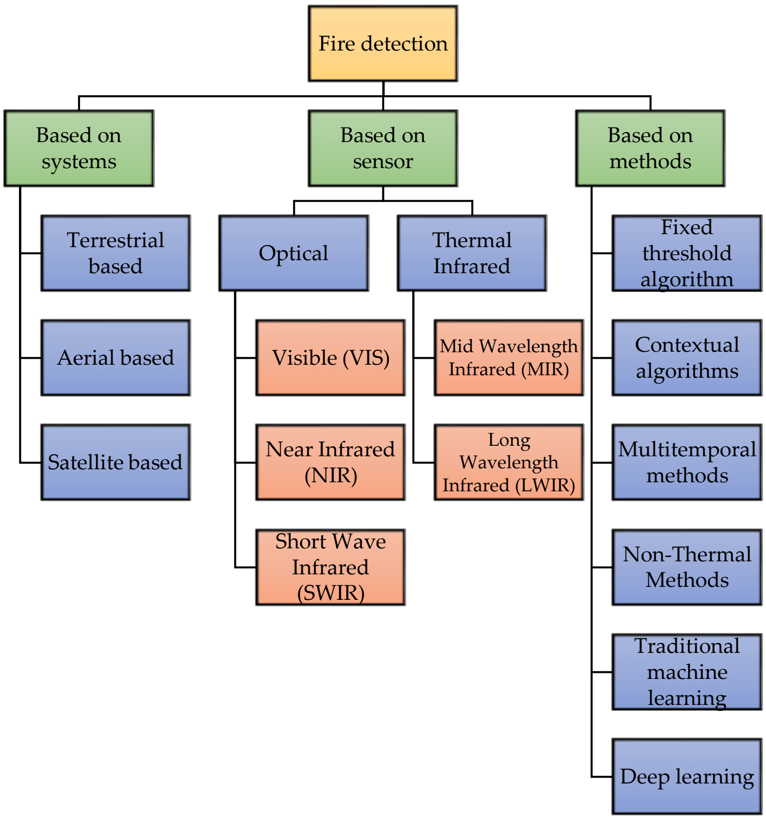


Figure 31: Fire detection methods.

This research focuses on satellite-based fire detection by including appropriate AI approaches for on-board wildfire computation and analysis based on section 4.3.2 suitable AI-algorithm and EO data are employed. Before proceeding, a detailed discussion of the satellite-based wildfire detection approach is provided. There have been numerous research efforts to identify wildfires from satellite imagery in recent years, mostly as a result of the vast number of satellites that have been launched and the drop in associated costs. Specifically, a constellation of satellites (E.g., Planet Lab) was developed for EO [258]. Satellites can be generally grouped into different categories based on their orbit, each with its own advantages and disadvantages. Table 10 shows the most significant categories of the orbits.

Currently, remote sensing satellites take photos of the earth, and the images are downlinked to the ground as soon as the satellite contacts the ground station network. From here, images can be loaded into machines that extract various forms of actionable knowledge, such as wildfire. Downloading imagery is an $O(n^2)$ problem usually provides significant latency when considering critical operations for extreme events management. If time is of the essence for detecting ignitions and thus speeding suppression response, it would be much quicker to have the fire mapping analytics right on board the satellite and only download vector data (either point or polygon) of the fire with the data already flagged to be forwarded to the appropriate wildland fire dispatchers (based on location). Having the coordinates of the event would allow satellite managers or even the satellite itself to prioritize the transmission of the imagery associated with the AI-generated wildland fire event. The mission architecture would be even more effective when considering a constellation of satellites adequately designed to manage extreme events. Having AI on-board of the satellite, data processing can be performed in real-time, and when a wildfire is spotted from one satellite, it will communicate this information to the other satellites in the constellation, thereby enabling TASO. The most important part of this is to show that the data can be processed and shared with the help of the AI that is on-board, and that only the information that can be used is downlinked rather than all the data. Preliminary analyses and results of a mission concept based on DSS for wildfire management is reported in [259].

Table 10: Satellite categories.

Orbit	Altitude	Advantages
Geostationary Earth Orbit (GEO)	Circular orbit with an altitude of 35,786 kilometres and zero inclination	• The satellite does not move at all relative to the ground, • Providing a constant view of the same surface area • High temporal resolution
Low Earth Orbit (LEO)	Altitude of 2000 km or less	• Requires the lowest amount of energy for satellite placement. • Provides high bandwidth and • Low communication latency
Sun-Synchronous Orbit (SSO)	Nearly polar orbit that passes the equator at the same local time on every pass. Typical Sun-synchronous Earth orbits are 600–800 km.	• Satellite will always observe the same scene with the same angle of illumination coming from the sun. • Have high spatial resolution

The majority of low Earth polar SSO orbits of EO satellites, with precise altitude and inclination estimations to guarantee that the spacecraft every time observes the very same scenario with the same angle of light from the Sun and that on each pass, the shadows appear the same [260]. The spatial resolution of sun-synchronous satellite data is high, but the temporal resolution is low (LandSat-7/8 [261] has an eight-day repeat cycle, whereas Sentinel 2A/2B [262] has a two-to-three-day repeat cycle at mid-latitudes). In contrast, GEO satellites have lower spatial resolution contrasts with their high temporal resolution. As a result, they are ineffective for detecting active wildfires in real time; instead, they are better suited for much less time-sensitive tasks such as estimating burnt areas [254]. EO satellite systems have been able to find wildfires because they can see a large area. Most satellites that take pictures of Earth use multispectral imaging sensors and are either in a GEO or Sun-Synchronous Orbit (SSO) region.

Table 11: EO **s**atellites and their characteristics. Adapted from [254].

(Satellite)-Sensor	Spectral Bands	Access to the Data	Spatial Scale	Spatial Resolution	Specs/Advantages/Limitations	Data Coverage	Accuracy Range
Terra/Aqua-MODIS	36 (0.4–14.4 µm)	Registration Required (NASA)	Global	0.25 km (bands 1–2) 0.5 km (bands 3–7) 1 km (bands 8–36)	Easily accessible, limited spatial resolution, revisit time: 1–2 days	Earth	92.75%–98.32%
Himawari-8/9—AHI-8	16 (0.4–13.4 µm)	Registration Required/ (Himawari Cloud)	Regional	0.5 km or 1 km for visible and near-infrared bands and 2 km for infrared bands	Imaging sensors with high radiometric, spectral, and temporal resolution. 10 min (Full disk), revisit time: 5 min for areas in Japan/Australia)	East Asia and Western Pacific	75%–99.5%
MSG—SEVIRI	12 (0.4–13.4 µm)	Registration Required (EUMETSAT)	Regional	1 km for the high-resolution visible channel, 3 km for the infrared and the 3 other visible channels	Low noise in the long-wave IR channels, tracking of dust storms in near-real-time, susceptibility of the larger field of view to contamination by cloud and lack of dual-view capability, revisit time: 5–15 min	Atlantic Ocean, Europe and Africa	71.1%–98%
GOES-16 and 18	16 (0.4–13.4 µm)	Registration Required (NOAA)	Regional	0.5 km for the 0.64 µm visible channel 1 km for other visible/near-IR 2 km for bands > 2 µm	Infrared resolutions allow the detection of much smaller wildland fires with high temporal resolution but relatively low spatial resolution, and delays in data delivery, revisit time: 5–15 min	Western Hemisphere (North and South America)	94%–98%
HuanJing (HJ)-1B—WVC (Wide View CCD Camera)/IRMSS (Infrared Multispectral Scanner)	WVC: 4 (0.43–0.9 µm) IRMSS: 4 (0.75–12.5 µm)	Registration Required	Regional	WVC: 30 m IRMSS: 150–300 m	Lack of an on-board calibration system to track HJ-1 sensors' on-orbit behaviour throughout the life of the mission, revisit time: 4 days	Asian and Pacific Region	94.45%
POES/MetOp—AVHRR	6 (0.58–12.5 µm)	Registration Required (NOAA)	Global	1.1 km by 4 km at nadir	Coarse spatial resolution, revisit time: 6 h	Earth	99.6%
S-NPP/ NOAA-20/NOAA—VIIRS-375 m	16 M-bands (0.4–12.5 µm) 5 I-bands (0.6–	Registration Required (NASA)	Global	0.75 km (M-bands) 0.375 km (I-bands) 0.75 km (DNB)	Increased spatial resolution, improved mapping of large fire perimeters, revisit time: 12 h	Earth	89%–98.8%

	12.4 μm) 1 DNB (0.5–0.9 μm)						
CubeSats (data refer to a specific design from [263])	2: MWIR (3–5 μm) and LWIR (8–12 μm)	Commercial access planned	Global	0.2 km	Small physical size, reduced cost, improved temporal resolution/response time, Revisit time: less than 1 h.	Wide coverage in orbit	-

Improvements in nanomaterials and microelectronics have made it possible to use CubeSats, which are small spacecraft that orbit close to the Earth. PhiSat-1 (Φ-Sat-1), launched on September 3rd, 2020 [18, 264, 265], is a six-units (6U) European satellite and is the first to show how transmitting down EO data can be made more efficient using on-board intelligence using AI. It is part of the FSS, which is made up of two CubeSats [54-57] carrying AI technologies. The two CubeSats collect data using hyperspectral optical equipment and state-of-the-art dual microwaves. They also test inter-satellite communications. One of the CubeSats' hyperspectral cameras takes many pictures of Earth, some of which are cloudy. The Φ-Sat AI chip filters out *erroneous* cloudy photos before transmitting them to Earth, sending only usable data. CubeSats are more cost-effective, are smaller than regular satellites and require less time to launch than traditional satellites. The detailed classification and their parameters are listed in Table 11. Currently, most of the data processing are performed on the ground, but there is a lot of interest in bringing at least some of the computing efforts on-board of the satellite. The employment of AI algorithms on board satellites for analysis and segmentation, classification, cloud masking, and potential risk detection will be the final frontier of satellite remote sensing. The European Space Agency (ESA) has been a leader in taking the first steps in this direction with the PhiSat-1 satellite. CNN for detecting volcanic eruptions using satellite optical/multispectral imaging has been proposed in [18], with the main goal of presenting a feasible CNN architecture for on-board computing. The authors of P. Xu et al. [266], presented an on-board real-time ship detection based on Deep Learning and utilising Synthetic Aperture Radar (SAR) data. Predicting, detecting, and monitoring the occurrence of wildfires obviously benefits officials, civilians, and the ecosystem, with advantages in preparedness, reaction times, and damage control. OroraTech [263], created in 2018, already has a range of international customers for its own wildfire service, notably SOPFEU Quebec, Forestry Corporation NSW in Australia, and Arauco in Chile. The system uses sensor data from a range of existing satellites to offer intelligence for contributing to environmental protection and other properties. OroraTech has launched a Thermal Infrared (TIR) imager on a Spire 6U CubeSat featuring TIR and optical imaging equipment and on-board AI processing in a first step towards vertical integration.

This chapter aims to look into whether AI approaches and on-board computing resources can be used to monitor dangerous events, such as wildfire detections, utilising hyperspectral satellite imagery. The results of this kind of analysis could be useful for future satellite missions, like the ESA Phisat 2 program. In this section, hyperspectral images taken from the PRISMA *(PRecursore IperSpettrale della Missione Applicativa)* satellite were considered, and the following main contributions were made:

1. A One-Dimensional (1D) CNN for detecting wildfires using PRISMA hyperspectral imagery is considered, and promising results are shown for the edge implementation on three different hardware accelerators (i.e., computer hardware designed to perform specific functions more efficiently when compared to software running on a general-purpose central processing unit).
2. It was demonstrated that AI-on-the-edge and iDSS reconfiguration paradigms are feasible for future mission concepts using appropriate architectures and mature astrionics technologies to perform time critical applications.

The proposed CNN is described in terms of the constraints imposed by the on-board implementation, meaning that the initial network has been streamlined and adjusted to comply with the intended hardware designs. It is worth noting that the detection of wildfires should be considered as an example test case, and the proposed methodology (or similar ones) can successfully be applied to other scenarios or tasks, as already discussed and demonstrated in other works [18].

4.2 Current Wildfire Detection Methods

A wildfire is a dynamic phenomenon that changes its behaviour over time. The presence of forest fuel aids the spread of fire. It is carried out by a series of intricate heat transmission and thermochemical processes that control fire behaviour [267]. Several mathematical models were created to characterise wildfire behaviour; each model was built based on diverse wildfire experiences in various nations. According to the input and environmental parameters, each model differs from the others (fuel indexing [268], [269]). Some countries' researchers have been able to incorporate some of these models into simulation programs or even develop their own ways of mapping the terrain and fire behaviour on monitoring screens to study and predict fire behaviour [270]. The form of a wildfire burning in a steady environment is an ellipse [271]. The environment can change over time, and different portions of a fire may be burning in different environments, such as humidity levels, wind speed, wind direction, slope, etc. The heterogeneity of the environment could result in a very complicated fire form [268, 272]. F. Tedim et al. [271] made an initial attempt to develop a gravity scale for wildfires that was comparable to the scales used for hurricanes (Saffir-Simpson scale) and tornadoes (Fujita scale). The first four categories are labelled as "normal fires," or incidents that can generally be put out within the bounds of technology and physical limits. Based on assessments of recent extreme wildfire incidents and a consolidation of literature, the three remaining categories are grouped as Extreme Wildfire Events (EWE; see Table 12). Table 13 shows a list of the most recent and significant wildfire incidents in Australia from 2007 until today. Natural disasters may have caused some of the fires or may have been caused directly or indirectly by human recklessness and environmental misuse (particularly the rise in temperature linked with global warming). One of the worst bushfires in Australian history ravaged Victoria. Many people were killed or injured in the Bushfire, which ravaged many towns and cities, destroying homes, businesses, schools, and kindergartens [273, 274]. From Table 13, it is evident that wildfire events are happening regularly. Since wildfires occur on a regular basis, there is a clear need for wildfire detection. To address this, the recent Australian bushfire is investigated, and an analysis is carried out. The designated AOI is located around 250 kilometres north of Sydney in Ben Halls Gap National Park (BHGNP), comprises 2500 hectares and is 60 kilometres south of Tamworth and 10 kilometres from Nundle. Because the park is located at a high elevation, it receives a lot of rain and has cool temperatures. However, in late 2019, a combination of high temperatures and wind speeds, as well as low relative humidity, created the conditions for high-intensity wildfire behaviour to develop. As can be observed in the PRISMA image acquired on December 27, 2019, active wildfires can be spotted across this AOI.

4.3 PRISMA Mission

A scientific and demonstration mission called PRISMA was launched aboard the VEGA rocket on March 22, 2019. The satellite mission was based on the *HyperSpectral Earth Observer* (HypSEO) project [275], which was a product of a partnership between the Italian Space Agency (ASI) and the Canadian Space Agency, served as the foundation for the early conceptual studies. Due to its ability to capture data globally with a very high spectral resolution and in a wide variety of spectral wavelengths, PRISMA is playing an essential role in the current and future international setting of Earth Observation for both the scientific community and end users. PRISMA offers the ability to collect, downlink, and preserve imagery of all Panchromatic/Hyperspectral channels totalling 200,000 km^2 daily practically on the entire global region, obtaining 30 km by 30 km square Earth tiles. There are two operational modes for the PRISMA mission: a primary mode as well as a secondary mode. The main method of operation is gathering panchromatic and hyperspectral data from specified individual targets as demanded by end users. The mission will have set up continual "background" work in the auxiliary mode of operation that will collect imagery to utilise the system's resources fully.

One modest class spacecraft makes up the PRISMA space segment. The PRISMA payload includes a hyperspectral/panchromatic camera featuring Visible to Near Infrared (VNIR) and Short-Wave Infrared (SWIR) detectors. It consists of a medium-resolution panchromatic camera (PAN, from 400 nm to 700 nm) with a 5 m resolution and an imaging spectrometer with a 30 m spatial resolution that can acquire in a continuum of spectral bands from 400 nm to 2505 nm, i.e., from 400 nm to 700 nm in VNIR and from 920 nm to 2505 nm in SWIR. The PRISMA Hyperspectral Sensor uses the prism to measure the incoming radiation's dispersion on Two-Dimensional (2D) matrix detectors to collect many spectral bands from the same ground strip. The 2D detectors immediately provide the "instantaneous" spectral and spatial dimensions (across-track) of the spectral cube, while the satellite motion (pushbroom scanning concept) provides the "temporal" dimension (along-track).

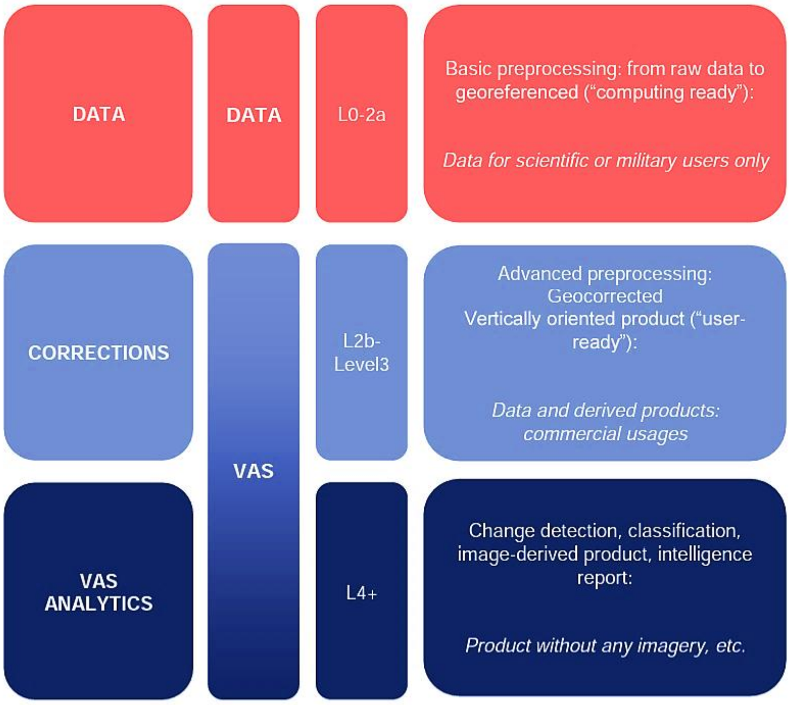


Figure 32: Levels of processing from data to services [276].

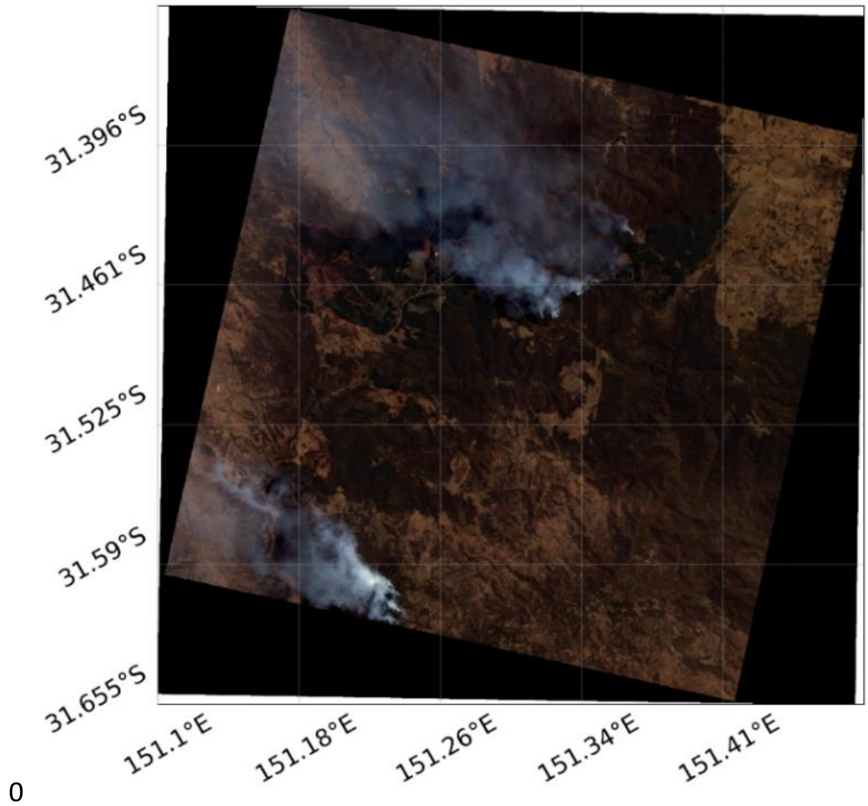


Figure 33: RGB composite of the selected region in New South Wales, Australia, as seen from the PRISMA acquisition.

Table 12: Classification of wildfires based on fire behaviour and capacity of control.1 Adapted from [271].

Fire Category		Real Time Measurable Behaviour Parameters			Real Time Observable Manifestations of Extreme Fire Behaviour ((EFB)				Type of Fire and Capacity of Control *
		Fireline Intensities (FLI)* (kWm^{-1})	Rate Of Spread (ROS) (m/min)	Flame Length (FL) (m)	Pyrocumulonimbus (PyroCb)	Downdrafts	Spotting Activity	Spotting Distance (m)	
Normal Fires	1	<500	<5 <15 b	<1.5	Absent	Absent	Absent	0	Surface fire Fairly easy
Normal Fires	2	500–2000	<15 <30 b	<2.5	Absent	Absent	Low	<100	Surface fire Moderately difficult
Normal Fires	3	2000–4000	<20 c <50 d	2.5-3.5	Absent	Absent	High	≥100	Surface fire, torching possible Very difficult
Normal Fires	4	4000–10,000	<50 c <100 d	3.5-10	Unlikely	In some localised cases	Prolific	500–1000	Surface fire, crowning likely depending on vegetation type and stand structure Extremely difficult
Extreme Wildfire Events	5	10,000–30,000	<150 c <250 d	10-50	Possible	Present	Prolific	>1000	Crown fire, either wind- or plume-driven Spotting plays a relevant role in fire growth Possible fire breaching across an extended obstacle to local spread Chaotic and unpredictable fire spread Virtually impossible
Extreme Wildfire Events	6	30,000–100,000	<300	50-100	Probable	Present	Massive Spotting	>2000	Plume-driven, highly turbulent fire Chaotic and unpredictable fire spread Spotting, including long distance, plays a relevant role in fire growth Possible fire breaching across an extended obstacle to local spread Impossible
Extreme Wildfire Events	7	>100,000 (possible)	>300 (possible)	>100 (possible)	Present	Present	Massive Spotting	>5000	Plume-driven, highly turbulent fire Area-wide ignition and firestorm development non-organised flame fronts because of extreme turbulence/vorticity and massive spotting Impossible

[1] Note: [a] Forest and shrubland; [b] grassland; [c] forest; [d] shrubland and grassland.

Table 13: Mostly relevant wildfires happened in Australia from 2007 to 2021 [277-280].

Year	Event name	Affected area	Burned area (acres)
1 June 2020–1 June 2021	2020–2021 Australian wildfire seasons	Nationwide	617,763
5 September 2019 – 2 March 2020	2019–20 Australian bushfire season (Black Summer)	Nationwide	46,030,000
February 2019	Tingha bushfire	New South Wales	57,870
11 – 14 February 2017	2017 New South Wales bushfires	New South Wales	130,000
January 2016	2016 Murray Road bushfire (Waroona and Harvey)	Western Australia	170,910
25 November – 2 December 2015	2015 Pinery bushfire	South Australia	210,000
15 – 24 November 2015	Perth Hills bushfire complex – Solus Group	Western Australia	24,750
October – November 2015	2015 Esperance bushfires	Western Australia	490,000
29 January – 20 February 2015	2015 O'Sullivan bushfire (Northcliffe – Windy Harbour)	Western Australia	244,440
2 – 9 January 2015	2015 Sampson Flat bushfires	South Australia	49,000
January 2015	2015 Lower Hotham bushfire (Boddington)	Western Australia	129,420
1 August – 9 August 2015	2015 Wentworth Falls Winter Fire	New South Wales	2,000
17 – 28 October 2013	2013 New South Wales bushfires	New South Wales	250,000
18 January 2013	Warrumbungle bushfire	New South Wales	130,000
4 January 2013	Tasmanian bushfires	Tasmania	49,000
27 December 2011 – 3 February 2012	Carnarvon bushfire complex	Western Australia	2,000,000
7 February – 14 March 2009	Black Saturday bushfires	Victoria	1,100,000
30 December 2007	Boorabbin National Park	Western Australia	99,000

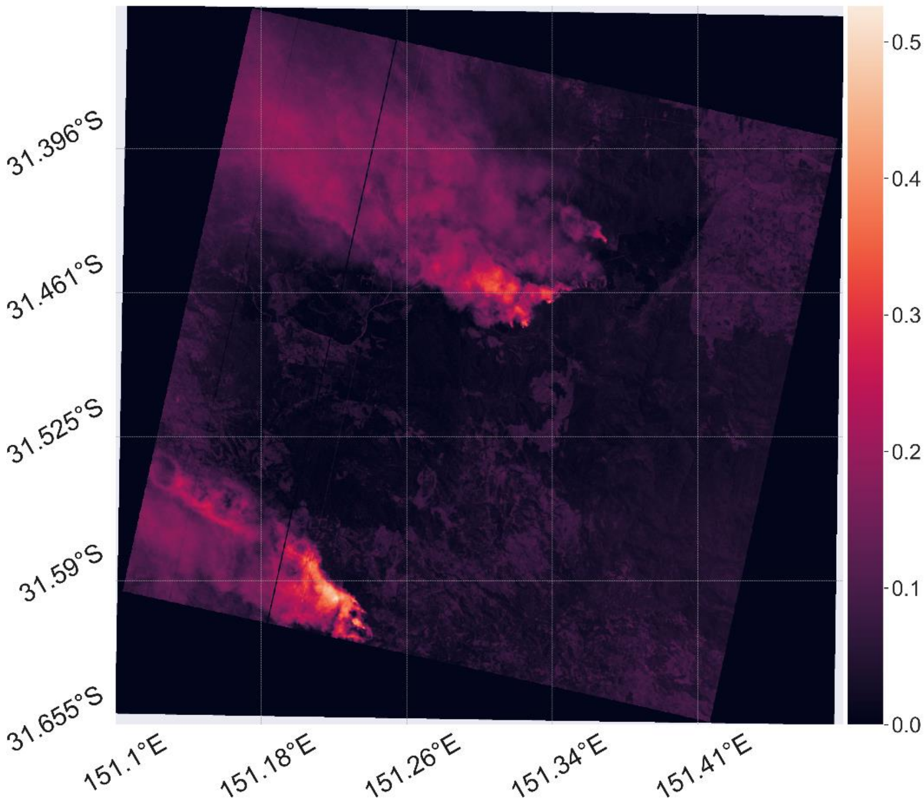


Figure 34: PRISMA level 2D VNIR band at 411 nm.

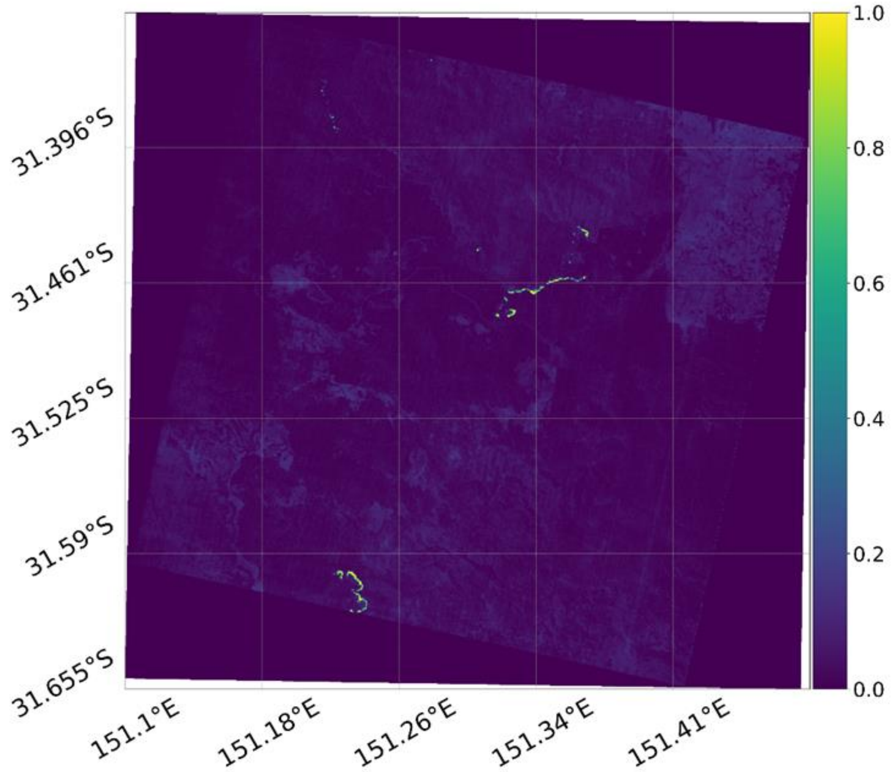


Figure 35: PRISMA level 2D SWIR band at 2490 nm. The three active wildfires are identified.

PRISMA data is made freely accessible for research purposes by ASI [281]. Different levels of data are available, and the differences are reported in Figure 32. In Hierarchical Data Format version 5 (HDF5) format, 30 m and 5 m resolution hyperspectral and panchromatic data are given with four choices:

- Level 1, radiometrically corrected and calibrated Top of Atmosphere (TOA) data.
- Level 2B, Geolocated at-ground spectral radiance product.
- Level 2C, Geolocated at-surface reflectance product.
- Level 2D, Geocoded version of the Level 2C Product.

The analysis in this paper was done with Level 2D data. The RGB composite of the research area is shown in Figure 33. However, direct information can be retrieved by looking at single bands. For instance, by looking at the VNIR bands of the L2D data, smoke can be clearly recognised, as shown in the 411 nm band presented in Figure 34, where smoke pixels can be easily separated from their neighbours. On the other hand, from the far SWIR channels, one can very easily retrieve information on active wildfires, as appreciated in Figure 35. Indeed, when looking at the reflectance product, the signal easily saturates when looking at active wildfires, as the signal captured from Earth is greater than the signal coming from the Sun (since the wildfires behave as an active power emitter).

4.3.1 Dataset Definition

The AI approach is used to implement automatic segmentation from the obtained image. From Figure 35, three active wildfires can be observed. The southern and the north-east wildfires are the bigger ones, whereas the north-west wildfire is quite small. For the training and validation, reference pixels must first be labelled. The reference pixels used in this investigation were manually labelled, and they are shown in Figure 36. The number of labelled pixels selected from the PRISMA image (after investigation of the spectra and looking at the false colour composites) is reported in Table 14. The north-east wildfire has been used as training and validation dataset, while the south and north-west datasets have been used as test datasets. The training set accounts for 70% of the labelled data of the north-east wildfire, while the remaining 30% was chosen for validation.

Table 14: Number of labelled reference pixels in Australia used for training and testing the CNN [218].

		Pixels per classes				
Wildfire Location	**Usage**	**0** *Fire*	**1** *Smoke*	**2** *Burned areas*	**3** *Vegetation*	**4** *Bare soil*
North-East	Train & Val	58	10	30	50	40
South	Test	11	11	9	10	10
North-West	Test	5	0	5	5	5

4.3.2 Automatic Classification with a 1D CNN Approach

The categorisation model utilised in this study was inspired by the Hu et al. [97] model, which is depicted in Figure 36. The PRISMA data's input pixel spectrum includes the SWIR and VNIR channels. Thus, it is an array with $C = 234$ element (after removal of some useless original data in the input hyper-cube). A 1D convolutional layer with a kernel of 3, $n_1 = 112$ filter, same padding, ReLU activation function, and $l2$ kernel regulariser is the first hidden layer. After the convolutional layer there is a max pooling layer with a pool size of 2 and a stride of 2 (notice that $n_2 = n_1$ in Figure 37). The result of this max pooling is then sent through a flattening layer before being connected to a 128-unit fully connected layer with

ReLU activation. A last layer is a dense unit with the SoftMax activation function for multiclass classification. It's worth noting that the values of C_1 and C_2 in the diagram are easily evaluable and rely on the network's architecture. The Adam optimiser and the categorical cross-entropy loss function are used to train the model. Python and Keras were used to build the entire network [282, 283].

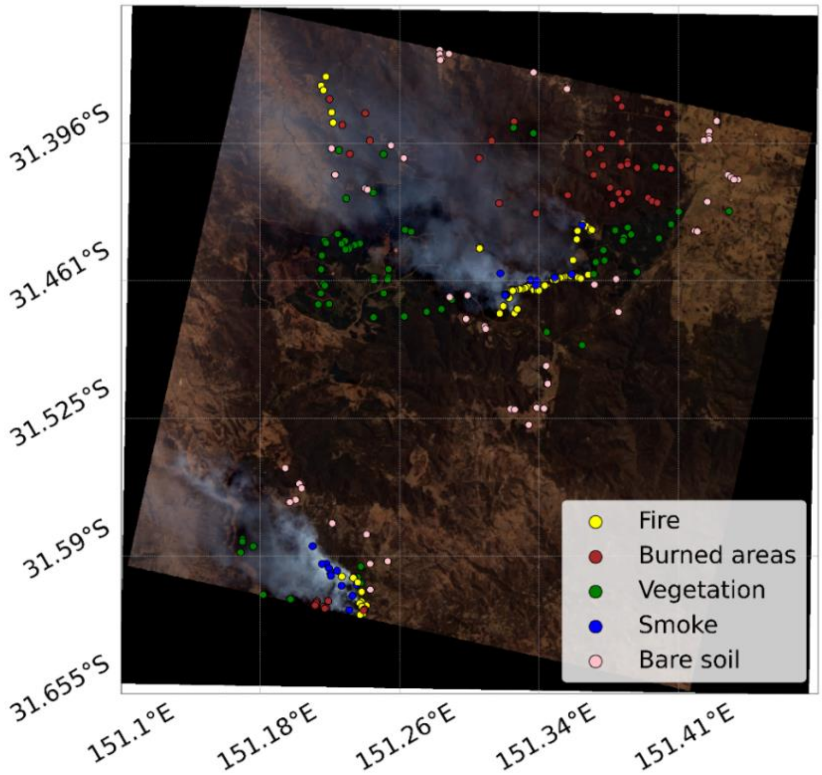


Figure 36: Labelled points defined in the PRISMA image for the five classes.

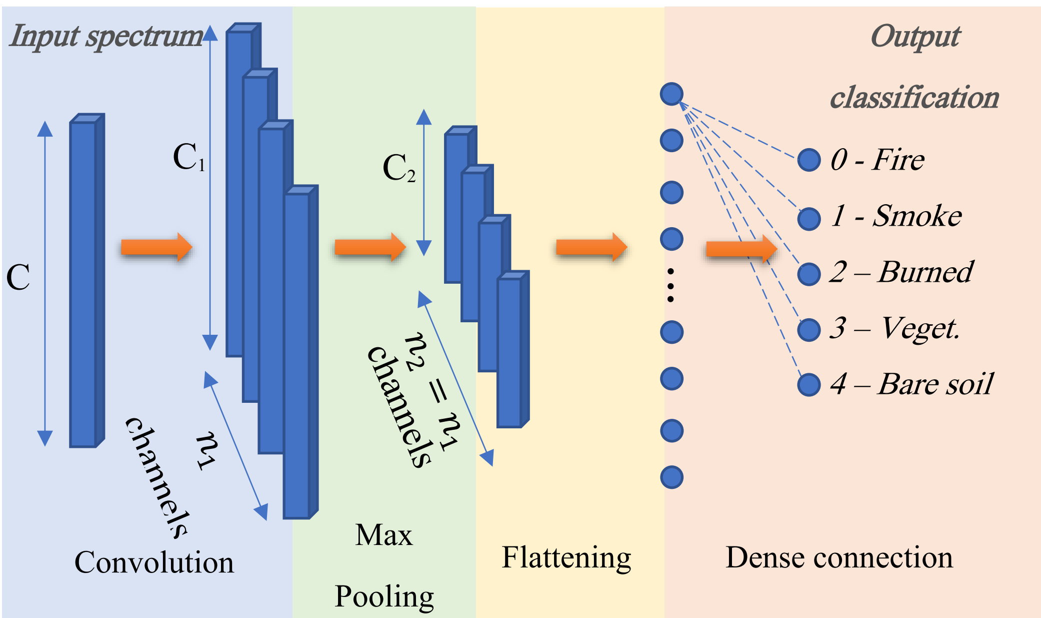


Figure 37: Multi-class classification CNN model [282].

4.4 Astrionics Implementation

The ultimate aim is to build a model that can be uploaded to an on-board astrionics system, so the network complexity, parameter count, and inference execution time must all be optimised. Due to the chip's restricted elaboration power, the utilisation of a small chip limited the ability to execute the specific classification model, necessitating the development of an accurate model. A prototype for executing the analysis has been created in order to evaluate the proposed methodology. The model has been modified to work with the chosen hardware and detect wildfire on-board.

A significant component of the architecture of many current AI solutions is cloud computing or storage. Several sectors find it challenging to apply the technology for real-world use cases due to concerns about confidentiality, latency, dependability, and bandwidth. Despite its resource restrictions, edge computing can somewhat help to ease these difficulties. The claim that edge and cloud computing are incompatible is untrue; edge computing actually enhances cloud computing. Inflated expectations for edge AI and edge analytics have peaked, according to the Gartner hype cycles for 2019 and 2018 [284]. Although the sector is still in its infancy, software frameworks and hardware platforms will advance with time to deliver value at a reasonable price. Three important AI industry leaders—Intel, Google, and Nvidia—are supporting edge AI by offering hardware platforms and accelerators with compact form dimensions. Although each of the three has benefits and drawbacks, it all depends on the application, budget, and amount of experience available; Table 15 compares the hardware accelerators [284].

Table 15: Edge AI device comparison [284].

Parameters	Nvidia Jetson Nano	Google Coral USB	Intel Movidius NCS
Inference time	~38 ms	~ 70 – 92.32 ms	~ 225- 227 ms
fps	~25	~ 9 – 7	~ 4.43 – 4.39
CPU usage	47- 50 %	135 %	87 -90 %
Memory usage	32 %	8.7 %	~ 7 %
OS	Ubuntu 18.04 aarch64	Raspbian GNU/License 10 (Buster)	Raspbian GNU/License 9 (Stretch)

The Intel Movidius Neural Compute Stick (NCS) is a high-performance, affordable Universal Serial Bus (USB) stick that may be used to implement DL inference applications, according to the comparison above. Great AI solutions are provided by the Google Edge

Tensor Processing Unit (TPU). The NVIDIA Jetson Nano, in conclusion, crams a lot of AI power into a little package. Intel Movidius stick and two Nvidia variants, the Jetson and TX2 are used for the research work [217].

4.4.1 Description of the Hardware Accelerators

This section describes the three selected astrionics hardware components, with specific reference to the accelerators, i.e., the Movidius Stick, the Jetson TX2 and the Jetson Nano.

4.4.2 Movidius Stick

The Intel Movidius Neural Compute Stick (NCS) is a compact fanless DL USB drive that is intended to be used for inferencing. The Movidius Visual Processing Unit, which is minimal in power but high in performance, drives the stick. It is equipped with an Intel Movidius Myriad 2 Vision Processing Unit (VPU). These are the main specifications [285]:

- Supporting CNN profiling, prototyping and tuning workflow.
- Real-time on-device inference (Cloud connectivity not required).
- Features the Movidius Vision Processing Unit with energy-efficient CNN processing.
- All data and power provided over a single USB type-A port.
- Run multiple devices on the same platform to scale performance.

The workflow for executing the software modules on the hardware system is depicted in Figure 38. Prior to running the experiments on the Movidius Stick, the CNN must be translated from its original format (for example, the Keras model) to an OpenVino format, which may be accomplished through the use of the OpenVino library. Because of the Movidius Stick, DL coprocessors that are inserted into the USB socket can be inferred more quickly than before. Before transferring the CNN onto the Movidius, it is necessary to optimise the network, which may be accomplished by utilising the OpenVino Intel's hardware-optimised computer vision library. Intel Distribution, the OpenVino toolkit, is extremely easy to use and is included with the Intel processors. Indeed, once the target Central Processing Unit (CPU) has been determined, the OpenCV optimised for OpenVino can handle the rest of the setup. The toolkit supports heterogeneous execution across computer vision accelerators (such as Graphics Processing Unit (GPU), CPU, and Field Programmable Gate Arrays (FPGA)) using a standard Application Programming Interface (API) in addition to enabling DL inference at the edge. It also decreases time to market by utilising a library of functions and preoptimised kernels, and it includes optimised calls for OpenCV.

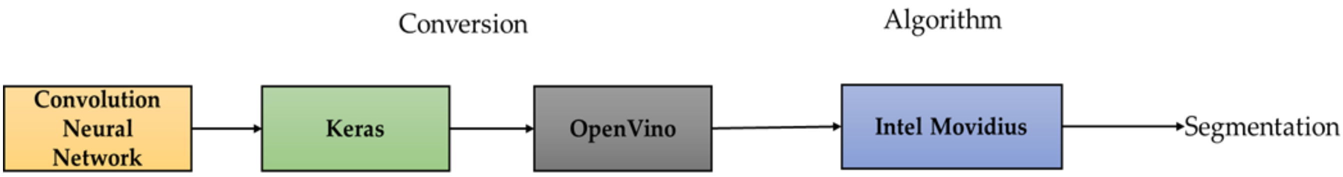


Figure 38: Block schematic for the Movidius stick implementation.

Following the model's implementation on Movidius, it is tested against the PRISMA hyperspectral images. Using the same settings used for the training and validation datasets, the image was processed. Figure 39 depicts the high-level block diagram for the installation and optimisation of OpenVino. The internal data structure or program that a compiler or virtual machine uses to represent source code is known as an Intermediate Representation (IR). An IR is made to facilitate additional processing, such optimisation and translation. The model is fed to the Model Optimiser before being delivered to IR, from which the *.xml* and *.bin* files required to execute OpenVino is obtained. The weight and biases are saved in binary form in a *.bin* file, while the standardised architectural arrangement (and other metadata) is stored in a *.xml* file as shown in Figure 40.

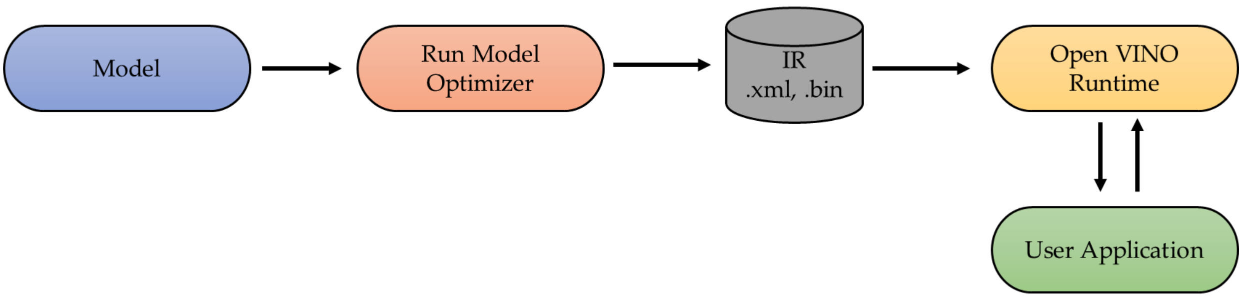


Figure 39: Block diagram for the optimisation and implementation with Intel OpenVino.

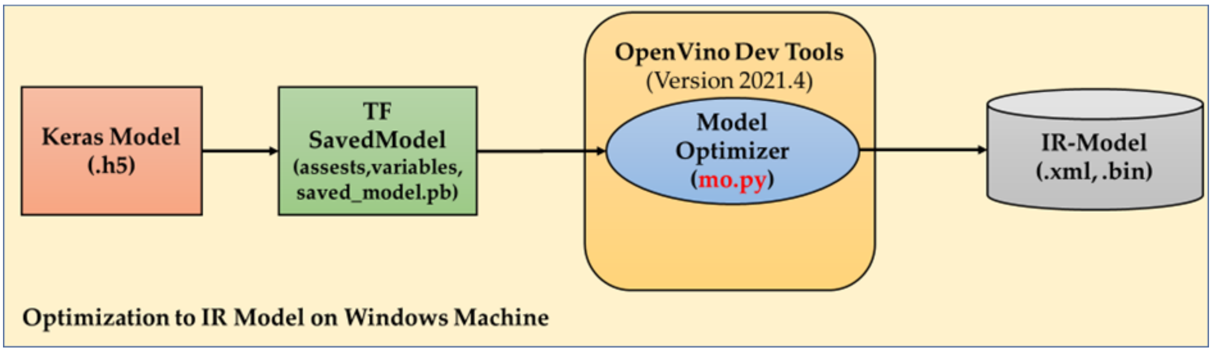


Figure 40: High-Level representation of Keras to IR Model Conversion.

Model optimisation was performed on a Windows machine and the implementation of the IR model on a Windows system with NCS2, as depicted in Figure 41.

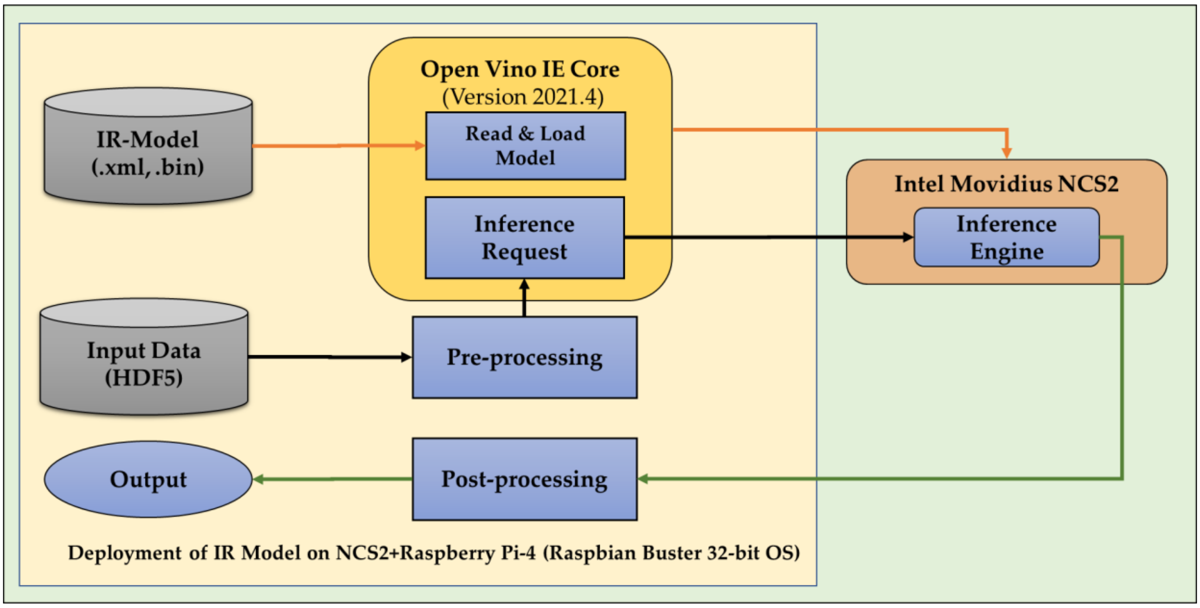


Figure 41: Deployment procedure on NCS2.

4.4.3 Jetson Nano

The recently released JetPack offers a complete desktop Linux environment for Jetson Nano that is built on Ubuntu. Along with major open-source frameworks like TensorFlow, MXNet, Keras, Caffe, PyTorch, and the Software Development Kit (SDK) also enable the native installation of frameworks for robotics and computer vision development like OpenCV. Thanks to complete interoperability with all these frameworks and NVIDIA's high Calibre platform, deploying AI-based inferences applications to Jetson has become simpler than ever before. Jetson Nano makes real-time computer vision and inference possible for a wide range of intricate Deep Neural Networks (DNN) models. Advanced AI systems, Internet of Things (IoT) devices with intelligent edge analytics, and multi-sensor autonomous robots are made possible by these capabilities. Using the ML frameworks, it is even possible to do transfer learning while retraining networks locally on the Jetson Nano [284, 286-289]. The procedure is shown in Figure 41, and the implementation procedure is the same as the Intel Movidius stick.

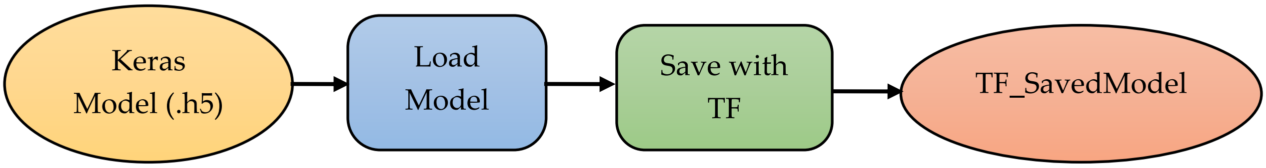


Figure 42: Conversion of Keras model to TF SavedModel.

4.4.4 Jetson TX2

The Nvidia Jetson series of embedded platforms provides edge devices with server-class AI computation capabilities. Regarding DL inference, Jetson TX2 is twice as energy efficient as its precursor, Jetson TX1, and performs better than a Xenon server CPU built by Intel. This increase in productivity reframes the potential for moving advanced AI from the cloud to the edge. With support from LSTM, Recurrent Neural Networks (RNNs), TensorRT libraries and the NVIDIA CUDA Deep Neural Network (cuDNN) library, Jetson TX2 accelerates cutting-edge DNN designs. The conversion of the Keras model to the TensorFlow model is represented in the deployment process in Figure 43, which illustrates this by using simply TensorFlow and TensorFlow TRT Integration to show the vital distinction. To run Neural Network (NN) inferences on their hardware, Nvidia developed the TensorRT NN framework, and the implementation procedures are the same as the Intel Movidius stick. TensorRT is highly performance-optimised on NVIDIA GPUs. Currently, it is probably the quickest method for running models. NVIDIA's TensorRT inference acceleration library enables the utilisation of NVIDIA GPU resources to the fullest extent possible at the cutting edge [289-291].

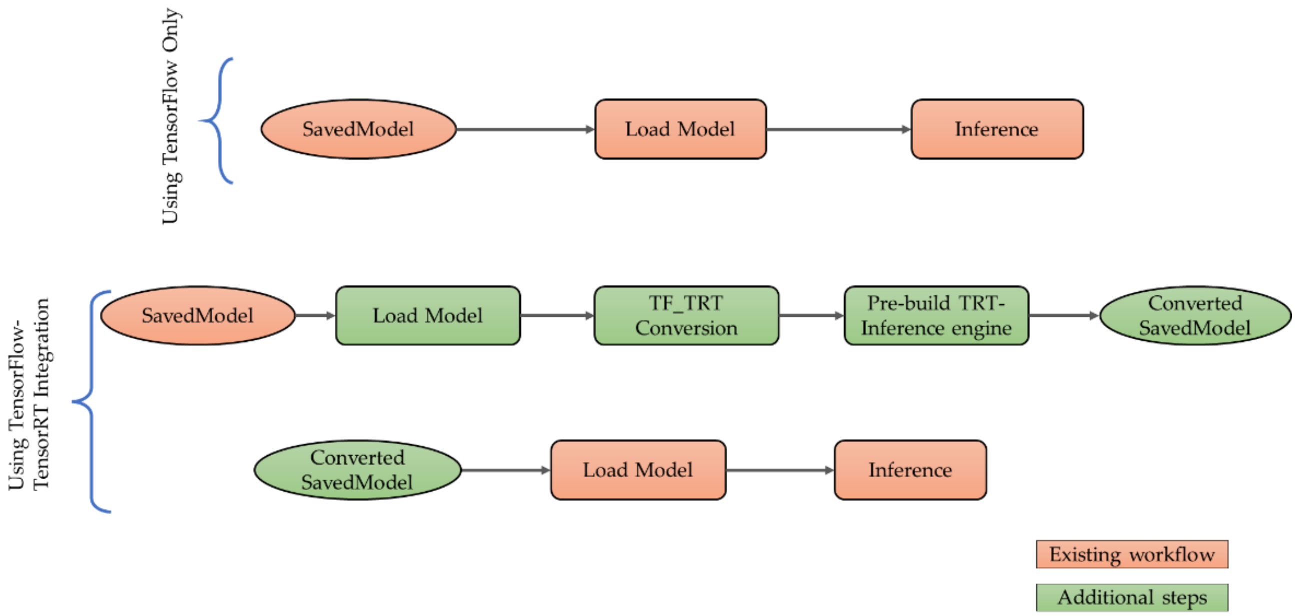


Figure 43: Inferencing procedure in Nvidia Jetson TX2.

4.5 Results and Discussion of 1D CNN

The following Table 16 summarises the outcomes of the training mission conducted over the southern wildfire. Using the validation dataset, the final overall accuracy of the model is 97.83 percent, which is marginally higher than the 96.87 percent reported by Amici et al. [292], where Support Vector Machines (SVM) were employed to achieve the accuracy.

Table 16: Validation dataset accuracy.

	Precision	Recall	F1 Score
0 – Fire	1.00	1.00	1.00
1 – Smoke	1.00	1.00	1.00
2 – Burned	1.00	1.00	1.00
3 – Vegetation	0.92	1.00	0.96
4 – Bare soil	1.00	0.92	0.96
Accuracy			0.98
Macro average	0.98	0.98	0.98
Weighted average	0.98	0.98	0.98

Only the inference problem is considered when evaluating ML implementation on hardware accelerators. This indicates that the training is carried out on a computer with advanced capabilities capable of handling the large amount of data needed for the training. As a result, the training in this paper was also done using ground computing capabilities (i.e., a personal computer with an Nvidia RTX2060), and the findings are presented in Table 17 [282].

Table 17: In the three areas indicated, Precision, Recall, and F1 scores were calculated. The dataset from Australia's North-East was utilised for training, while the others were used for testing.

Wildfire Location	Precision	Recall	F1 Score
Australia, North-East	0.98	0.98	0.98
Australia, South	0.98	0.98	0.98
Australia, North-West	1.00	0.95	0.97

The results obtained by deploying the CNN into the three hardware accelerators revealed that the performances stated in Table 18 were not affected by the deployment of the CNN. As a result, this section describes the deployment performance in terms of the inference time and the amount of power consumed.

a. **Results on the Movidius:** The results of the deployment on the Movidius indicate that the accuracy has not varied compared to the values presented in Table 18. At the same time, the inference time is approximately 5.8 milliseconds, and the computing power is 1.4 watts on average.
b. **Results on the Jetson TX2:** The results of the deployment on the Jetson TX2 reveal that the accuracy has not changed compared to the values reported in Table 18. On the other hand, the inference time is approximately 3.0 milliseconds, and the computational power is 4.8 W on average (2.1 W if considering the power consumed by the GPU only). It is important to note that these results are related to the TX2 setup that provides the least inference time and the maximum power consumption. Other configurations can be set up to lower the amount of power that is consumed, so it is essential to keep that in mind (and increase the inference time).
c. **Results on the Jetson Nano:** The findings of the deployment on the Jetson Nano demonstrate that the accuracy has not changed compared to the values reported in Table 18. On the other hand, the inference time is approximately 3.4 milliseconds, and the computational power is 2.6 watts on average (2.0 W if considering the power consumed by the GPU only). Concerning the Jetson TX2, these findings are associated

with the configuration that offers the quickest possible inference time at the expense of the highest possible power consumption.

Table 18: Inference time and power consumption on three hardware accelerators.

HW accelerator	Inference time (ms)	Power Consumption (W)
Movidius	5.8	1.4
Jetson TX2	3.0	4.8 (2.1 GPU only)
Jetson Nano	3.4	2.6 (2.0 GPU only)

In light of the findings discussed in part before this one, the following is worth further investigation. The deployment of the hardware accelerators in all the reported studies used final models with weights given in float 32. As a result, the precision, recall, and F1 results remain unchanged from those achieved during the Personal Computer (PC) training and testing method. On the one hand, this result was possible because, due to the small dimension of the CNN model, further weight quantisation was not required (i.e., results in inference time and power consumption were already consistent with expected and/or required values without additional weights quantisation). Nevertheless, it should be noted that if the model needs to be improved further in terms of weight compression, embedding, or quantisation, classification performance may suffer. In any case, this investigation's findings show that the weights' data format does not need to be changed for the proposed application, and the classification performances are nearly identical while using a PC or a hardware accelerator.

Table 18 compares the inference timings are perfectly consistent with a real-time early detection service. It should be noted that this time only pertains to the CNN model's inference time and does not include the pre-processing of the image or the extraction of the spectral signature for the pixel of interest. However, this preliminary assessment verifies the method's practicality and the possibility of considering it for future missions. Table 18 also reports power usage that is generally in line with space missions. When it comes to large platforms like PRISMA, all of the reported solutions are in line with the platform's total power budget; however, when it comes to CubeSats or small satellites, the Movidius and the Jetson Nano appear to be the most promising options.

As a review of the preceding arguments, this study highlights the potential for future missions to include on-board hardware accelerators to provide early-warning services [293]. However, if the input data and model complexity are consistent with the ones mentioned here, these conclusions could be applied to other image processing work. The comparison of the three hardware solutions reveals that the Jetson Nano is the most promising technology, as it offers the best combination of power consumption and inference time (even though the final choice may be influenced by other factors such as the hardware accelerator's compatibility with the on-board computer, mechanical and/or electrical interfaces, and so on). Furthermore, it is important to remind the reader that this comparison of accelerator technologies is far from complete, as additional boards exist that were not examined in this work for the sake of simplicity (for instance, the Google Coral TPU or FPGA system-on-chips). This work, on the other hand, answers the question of whether or not AI can be used to handle complex data like hyper-spectral photography, indicating that current technology is ready and efficient.

4.6 Reconfiguration in iDSS

The current state-of-the-art operation of DSS is done when one of the satellites picks/detects the event, then it is sent to the ground control, and the ground control operators will do the reconfiguration, which is not so great in time-critical applications. For real-time/near real-time operation, adding the reactive elements within the architecture endows iDSS operations. This can be achieved through ISL, with which the TASO can be performed in DSS. With ISL, DSS can communicate, interact and cooperate with each other. ISL makes up for the lack of robustness in DSS, which results in an increase in the amount of data exchanged and communication that takes place on-board the satellite in the DSS. Liz Martinez et al. [294] provide the various strategies that are suited for DSS. ISL can be classified as a 1) Ring, 2) Star, 3) Mesh, and 4) Hybrid configuration depending on the communication linkages that are established between the DSS. These topologies are depicted in Figure 44, with the ISL represented by the arrows. Liz Martinez et al. provide a wide range of solutions in their article [294] that are ideal for DSS. Communication by Radio Frequency (RF) is the form of transmission used in wireless networks more commonly than any other method. However, current space optical communication promises a bigger number of benefits, such as an improved data rate, protection, lower power consumption, and a decrease in the weight of satellites. In the past, RF technology was used for inter-satellite communication; however, modern satellites are increasingly resorting to technology based on lasers and optics to connect with one another. Utilising technologies that are based on lasers comes with a multitude of advantages. To begin, infrared laser rays have a greater frequency when compared to RF, which results in a shorter wavelength. As a direct consequence, they can send a greater quantity of data in a single transmission. Second, in contrast to radio waves, lasers experience significantly less dispersion difficulty when transmitted over extensive distances. Because of this, intercepting them is far more complex, resulting in a significant increase in the security guaranteed to the data transfer [295-297]. Figure 45 illustrates the applicability of RF and optical communication (using two versions, i.e., Avalanche Photodiode (APD) and Erbium-Doped Fiber Amplifiers (EDFAs)) by plotting data rate against distance. Laser-based mesh topologies will be excellent for iDSS operations because they are more dependable and well-suited for real-time processing applications [259].

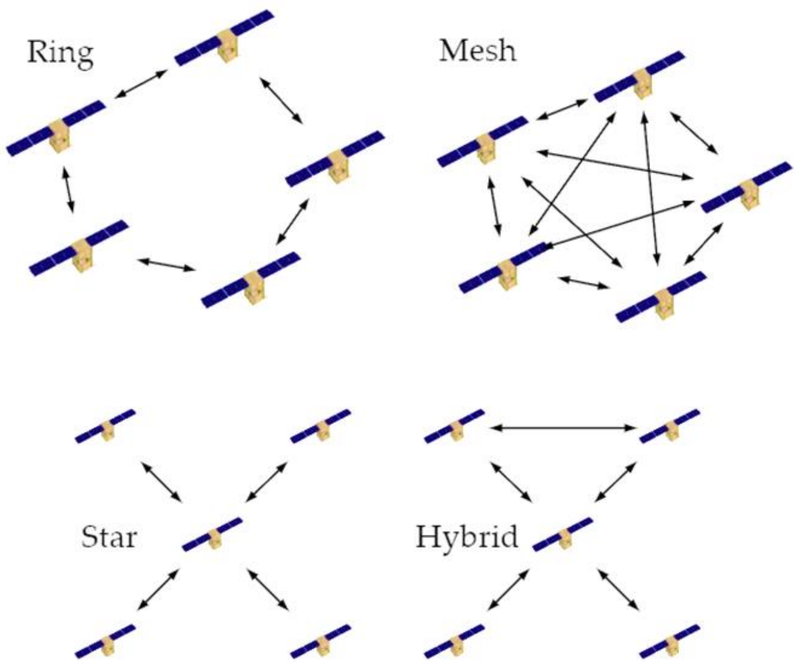


Figure 44: ISL classification [294].

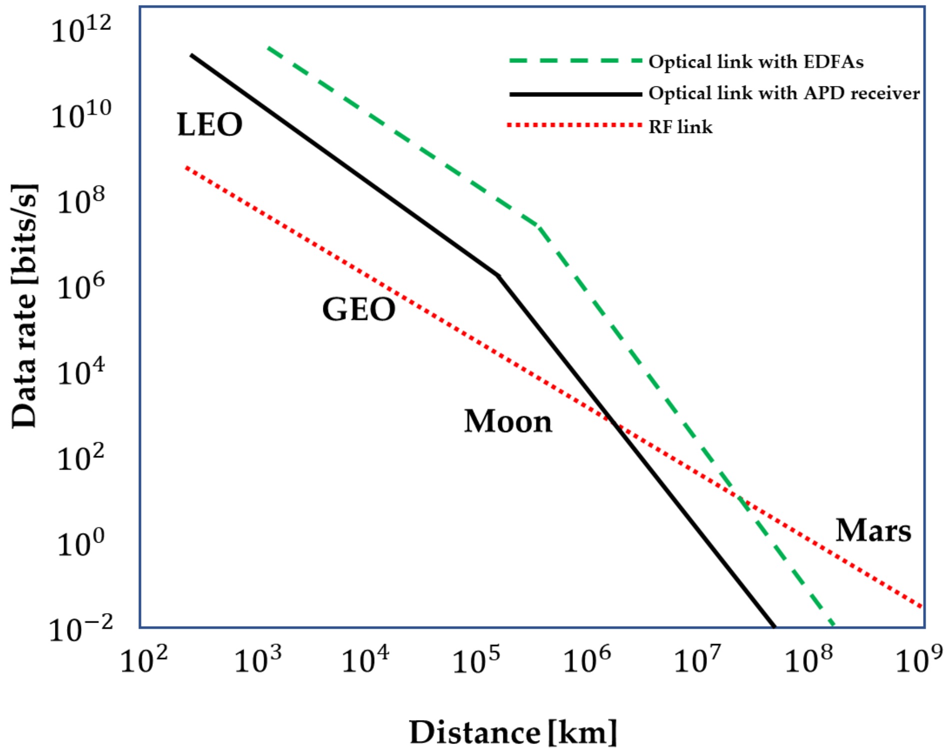


Figure 45: Link distance against data rates for optical and RF ISL systems. Adapted from [298].

iDSS collaborates actively through ISL, sharing information to achieve a common mission objective. ISL relationship between the ground station network, with the iDSS orbital plane and other orbits, is shown in Figure 46. The proposed iDSS is shown in Figure 47 with ISL to provide near real-time disaster management.

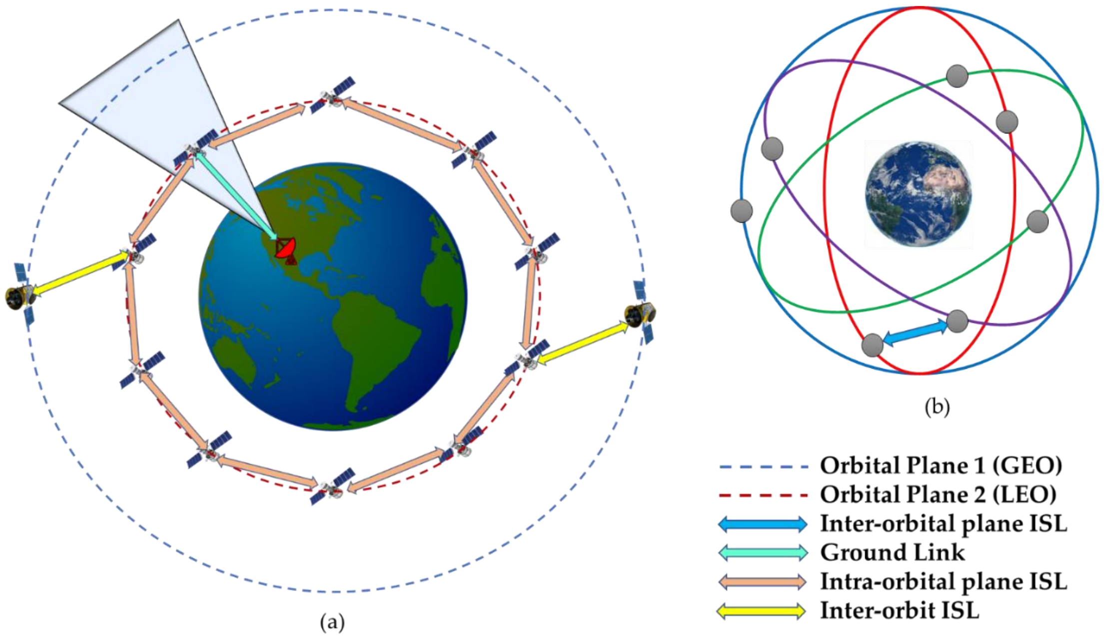


Figure 46: (a) ISL relationship between the orbits and the ground station (b) Proposed iDSS constellation and their ISL.

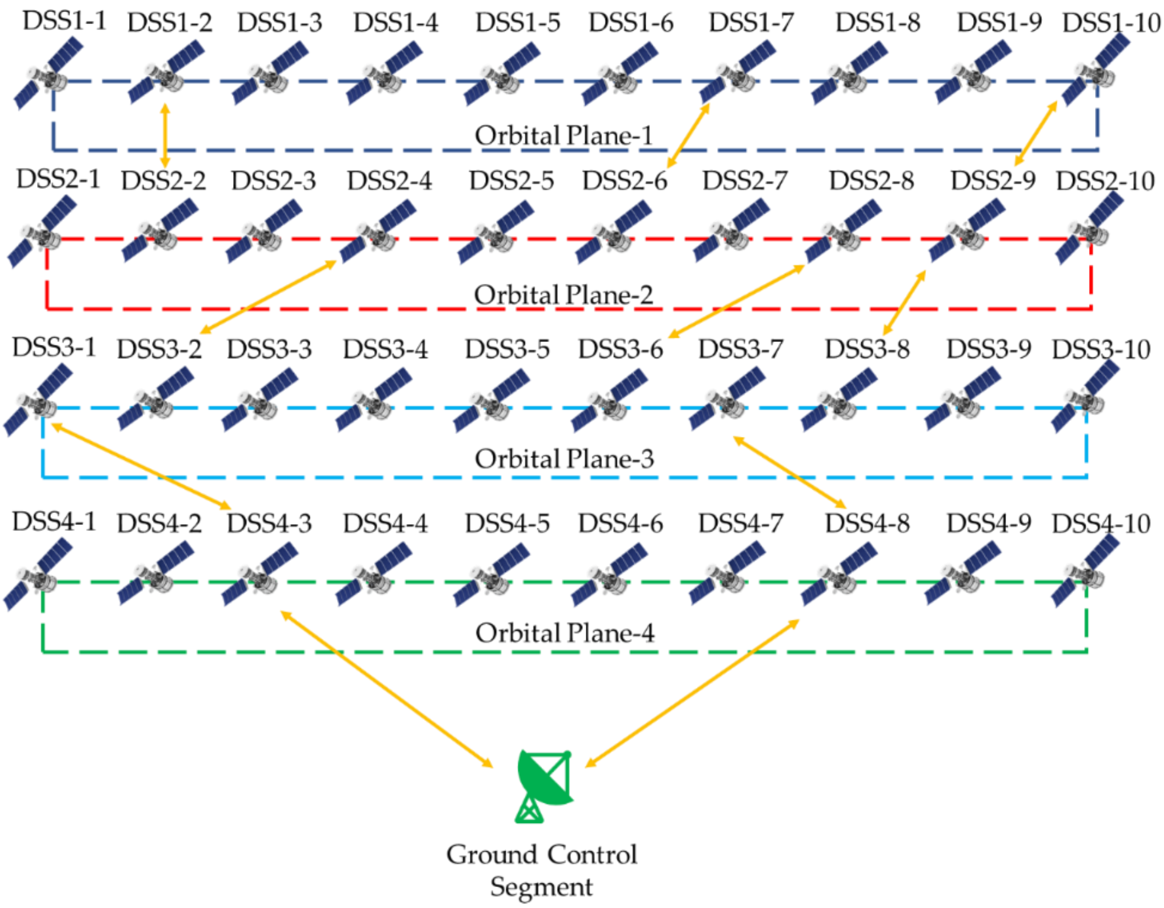


Figure 47: Proposed EO constellation illustration with inter-orbital plane ISL and ground link.

4.6.1 Coverage Analysis

The proposed iDSS is considered in near-circular orbit (i.e., eccentricity is ~0.001) with 500km altitude and inclination 55° with 40 satellites equally spaced (plane spacing 36°) in 4 orbital planes. Here continuous coverage problem, one can disregard the values of the Right Ascension of the Ascending Node (RAAN) and Mean Anomaly. All the participants in the proposed constellation are assumed to be similar and carry the same optical payload. Satellites are often situated in orbital planes that are complementary to one another, and they communicate with each other through ISL and globally dispersed ground stations. The Walker scenario will be suitable for the constellation model. Further, the Walker Delta design is appealing for this research work because of its simplicity and economic feasibility [46, 224]. The parameters i, N_s, p, and f indicate the distribution of satellites in space, where i is the inclination, N_s is the number of spacecrafts, p is the total number of orbital planes, and f is the phase difference between the participating spacecrafts in the plane that forms the Walker Delta constellation pattern. The number of satellites in each orbit is given by $s = \frac{N_s}{p}$, where

$p \mid N_s$ (p divisible by N_s). To avoid satellite collisions, the phase difference between the neighbouring spacecrafts of a specific plane is calculated using $f \times \frac{360°}{S}$, where f is an integer between 0 and (p – 1). iDSS are adaptable based on the owner/operator requirements. In general, all the satellites will be in cyclic planning, and the camera is always nadir pointing. If one of the satellites in the constellation detects the event, i.e., wildfire, then the satellite communicate to the rest of the constellation, and the objective will be, to take as much imagery as possible and do the data processing on-board with the hardware accelerators. Then send the actionable information to the owner/operators. A typical reconfiguration is shown in Figure 48, where once the wildfire is detected, the satellite will do an active Attitude and Orbit Control System (AOCS) for real-time/near real-time event management.

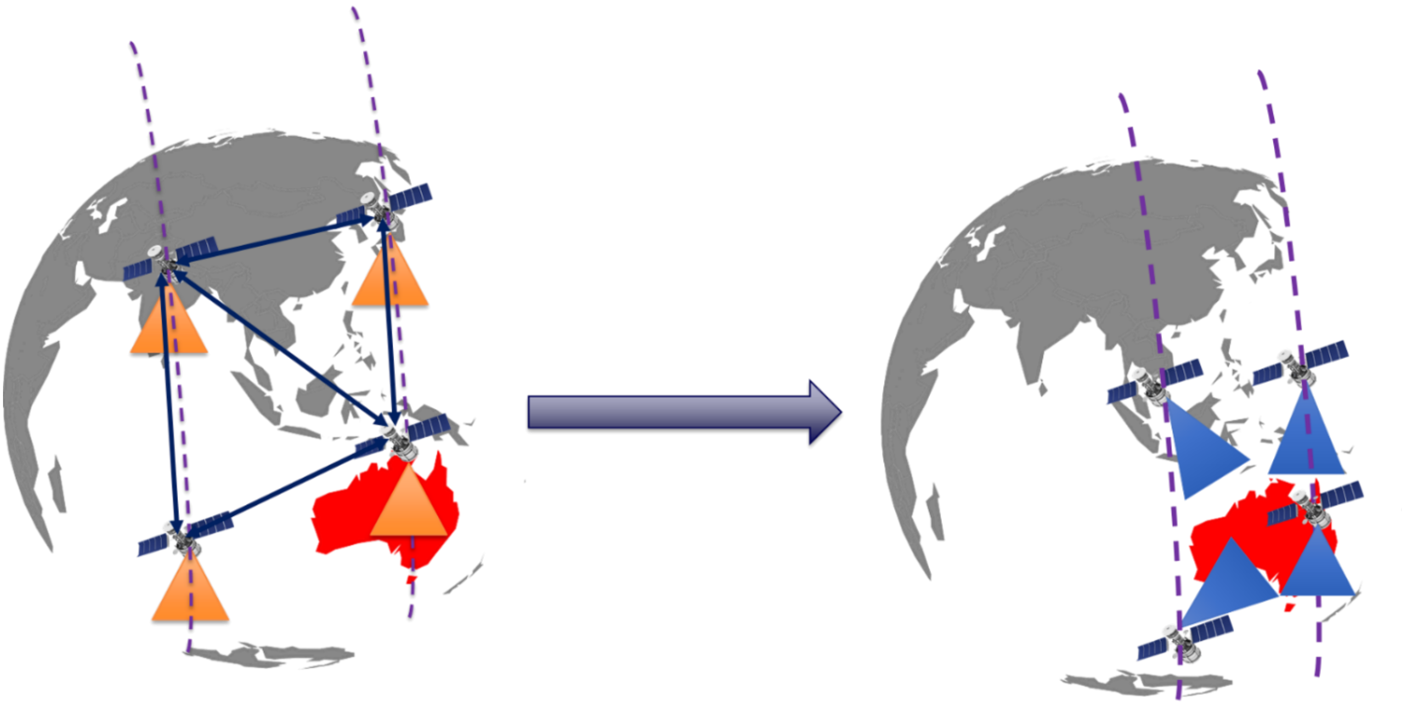


Figure 48: iDSS reconfiguration.

As a satellite observes a region on Earth, it projects a circular or rectangular imprint on the surface. The instantaneous coverage of the satellite is the distance between the satellite and a target point in the satellite FOV region (imprint region) at a given time [197, 198]. Another fundamental parameter for the computation of the coverage and *System Wide Access,* which is the time that at least one satellite's camera can observe the AOI during this timeframe, must be calculated in order to compute coverage and system wide access. The corresponding percentage quantity is known as the system-wide access percentage, and it is calculated using the following equations:

$$SWAD = n \cdot S_c \tag{11}$$

$$SD = ST_{Start} - ST_{Stop} \tag{12}$$

$$SWAP = \frac{SWAD}{SD} \cdot 100 \tag{13}$$

where $SWAD$ is System Wide Access Duration, n is the number of elements in system wide access status whose value is true, i.e., 1, S_c is the spacecraft sample time, and $SWAP$ is the System Wide Access Percentage. The equations that relate to the above are for the Nadir pointing, and they can also be used for systems with reconfiguration:

$$SWAD_T = N \cdot S_c \tag{14}$$

$$SD = ST_{Start} - ST_{Stop} \tag{15}$$

$$SWAP_T = \frac{SWAD_T}{SD} \cdot 100 \tag{16}$$

where, $SWAD_T$ is system wide access duration with reconfiguration, N is the number of elements in system wide access status with reconfiguration whose value is true, S_c is the spacecraft sample time, which is considered 30 seconds for both cases, and $SWAP_T$ system wide access percentage with tracking.

4.6.2 Results and Discussion

The TASO can be accomplished by including reactive elements in the iDSS. In our research works, a 1D-CNN was investigated for spotting wildfires on-board the satellite employing PRISMA hyperspectral imagery and encouraging results for the edge implementation on three different hardware accelerators were demonstrated. It was demonstrated that AI-on-the-edge paradigms for future mission ideas are viable by utilizing appropriate CNN architectures and established technology to perform time- and power-efficient inferences [217, 256, 257]. The analysis in this research was done with Level 2D data; the analysis's AOI is shown in Figure 49.

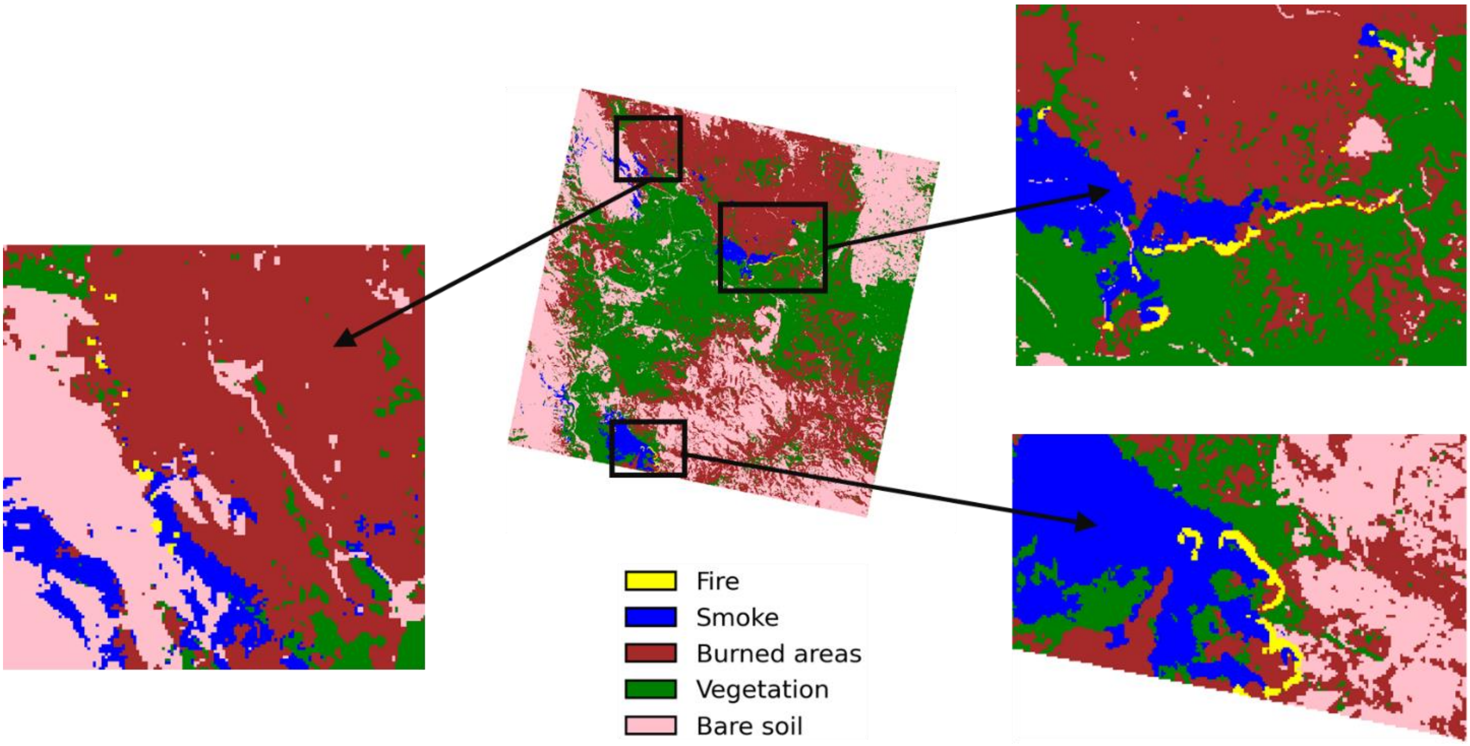


Figure 49: Wildfire segmentation of Bushfire [257].

All of the reported results in Section 4.5 (Table 18) are in line with the KANYINI spacecraft platform's total power budget. From our previous work [217, 256, 257], the Intel Movidius (Inference time is 5.8 ms and Power consumption is 1.4 W) and Jetson Nano (Inference time is 3.4 ms and Power consumption is 2.6 W) appear to be the most promising options. For our situation, the Jetson Nano and Intel Movidius are considered on-board the constellation for detecting wildfires. With the above results and our previous work, the TASO is possible by incorporating reactive elements in the iDSS architecture.

This simulation depicts an investigation of the AOI on the ground and conical sensors on-board a heterogeneous constellation[2] of satellites. The AOI and a satellite's conical sensor are said to have access if the ground station is within the conical sensor's FOV and the conical sensor's Elevation Angle (EA) with respect to the AOI. The simulation employs a constellation of 40 LEO satellites at 500 km altitude to replicate the KANYINI mission in a near circular orbit with AOI. The AOI is chosen to generalize the simulation results based on the wildfire occurrence in the four different continents. Each satellite carries a 30-degree FOV

[2] ***Homogeneous constellation:*** A constellation whose member spacecraft employ functional identical bus, payload, and operational characteristics (e.g., MMS and Iridium).

Heterogeneous constellation: A constellation whose member spacecraft employ different bus, payload, and operational characteristics.

camera, and the entire satellite network is tasked with imaging the AOI during the sun sufficiently illuminates it. The satellite's EA regarding the AOI should be at least 30 degrees to acquire high-quality imagery with minimal atmospheric distortion. Calculating the times when each satellite can image the site over an imposed 6-hour interval is necessary. It is also necessary to calculate the percentage of time that at least one satellite's camera can observe the place during this timeframe which SWAP provides. The existence of the AOI within the contour indicates that it is within the FOV of the payload camera. Figure 50 (a) shows the visualization FOV of the Satellite. It is necessary to calculate the system-wide access percentage, which is the percentage of time from the simulation start time to the stop time when at least one satellite can image the site, and calculate the times when each camera can capture the AOI.

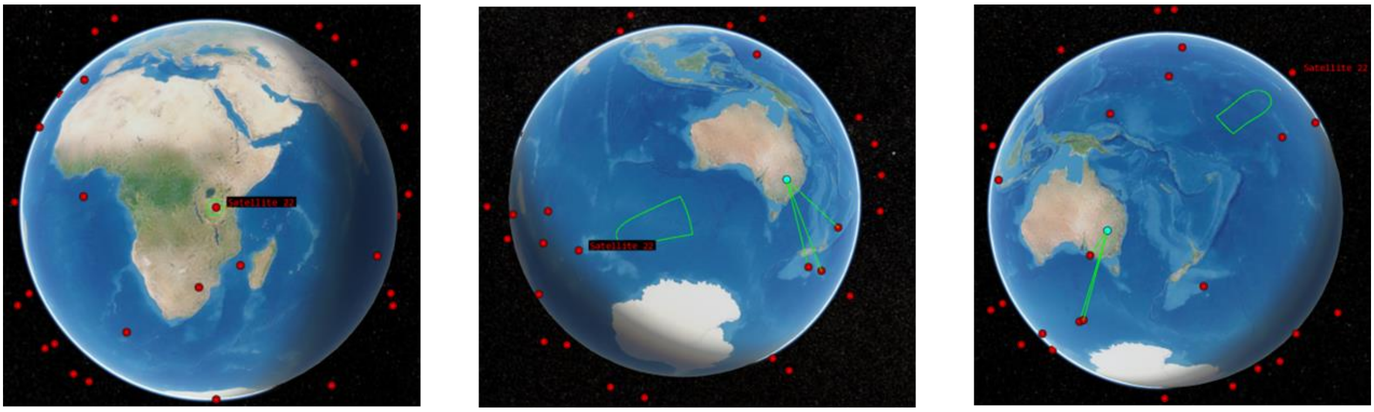


Figure 50: Satellite Field of View (a) Nadir pointing (b) Reconfiguration at the entry (c) Reconfiguration at the exit.

The satellite's default attitude arrangement is Nadir pointing. Because the cameras are by default aligned with the yaw axis, they always point straight down, and the AOI is no longer visible to the cameras before their EA falls below 30 degrees. As a result, this cumulative access percentage is constrained by the FOV of the cameras. Then the satellite cameras will be continuously pointed at the AOI through active attitude control adjustment; the AOI is observable as long as the Earth is not in the way, as seen in Figures 50 (b) and (c). As a result, the system-wide access percentage will now be limited by the AOI's minimum EA rather than the camera FOV. This is done based on the owner/operator requirement for the requested time period. The access periods in the former scenario began and terminated when the site entered and exited the camera's FOV. Specifically, it enters the FOV after the camera's EA exceeds 30 degrees and exits before the camera's EA falls below 30 degrees. The camera will be pointed at Nadir for the rest of the period. The system-wide access for four different AOIs is shown in Table 19 for the NADIR pointing and tracking configuration. Table 19 contains the latitude and longitude coordinates of the four selected AOI, expressed in the World Geodetic System (WGS84). The total simulation was carried out for 21600 seconds, i.e., 6 hours, and the respective system wide access with Nadir pointing and tracking is reported. Because the cameras are firmly affixed to the satellites, each satellite must be constantly reoriented (i.e., manoeuvred with the on-board actuators) along its orbit so that its yaw axis tracks the AOI location.

Table 19: System wide coverage parameters with respect to a scenario duration of 6 hours, i.e., 21600 seconds.

Location	Latitude (deg)	Longitude (deg)	n	SWAD (sec)	SWAP (%)	N	$SWAD_T$ (sec)	$SWAP_T$ (%)
Africa	11.2027	17.8739	10	300	1.3889	545	16350	75.6944
Australia	-31.25	146.92	5	150	0.6944	691	20730	95.9722
Europe	40.1209	9.0129	10	300	1.3889	712	21360	98.8889
North America	44	-120.3	14	420	1.9444	699	20970	97.0833

In Figure 51, the list of satellites that will access the AOI in the proposed constellation and the length of time they will have access to Australia is shown, and their corresponding orbits are presented for the simulation time.

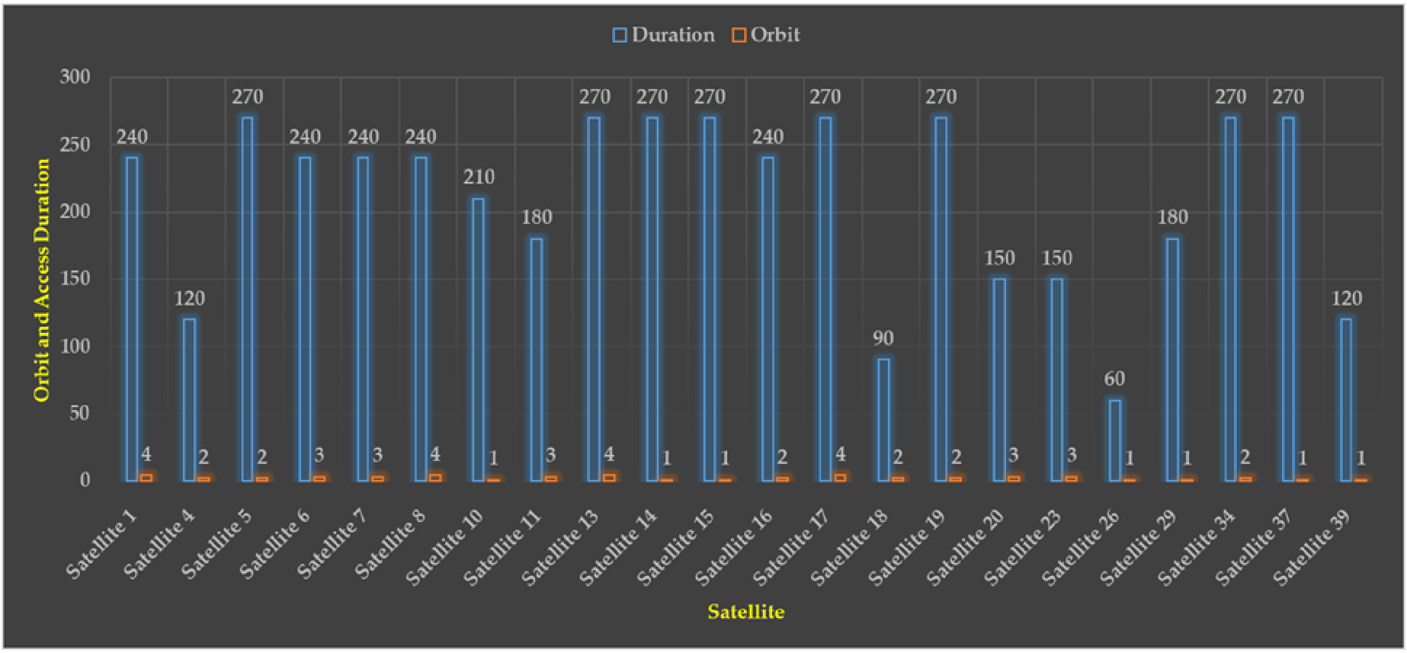


Figure 51: Australia satellite access duration with tracking and its orbit.

a) Australia

Due to its climate, topography, and vegetation, Australia is susceptible to large-scale wildfires due to the interaction of all three factors. In the most recent year, there have been multiple instances of wildfires breaking out. A region in New South Wales with a high risk of being affected by wildfires has been considered, and the analysis is currently taking place [299]. According to the analysis findings, the computed reconfiguration coverage for the Australian AOI is 95.9722%. This represents a coverage level that is nearly equivalent to real-time for the monitoring of catastrophic events. Examining Figure 52 and Figure 53, which depicts the system with access status being given for the (a) Nadir configuration and (b) the reconfiguration, which comprehends this result in a more clear and more concise manner. In the second scenario, the AOI is hidden from view for very short periods, which endows real-time/near real-time monitoring.

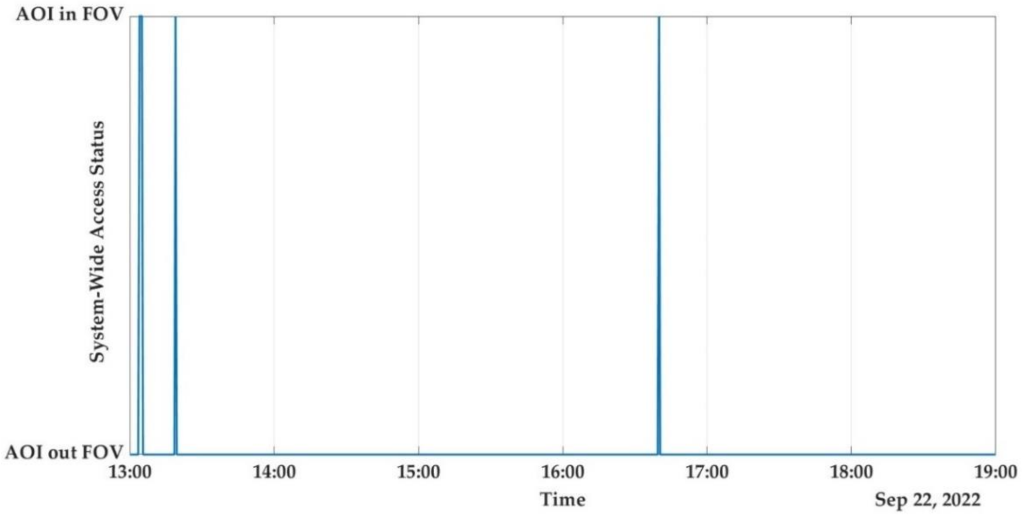


Figure 52: Australia system wide access status with Nadir configuration.

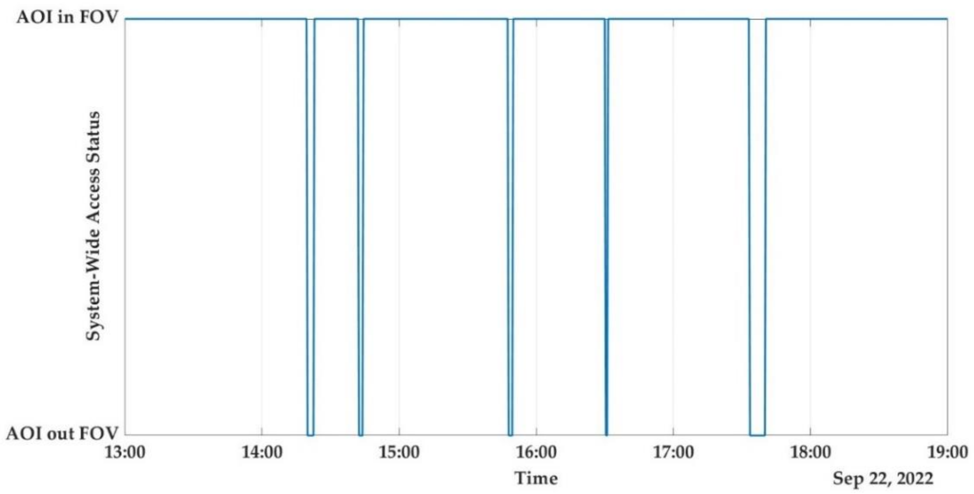


Figure 53: Australia system wide access status with reconfiguration.

b) Africa

In southern Africa, drier conditions have become more pronounced over the years, which has led to an increase in wildfire occurrence and extreme drought conditions. These conditions, either on their own or in combination, have led to a loss of crop productivity, the deaths of livestock and other wildlife, famine, the degradation of ecosystems, and a reduction in water quality and quantity. There is anticipated to be a 5.4% rise in the annual burned area throughout southern Africa in particular. These conditions, which are typical of southern Africa with their significant variations in rainfall and regular droughts, make the arid and semi-arid regions more prone to the outbreak of wildfires [300]. Considering all of these factors, Angola is factored into our analysis, and the results are shown in Figure 54 and Figure 55. In conclusion, there is a decrease in performance in the African site, which reflects a value for $SWAP_T$ that is 75.6944%. This is due to the geographical position of Angola in the globe (the distance between spacecrafts is maximum when close to the equator).

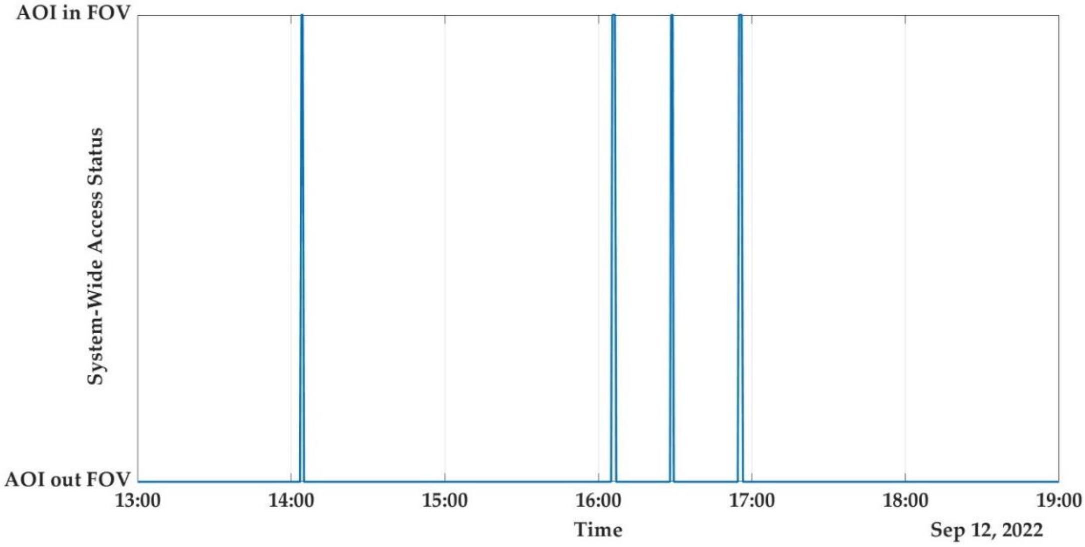


Figure 54: Africa system wide access status with Nadir configuration.

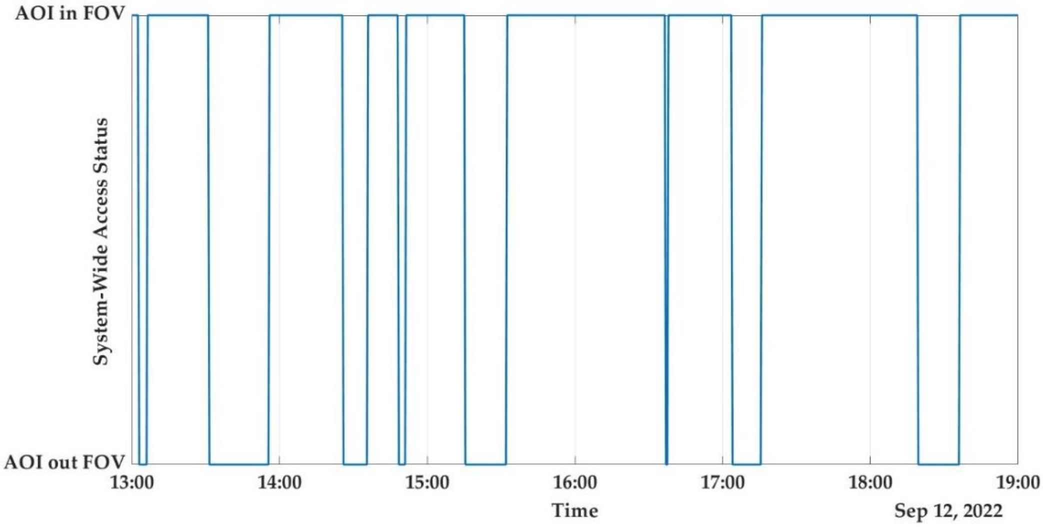


Figure 55: Africa system wide access status with reconfiguration.

c) Europe

In Europe, the island of Sardinia, located in Italy, is home to many urban interfaces, recreational values, and highly valued agricultural areas, all of which are in danger of being destroyed by severe wildfires due to the island's large population density. Individuals start most of the fires that occur on these islands and can be traced back to human negligence, agricultural and pastoral land use, and intentional arson. Based on the collected data from 1995 to 2009, the island of Sardinia has an annual average of 2219 fires, and the size of each fire is on average 7 ha. Each year, wildfires consume an area that is equivalent to 16,601 hectares on average, with the largest fire ever recorded consuming 9029 hectares. While fires larger than 50 ha make up only 1.8% of all fires (or about 40 per year), they are responsible for 68.7% of the total annual area that is burned [301, 302]. In accordance with the results of the simulation, as shown in Figure 56 and Figure 57, with the capability of reconfiguration, the $SWAP_T$ percentage is 98.889%, which provides real-time/near real-time monitoring over the region in the event of a wildfire is detected.

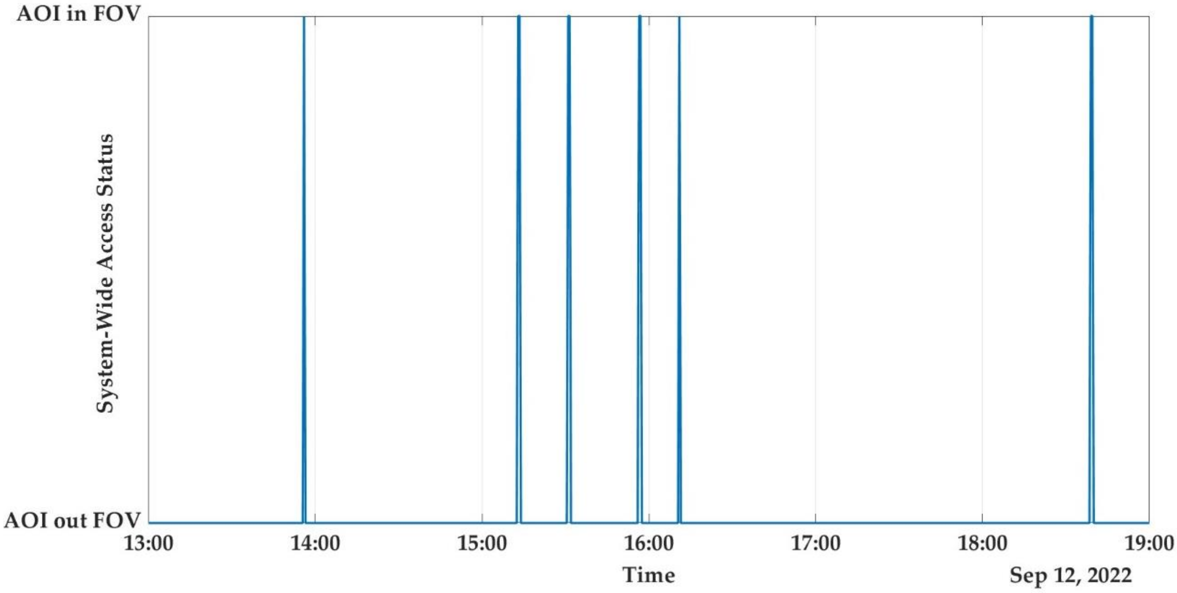


Figure 56: Italy system wide access status with Nadir configuration.

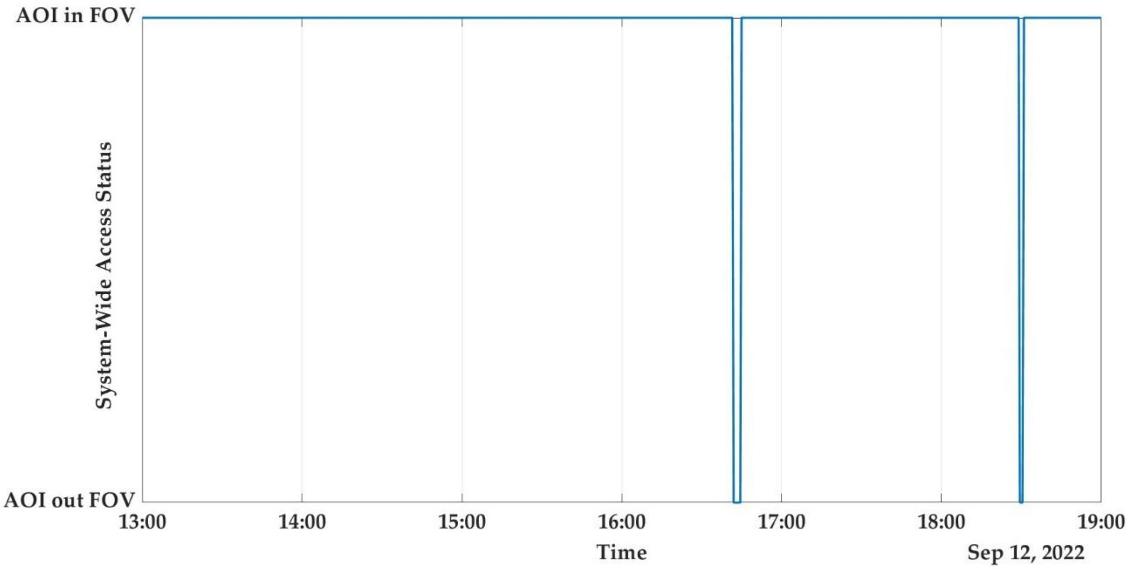


Figure 57: Italy system wide access status with reconfiguration.

d) North America

The western United States of America (USA) is facing a growing threat from wildfires, which are being fuelled by synergies between historical fire suppression efforts, shifting land use, insects and disease, and climate shifts that are becoming drier and warmer. In the United States, wildfires, which are the most significant form of natural disturbance in temperate forest ecosystems, affect an average of 4,500 km^2 each year. Communities and land managers in areas at risk of wildfire have an immediate need for mitigation strategies to lower the likelihood of wildfires and adaptation strategies to improve the resilience of ecosystems in the face of changing weather patterns and fire patterns. One of the regions that have been impacted is the rugged terrain of north-east Oregon, whose economies have traditionally been dependent on the region's forests and other natural resources. For the purpose of this research work, the Oregon region has been chosen, and after conducting the analysis, it was determined that the $SWAP_T$ is 97.0833% as shown in Figure 58 and Figure 59, which guarantees continuous coverage over that AOI.

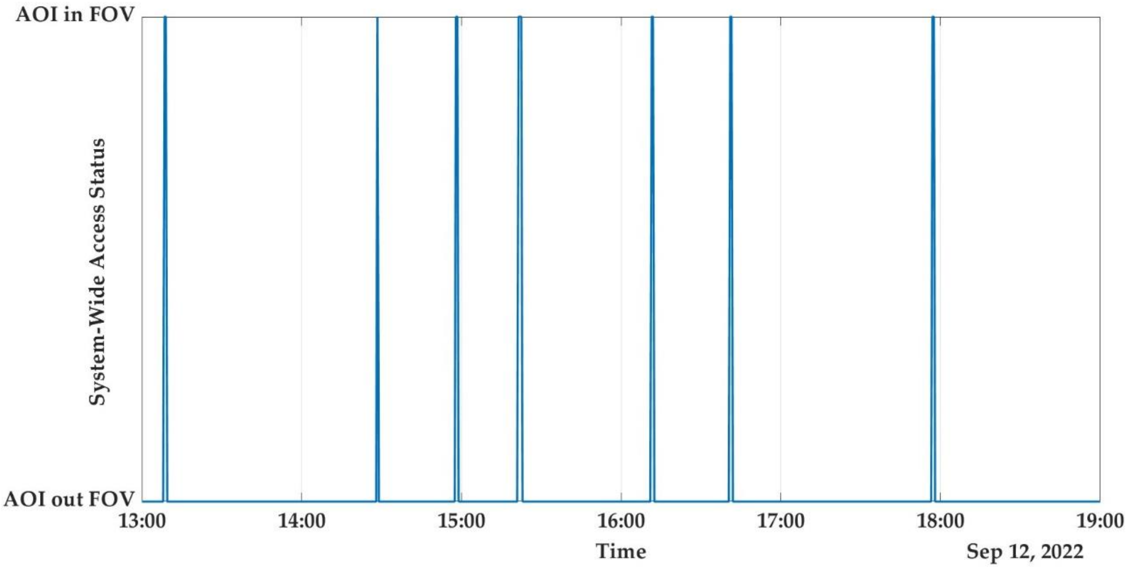


Figure 58: USA system wide access status with Nadir configuration.

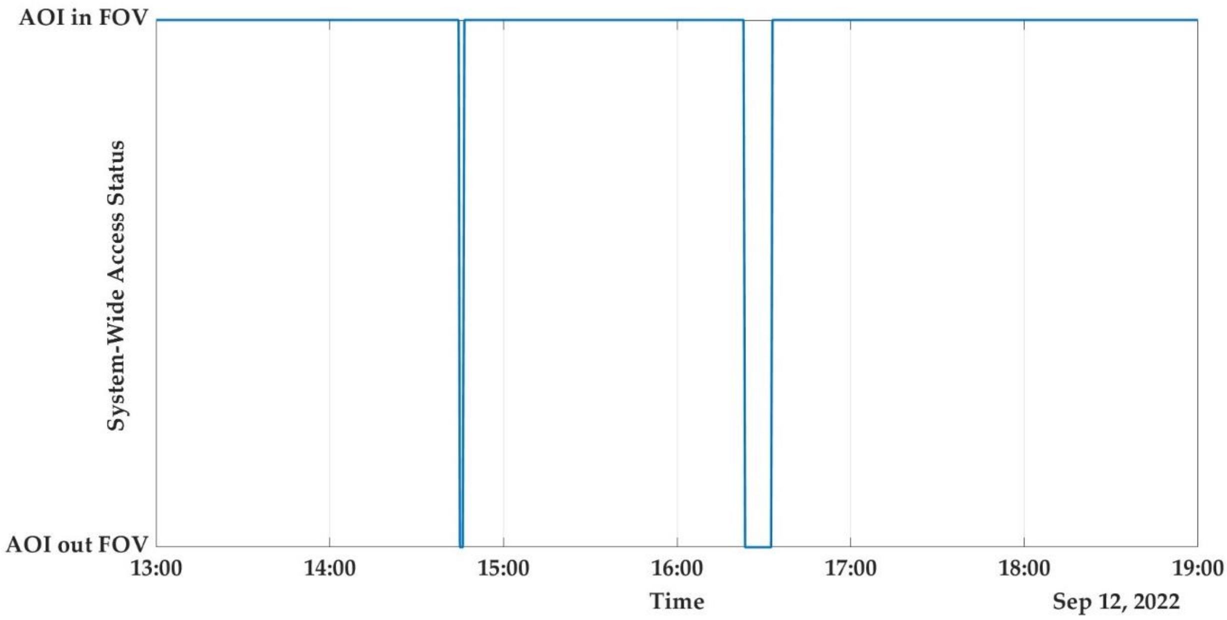


Figure 59: USA system wide access status with reconfiguration.

These findings suggest that astrionics, i.e., hardware accelerators for on-board edge computing, could be considered for future space missions. This would allow for the improvement of the framework, the efficient organisation of space-to-ground dataflow, and the provision of real-time or near real-time information, which could be very helpful in managing extreme events and humanitarian emergencies. It was discovered that the outcomes were influenced by the direction in which the sensors were pointed when the measurements were taken. These results can also be affected by the orbits of the satellite, the minimum EA of the AOI, the camera mounting position, and location in respect to the FOV of the satellite if the satellite is not continually pointed at the AOI. The orbits of the satellites can be altered by employing Keplerian parameters and by modifying to the appropriate AOI in accordance with the needs of the owner or operator. In the future, cameras will be able to be mounted on gimbals that really can rotate freely of the satellite, and the many sensors that are distributed across the constellation will be able to be used to improve the results. This not only makes it possible for the satellites to point directly downward, also known as Nadir pointing, but it also makes it possible for the gimbals to be adjusted so that they can track the AOI independently, and it makes it possible for heterogeneous sensors to provide valuable data at a diverse array of wavelengths.

4.7 Conclusion

The objective of this research work is to use CNN models for the analysis of hyperspectral data to examine the performance of astrionics. Australian bushfire investigation has been used as a working example, and input data has been taken into account for hyperspectral imagery obtained with PRISMA. Considering three distinct hardware accelerators—the Intel Movidius Myriad 2, the Nvidia Jetson Nano and the Nvidia Jetson TX2—demonstrated that the on-board application is possible in terms of both inference time and power consumption. These accelerators were employed to show that the on-board application is practicable. In line with other earlier works in the literature, this paper suggests the opportunity to investigate hardware accelerators for on-board edge computing in upcoming space missions in order to improve the services, better manage the entire space-to-ground dataflow, start providing real-time information, and enable TASO which could be really important in the event of disasters and extreme event management. Future research will evaluate other accelerators, including the FPGA and Google Coral TPU. The proposed method is evaluated in DSS for real-time disaster management and will increase the AOI coverage and decrease the revisit time. A Low Earth Orbit (LEO) iDSS constellation is suggested in this research study for real-time or near real-time wildfire monitoring. It has been shown that the on-board application is feasible in terms of inference time and power consumption. The proposed satellites feature hardware accelerators on board for edge computing, which is performed utilizing COTS components. This research work shows that real-time/near real-time monitoring is possible by altering the camera FOV, which is consistent with our earlier results. Since the iDSS is always connected through ISL, it is not necessary to always do active AOCS; instead, only when one of the constellation's satellites detects a wildfire this can communicate the other nearby satellites and perform active reconfiguration to collect as much data as possible. The results show that the proposed work can provide almost near real-time monitoring for Australia using the chosen constellation, which has a constellation system wide access percentage of 95.9722%. In order to enable the TASO in iDSS, an enhanced model using CNN will be embedded within this framework in future research. In order to improve the framework, efficiently organize space-to-ground dataflow, and provide real-time/near real-time information, which could be very helpful in disaster and extreme event management, this research also suggests that hardware accelerators for on-board edge computing can be considered for future space missions. It was observed that the results were affected by the direction in which the sensors were pointing. These outcomes are also affected by the satellite's orbits, AOI's minimum EA, camera mounting position and placement in relation to the satellite's FOV if they are not constantly pointing at the AOI. The satellites' orbits can be modified using Keplerian parameters and adjusting to the desired AOI based on owner/operator requirements. In the future, cameras can be put on gimbals that can rotate independently of the satellite, and the heterogeneous sensors in the constellation can be used to enhance the outcomes. This not only allows the satellites to look straight down, i.e., Nadir pointing but also allows the gimbals to be adjusted to track the AOI independently, as well as the heterogeneous sensors able to provide useful data at different wavelengths. Further, the applicability of iDSS for rare events in astronomy and astrophysics-based missions will be considered. Furthermore, heterogeneous satellites and sensors will be considered, and effective scheduling and planning will be considered on-board the satellite without the involvement of a human operator. Humans will play a supervisory role in the operation, shifting from human-in-the-loop to human-on-the-loop, enabling trusted autonomous operations. In future work, the resilience component of iDSS will be taken into consideration, i.e., if one satellite malfunctions, the others in the iDSS can reconfigure themselves to continue the mission and successfully accomplish it.

Chapter 5

Maritime Domain Awareness

The core focus of this chapter is the applicability of iDSS to maritime monitoring. Given that reliable and robust maritime security arrangements are essential for supporting Maritime Domain Awareness (MDA) and Intelligence, Surveillance, and Reconnaissance (ISR) activities. It is possible to do this with the help of satellite technology. The constellation of formations is proposed as a solution for this issue, and the autonomous orbital control of the proposed configuration is investigated and the findings are presented.

5.1 Research Background

A satellite technology-based approach becomes essential for countries in the southern hemisphere, with a sizeable maritime domain to protect in terms of sovereignty and sovereign rights, naval assets, infrastructure, resources and people [303-305]. These capabilities can significantly help with resource and biodiversity preservation, economic and environmental sustainability, disaster mitigation, and security at marine, as well as supporting sea safety and security [306]. This is especially true for isolated regions like Australasia [307, 308] and resource-rich regions such as the Gulf of Guinea [309], the South China Sea [310], Micronesia [311], the Argentine Sea [312], the Mediterranean Sea [313] and the Indian Ocean [314], to name but a few. According to the United Nations (UN), Illegal, Unreported and Unregulated (IUU) fishing significantly contributes to more than 90% of global fisheries stocks getting fully exploited, overexploited, or depleted, affecting regions most impacted by climate change. This practice also accounts for one-fifth of global fisheries catches, which can be worth up to $23.5 billion per year, making it the third most lucrative business natural resource crime after timber and mining [315]. For MDA, satellites can provide data for tracking ship movements, i.e., for ISR operations, and data for observing the marine environment, such as meteorological and oceanographic conditions. In 2014, the International Maritime Bureau (IMB) estimated that maritime piracy caused US$16 billion in economic losses annually, mainly due to theft, transportation delays, insurance costs, anti-piracy measures, etc. The DSS involves a set of small satellites working together that can simultaneously cover larger areas and outperform a single large (i.e., monolithic) satellite, which is often more expensive and less effective. DSS has many advantages, including easier design, faster build time, lower replacement costs and increased redundancy [22, 65, 259, 316]. One issue is to keep the formation geometry (required to accomplish the mission) while avoiding inadvertent collisions due to uncertainty in the state of the formation and/or failures. Recent research focuses on various control strategies to address these changes, including the possible adoption of AI techniques [24, 317]. The requirements of the application have been met with SAR. Figure 60 depicts a possible classification and examples of SAR satellite mission types [318, 319]. The satellite systems are classified into monolithic and DSS, and the latter is divided into several possible implementation branches as the constellation (flying far from each other, without relative navigation/control), formation flying (close flight, requires relative control) and other options as swarms or hybrid approaches [317].

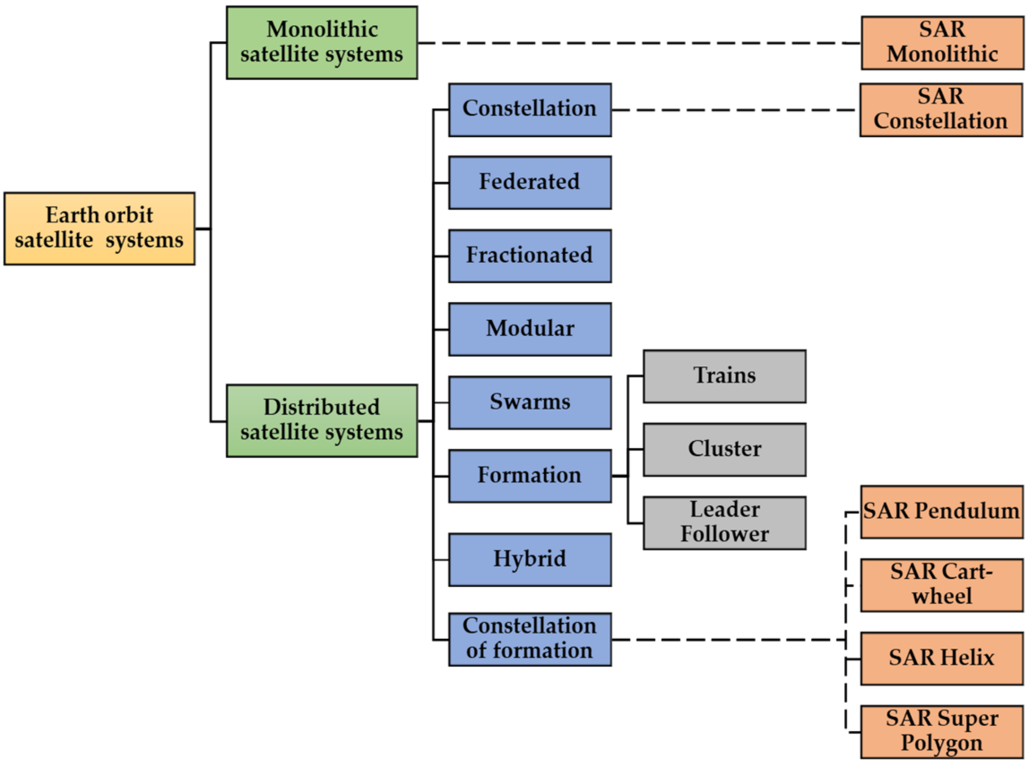


Figure 60: A possible satellite system classification with SAR satellite implementation examples.

In this research work, a *constellation of formations* is proposed to combine the benefits of the repeat cycle given by the constellation with the single-pass products allowed only by the formation flying distribution. An example of a monolithic SAR satellite was the Envisat [320], which provided a repeat cycle of 35 days. To perform interferometry, the product shall construct the interferogram from different acquisitions of the same scene, which in this case would be separated for 35 days. The constellation solution reduces the revisit time to a few days as the Satellite System for Emergency Management (SIASGE) system (Satélite Argentino de Observación COn Microondas (SAOCOM-1) and Cosmo-Skymed SAR constellations). However, this could not be enough for applications needing real-time generation of the interferogram. Further, real-time interferometry is required for the MDA application, which must be computed on every single pass over a target zone or AOI. This requirement may be derived from two primary motivations: the need for a fast determination of the interferogram, and the need for high coherence in the interferometry, in order to avoid artifacts caused by differences in the background of the scene due to the atmospheric changes or other effects not related to this specific application. Furthermore, as a new iDSS architecture type, a constellation of these formations with AI on-board for data processing is considered to keep this feature and reduce the revisit time. This also investigates the possibility of allocating control accelerations among satellites on each formation as a function of the formation objective (relative geometry) and the constellation objective (ground track repetition cycle period). he following contributions were made:

- A safe multi-baseline shifted-Helix Formation Flying is proposed for Along-Track Interferometric Synthetic Aperture Radar (AT-InSAR) Distributed Satellite System (iDSS), in the context of a Maritime Domain Awareness (MDA) mission over Australia.
- Autonomous orbital control is evaluated for reconfiguration and maintenance of this iDSS formation.
- For an increased revisit of maritime surveillance, a novel iDSS Archetype, "Constellation of Formations", is proposed, with an associated autonomous control law evaluated by simulations.

The objective of this work is not to define the constellation parameter but to propose a concept for its implementation (adding autonomy) by employing this two-level (constellation/formation) autonomous orbit control. Single pass interferometry is selected to maximise the coherence, which can be achieved by satellites flying very closely using a Formation Flying (FF) approach, as was implemented on TanDEM-X [321]. This pioneering mission generated new SAR products by defining the acquisition modes as a function of the relative orbits between both satellites, each of them having a complete SAR instrument. Satellite Formation Flying (SFF) is the coordination of multiple neighbouring satellites to accomplish an objective/goal stated in terms of the relative orbits between them. There are various formation flying mission configurations to satisfy the user requirements. Each configuration can be obtained by small changes in the orbital parameters of each deputy satellite with respect to the nominal parameters of the chief satellite. In order to meet the needs of the users, different configurations of formation flying missions have been proposed. SFF can be classified depending on the configuration, mode of operation, and other factors. Figure 61 shows a possible Formation Flying classification.

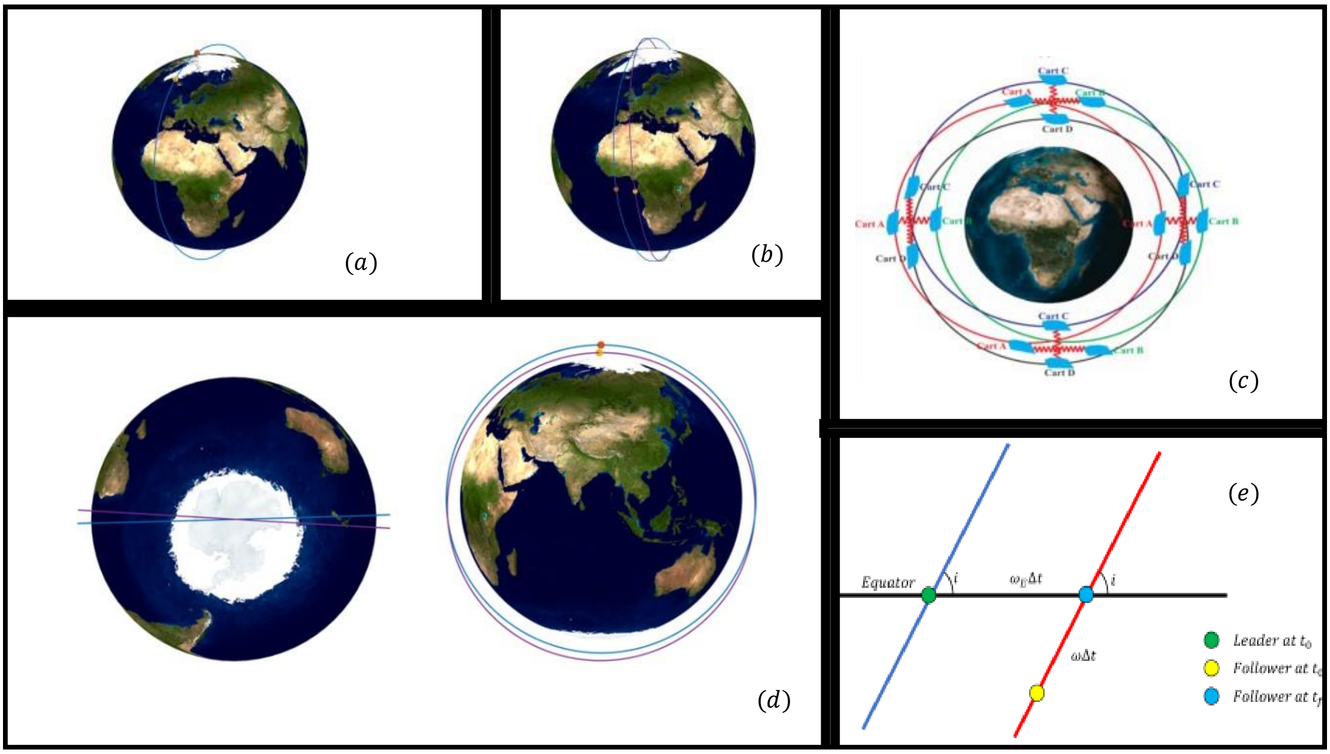


Figure 61: Chief-Deputy Classification (a) Leader-Follower, (b) Pendulum, (c) Cartwheel, (d) Helix Configuration, (e) Same-Ground Track.

5.2 Synthetic Aperture Radar

A satellite radar instrument produces and transmits its own energy using a known microwave signal pattern, then records the reception of that signal reflected back after interacting with the Earth's surface. When this instrument moves with a known velocity relative to the Earth's surface, the reflection also adds azimuthal information due to the Doppler frequency deviation and is referred to as Synthetic Aperture Radar data collection.

SAR data must be interpreted differently from optical images because the signal responds to surface characteristics like structure and wetness rather than being a static image. As compared to optical technology, SAR technologies can "see" through the darkness, clouds, and rain to notice trends in habitat, water and moisture levels, the consequences of natural or human disturbance, and variations in the Earth's surface as a result of quakes or sinkhole openings. These products are created by analysing the reflections of signals off a target location and measuring the two-way transit time back to the satellite, its frequency deviation, and the polarisation changes. The SAR interferometry technique "interferes" (differences) two SAR images of the same area, producing maps called interferograms that reveal ground-surface displacement (range change) here between two time periods. The phase differences are used to extract information about the captured objects (in comparison to a single image). As a result, at least one aspect ("Baseline") must differ between the images.

Future SAR missions will benefit from increased capability, reliability, and flexibility as a result of this spatial separation [321]. Applications for multistatic SAR systems include single-pass cross-track and along-track interferometry, spaceborne tomography, wide-swath imaging, resolution augmentation, ground-moving target acquisition, interference suppression and multistatic SAR imaging. Simultaneous data collection from numerous satellites reduces temporal and atmospheric disruptions, enhances performance, and allows the identification of rapid changes.

Along-Track Interferometric Synthetic Aperture Radar (AT-InSAR) systems are employed to estimate the radial velocity of targets moving on the ground by combining the interferometric phases, which are acquired by combining the two intricate SAR images obtained by two antennas spatially separated along the platform moving direction [322]. The AT-InSAR can be used in various applications such as monitoring real-time traffic management, ocean currents, coastal surveillance, ice drift, etc. In AT-InSAR, the baseline difference is an along-track distance, with a magnitude depending on the mission type, and determines the time difference associated with the pass of the satellite over the particular target, hence measured in seconds for single pass interferometry (two consecutive satellites looking at the same target) or days/years for multiple-pass interferometry (i.e., to process a stack of images taken on different passes over the same scene by the same satellite, other different satellites in a SAR constellation). Figure 62 shows different types of SAR in a simplified classification. The two main branches, interferometric and polarimetric, can also be combined as in the POLInSAR techniques [323].

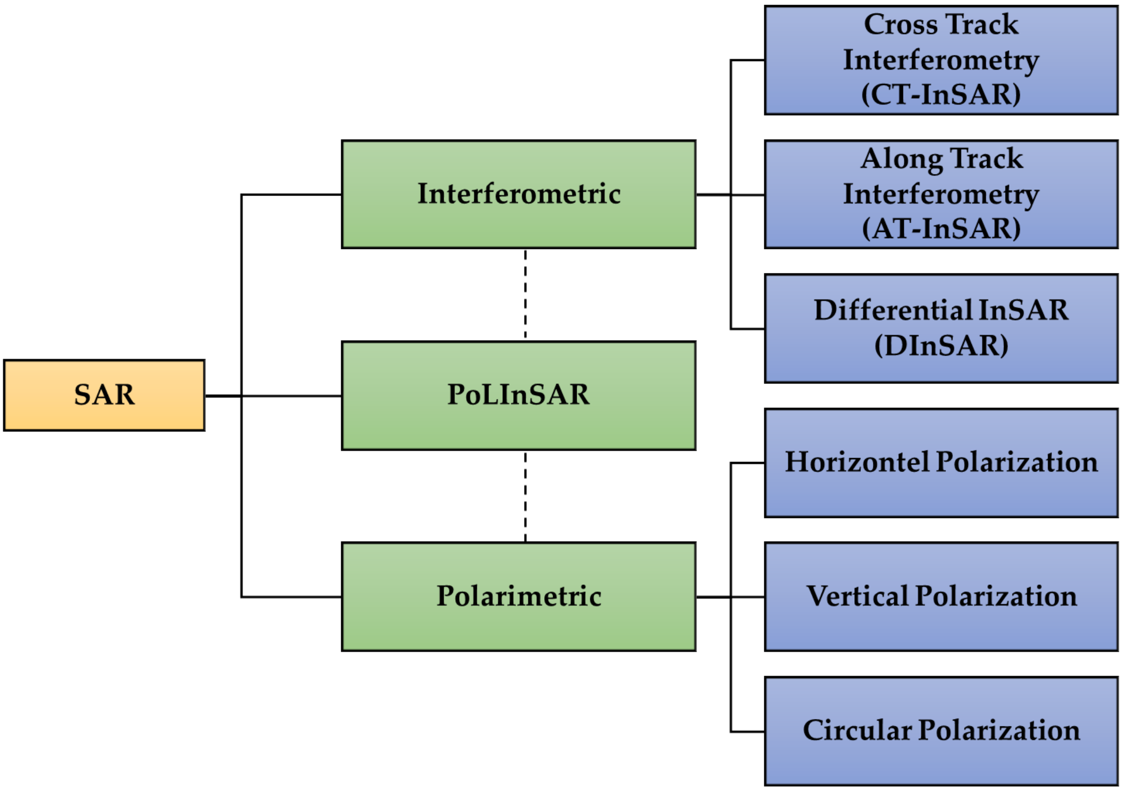


Figure 62: A simplified SAR classification.

The interferometric SAR missions can be implemented by multi-static SAR, which is characterised by their relative position or equivalent time, known as the baseline. This subfield of SAR categorisation will serve as the primary focus of the analysis. Some baseline types usually implemented on SAR interferometric missions are shown in

Table **20** and the baseline position difference is shown in Figure 63 for Along-Track interferometry. These baselines can be implemented in multiple passes, on which the interferogram is constructed with data of points of view obtained after several days when there is a repetition cycle on the ground track, or along a single pass when the multi-static SAR is composed by neighbouring satellites flying in formation. The latter may improve the SAR product in several ways, as the interferogram can be obtained in almost real-time.

Table 20: Some Baseline Types for Interferometric SAR [321].

Baseline	SAR type	Measurement and Application
$\boldsymbol{\Delta\varphi}$ *(Look angles difference)*	Across-Track	Topography, Digital Elevation Models
$\boldsymbol{\Delta t = ms, \ldots, s}$	Along-Track	Ocean currents, moving object detection
$\boldsymbol{\Delta t = days}$	Differential	Glacier/ice fields, lava flows, hydrology
$\boldsymbol{\Delta t = days, \ldots, years}$	Differential	Subsidence, seismic events, volcanic activities, crustal displacements
$\boldsymbol{\Delta t = ms, \ldots, years}$	Coherent Estimator	Sea-surface decorrelation times, land cover classification

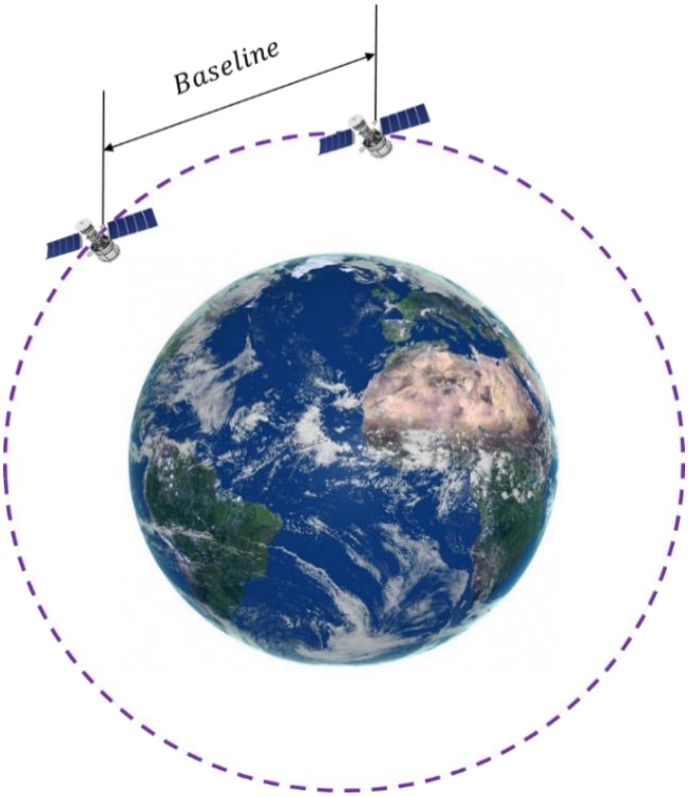


Figure 63: AT-InSAR Baseline difference.

In FF, several topologies exist to implement the relation between the satellites, for instance, the typical leader-follower approach. In this case, a Deputy satellite (also called follower or secondary) can choose from among the several formation geometries described previously to follow the chief, which can be reconfigured on-board [324]. The chief may also have an autonomous control objective to maintain the absolute orbit (for instance, drag-free), which the Deputy therefore follows. Beyond the topology, designers can see that there are objectives to follow the absolute orbit (for example, to follow a ground track), and other objectives is to follow a specific relative orbit geometry within the formation. A more general approach will be proposed to deal with these two types of objectives, to be presented as a Constellation of Formations. The type of formation useful for the MDA application is first taken into account, followed by simulation results using low-thrust continuous control for formation maintenance and reconfiguration.

5.3 Robust Multiscale AT-InSAR

As mentioned, the Along Track formation relies on precise control of the along-track separation to avoid collision between the leader (or chief) and follower. This entails a risk due to the typical drift between satellites in case the orbit control was not active for a period of time. On the other hand, the possible need for two different baseline scales simultaneously multiplies the risk, as there are now three possible collision events if only one more follower satellite is added to the configuration. D'Amico showed in [325] a description of a relative orbit in terms of Relative Orbital Elements $\delta\underline{\alpha} = (\delta a, \delta\lambda, \delta\underline{e}, \delta\underline{i})$, where $\delta\underline{e} = (\delta e_x, \delta e_y)$ is the relative eccentricity vector and $\delta\underline{i} = (\delta i_x, \delta i_y)$ is the relative inclination vector. The orbit phase difference is given by $\delta\lambda$, while the semi-major axes relative difference is δa. A safe formation is guaranteed when $\delta\underline{e}$ and $\delta\underline{i}$ are parallel or anti-parallel, for $\delta a = 0$. A strict AT-InSAR formation only has an orbit phase difference between satellites, thus it is not possible to guarantee the safe condition (as $\delta\underline{e} = \delta\underline{i} = \underline{0}$, here parallelism cannot be evaluated.). Here a variation of this along track formation is proposed and as follows:

1) To add a small helix component to each follower relative to the chief, where the across-track component is one order of magnitude smaller than the chief/follower along-track baseline.

2) To scale the chief/follower relative eccentricity and relative inclination vectors in order to generate low-risk "pipes" for each satellite.

Figure 64 shows a particular case of parallel relative eccentricity and relative inclination vectors for two followers with respect to the chief.

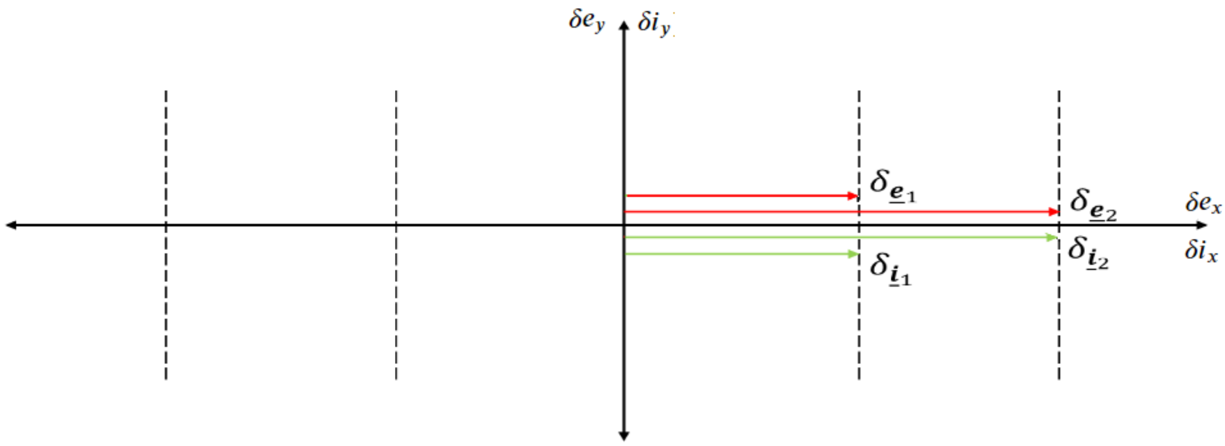


Figure 64: Example of relative eccentricity $\delta_{\underline{e}_i}$ (in red) and relative inclination $\delta_{\underline{i}_i}$ (in green) vectors of two followers $i = 1,2$.

Notice that the difference between the followers also preserves the relative eccentricity and relative inclinations vectors as collinear, therefore achieving a safe condition. This formation can include more followers by adding other scales on the same axis, preserving the collinearity between the relative eccentricity and inclination vectors for the given follower. On the other hand, different along-track baselines could be chosen for each of these followers. As the chief orbit is sun-synchronous and frozen, the satellite altitude and relative orbit baselines are guaranteed to repeat for the same latitude; hence the interferogram products generated on each pass have geometric coherence between different passes, enabling to perform differential interferometry by taking a set of images generated by the SAR system for the same zone. Moreover, the multiple along-track channels make it possible to track different velocity ranges for the targets on the Earth's surface, see [325], which can also be compared along the same pass by adding followers with different along-track baselines with the safe configuration previously presented.

The following equation, adapted from [325] (Equation 2.22), defines a metric δr_{rn}^{min} to evaluate the minimum distance, on the radial/normal plane, between two satellites in a formation, by using the Relative Orbital Elements, as follows, for a chief orbit with semi-major axis (a_c):

$$\delta r_{rn}^{min} = \frac{\sqrt{2} a_c \left|\delta \underline{e} \cdot \delta \underline{i}\right|}{\sqrt{\delta e^2 + \delta i^2 + \left|\delta \underline{e} + \delta \underline{i}\right| \cdot \left|\delta \underline{e} - \delta \underline{i}\right|}} \tag{17}$$

where $\delta e = \left|\delta \underline{e}\right|$ and $\delta i = \left|\delta \underline{i}\right|$. The following section will use this metric to evaluate the stationary regime after reconfiguration using an autonomous orbit control.

5.4 Autonomous Orbit Control

The Relative Orbital Elements are used by an autonomous feedback orbit control law derived in [324]. This control law guarantees a bounded control acceleration expressed in the Radial, Transverse, Normal (RTN) frame. This section examines the reconfiguration between the unsafe along track formation and the safe one proposed in the previous section. Figure 65 shows the simulation results for two followers after a reconfiguration manoeuvre starting with a pure along track (unsafe) condition. Figure 65 (a) shows the eccentric vector $\delta\underline{e}$ and the inclination vector $\delta\underline{i}$ components, which achieves a final state close to the pattern defined in Figure 64. Figure 65 (b) shows the evaluation of the radial/normal minimum distance metric δr_{rn}^{min}. Figure 65 (c) shows the along-track separation, which can be defined dynamically for each of the followers. As there is a small Helix component added to the along-track formation, there will also be an oscillation on the along-track distance, whose amplitude doubles the amplitude of the radial/normal component. The proposed geometry only sketches the idea of the relative geometry, while the application should give the definition of the parameters and can be changed dynamically using autonomous orbit control. This section examines the reconfiguration between the unsafe along-track formation and the safe one proposed in the previous section.

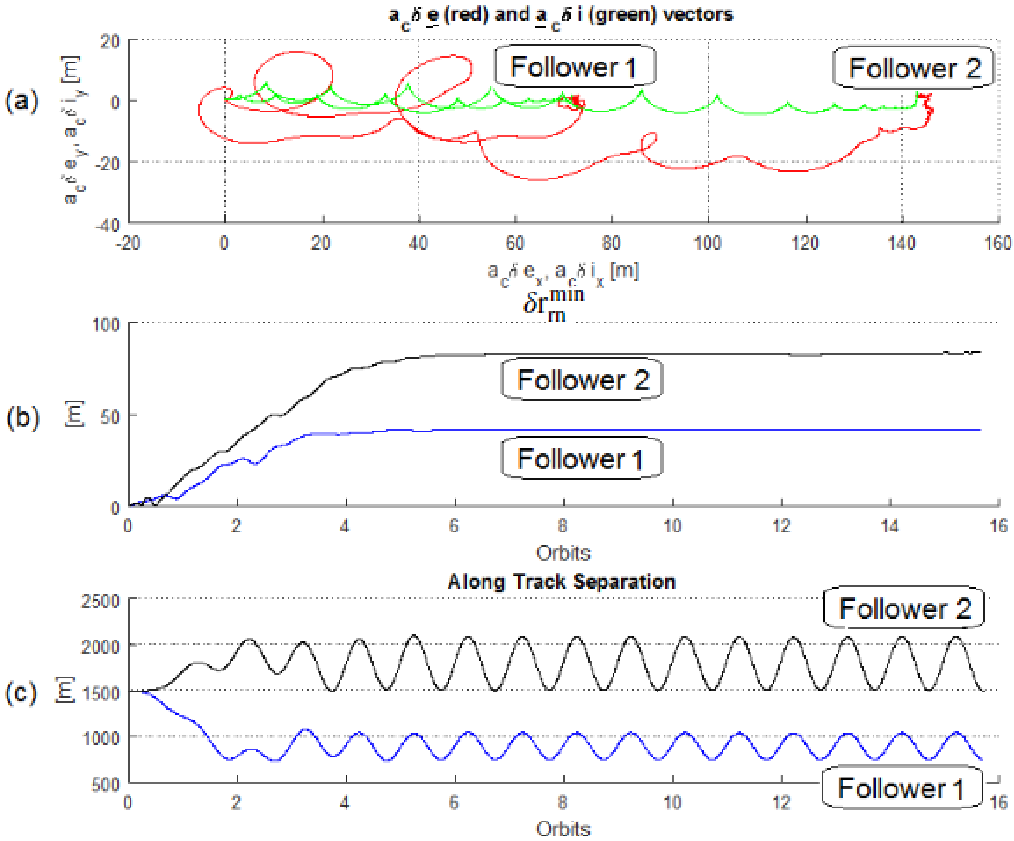


Figure 65: Simulation of a Robust Multibaseline AT-InSAR formation. (a) shows the components of $\delta\underline{e}$ and $\delta\underline{i}$, (b) shows the $\delta\underline{r}_{rn}^{min}$ and (c) shows the along track separation between each follower and the chief.

Figure 66 shows the result for follower 1 with an orbit control period of 20 seconds, and assuming here ideal orbit navigation. For a satellite mass of 100kg, the simulated thrust bound would be 1mN in all directions. Figure 67 shows the coordinates given by the difference between relative perigee angle φ and relative ascending node angle θ, with respect to the norms of the relative eccentricity $\delta\underline{e}$ and relative inclination $\delta\underline{i}$ vectors, scaled by the chief's semi-major axis a_c [325]. Notice that the beginning of the trajectory is at the origin of δe and δi , and thus it is not under a safe condition, while at the end, the difference $\phi - \theta$ is nearly zero (i.e., the relative eccentricity and inclination vectors are collinear) while the stationary relative eccentricity and inclination norms are approximately 40 meters when scaled by the semi-major axis. Therefore, the reconfiguration achieves the desired baseline and the required safe condition.

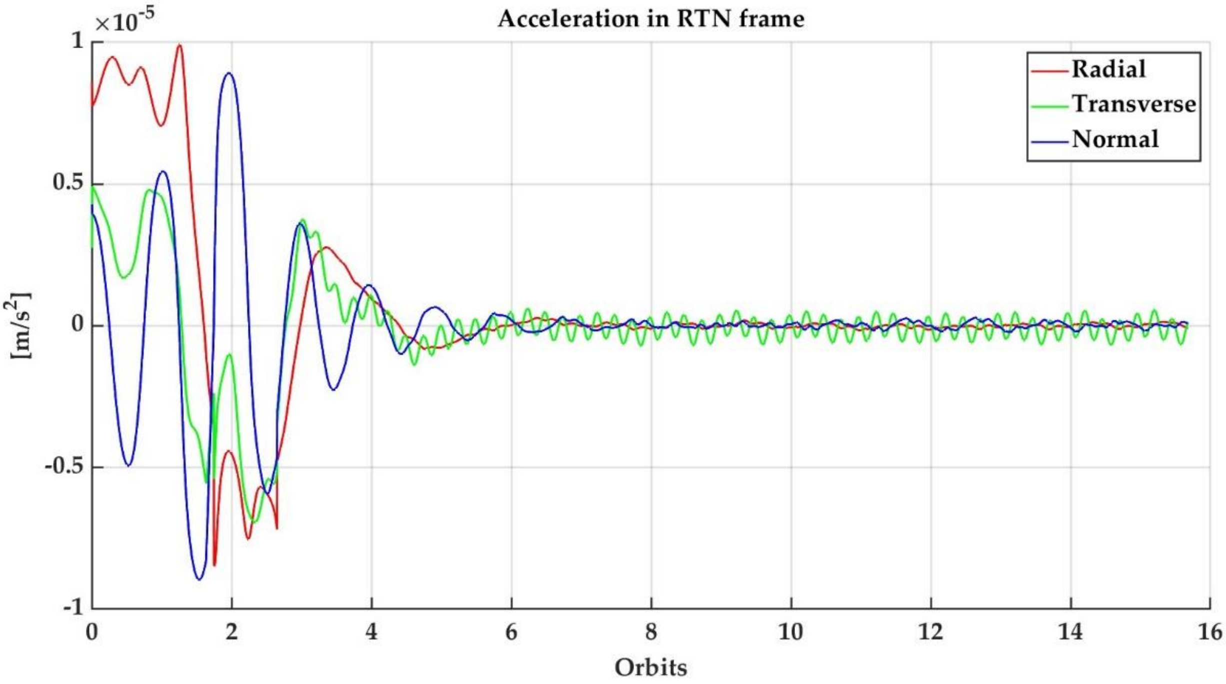


Figure 66: Control acceleration for the follower 1 - chief formation, where red is the Radial, green is the Transverse and blue is the Normal component.

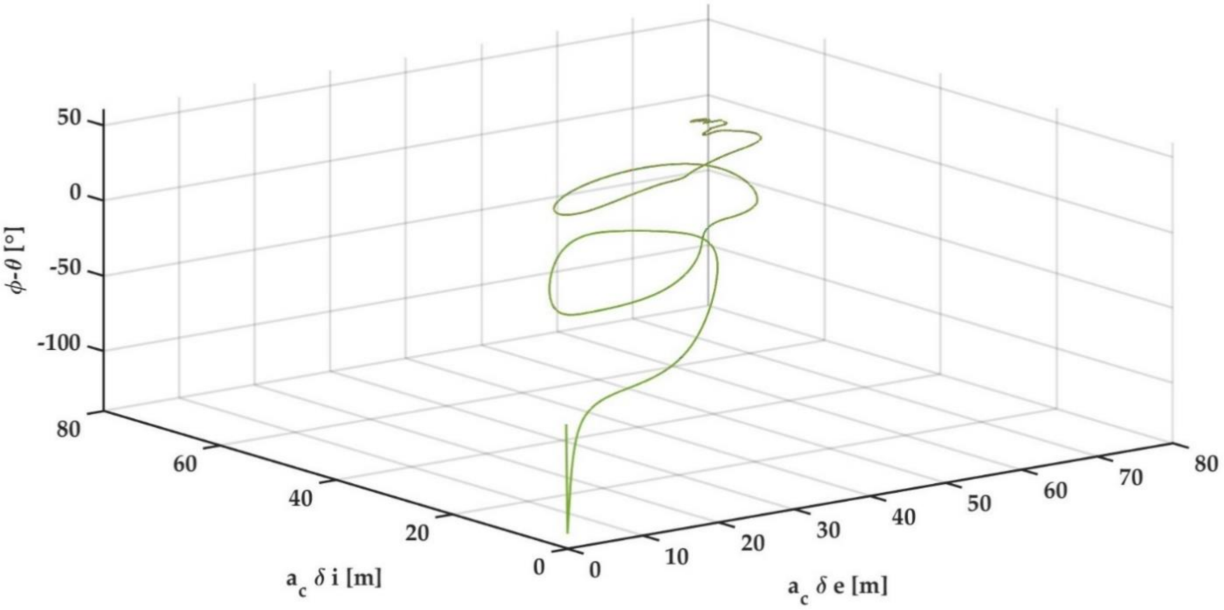


Figure 67: Safe condition evaluation for the follower 1 – chief formation.

5.5 Constellation of Formations

The formation presented in the previous section has a ground-track repeatability described by the repeatability of any of the ground tracks of the satellites in the formation. These ground tracks limit the observation coverage for a given instrument's field of view. A constellation of satellite formations is proposed to increase the repeatability and coverage of the desired iDSS while keeping the formation advantages. Figure 68 shows an example of two formations flying in a constellation. Notice that each formation preserves the single-pass SAR interferometry objective. On the other hand, the resulting constellation can be designed by using the centre of mass of each formation as the constellation-equivalent satellite.

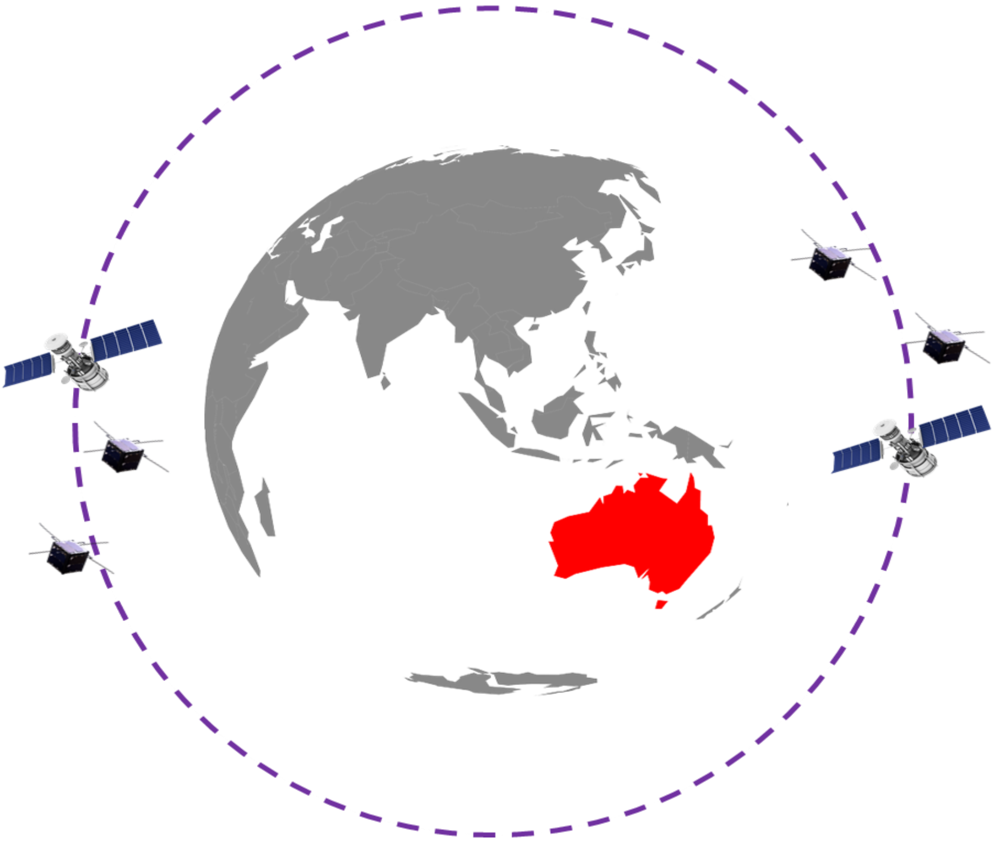


Figure 68: A constellation of two formations with three satellites.

Let the constellation $C = \{F_1, F_2, \dots, F_{Nc}\}$ be a set of N_c formations F_i, each of them composed by N_{F_i} satellites, i.e. $F_1 = \left\{s_{1,1},\ s_{1,2},\ \dots,\ s_{1,N_{F_1}}\right\}$, $F_2 = \left\{s_{2,1},\ s_{2,2},\ \dots,\ s_{2,N_{F_2}}\right\}$, etc. As each satellite $s_{i,j}$ belongs to the formation F_i and to the constellation C, there are at least two objectives for the orbit controller:

O_F) To keep the relative orbit of the satellite $s_{i,j}$ within the formation F_i.

O_C) To keep the formation F_i in the constellation C.

The formation objective O_F has been treated in the previous section by using the relative orbital elements to describe the feedback error. However, there is flexibility in implementing the control law that designers can now use. For any given acceleration control determined by the follower relative orbit feedback law, designers can implement it on the follower alone, on the chief satellite alone, or on both. In this case, the freedom is used to implement the formation orbit control laws in such a way that *it does not modify the dynamics of the formation's centre of mass*.

The constellation objective O_C is stated considering the absolute orbit elements as generated by the constellation design, which is typically free of non-conservative forces, i.e., without drag and solar pressure effects, and with a certain reduced order model of the gravitational effects. On the other hand, to build the constellation control error estimation of the centre of mass of each formation F_i is needed. This requires the knowledge of the centre of mass and the mass of each of the satellites but has the benefit of enabling a smoothing of the navigated orbit, as the centre of mass is a weighted average. In particular, if all the satellites have the same mass and independent identical navigation error distributions, the navigated position and velocity of the centre of mass will have their standard deviation reduced by a factor of $\sqrt{N_{Fi}}$. On the other hand, if there is a satellite dominant in mass, the centre of mass navigation error is dominated by the navigation error of this satellite. In this way, the constellation objective has been reduced to the problem of the control of the centre of mass of the systems or particles determined by each formation. Once the control acceleration for the centre of mass is computed, this is translated into the specific forces to be implemented on each satellite, considering their masses. A high-level control system architecture of satellite is shown in Figure 69. Additional features of this proposal are developed in the following sections.

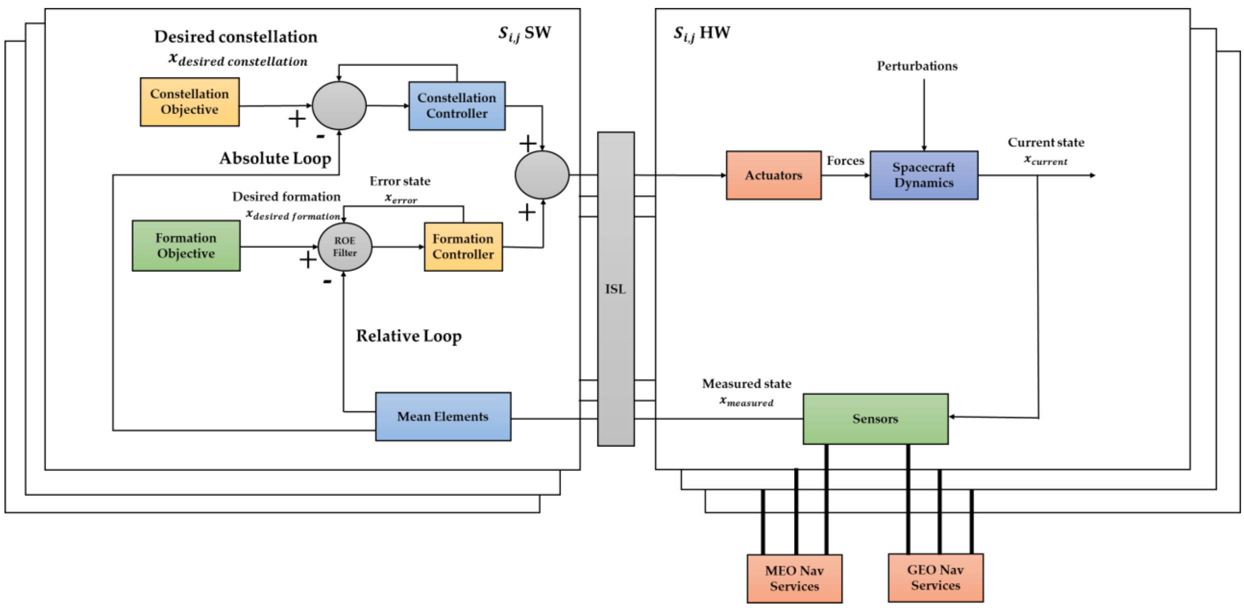


Figure 69: Spacecraft control system high-level architecture.

5.5.1. Dedicated Navigation

It is well known that the relative navigation based on Global Navigation Satellite System (GNSS) receivers can be improved by using interferometry in the L-band by means of the carrier phase (see [326]), which is known as Real-Time Kinematics (RTK). Other sensors can also be used to improve the accuracy of relative navigation, which confirms the benefit of implementing specific feedback for the formation control separated from the absolute control. On the other hand, the maintenance of the absolute orbit within the constellation must use absolute information, which can be implemented by using Precise Point Positioning (PPP) as proposed in [327].

The Relative Orbit Elements (ROE) for each of these objectives are computed using the Mean Orbit Elements (MOE) based on the Ustinov parameters and the analytic formulas as shown in [324] and the works of literature. However, this could not be enough to attain the high accuracy needed for autonomous orbit control, even using the PPP and RTK methods. To this end, a nonlinear filter with finite time memory can directly smooth the control error given by the Relative Orbital Elements, which is compatible with low thrusts, as shown in [327]. This filter can be applied by storing all the implemented control accelerations and measured Relative Orbital Elements, during a certain time horizon, for instance, the last (moving) orbit period. The resulting smoothed control error has enough accuracy to enable autonomous orbit control with a feasible propellant consumption (i.e., the navigation noise is not translated into a permanent actuation and waste of propellant).

5.5.2. Dedicated Control

Every satellite on the formation implements the same control computed for the centre of mass of this formation to preserve/achieve the constellation objective. This can be seen as a "common mode" control, using absolute orbit navigation of the centre of mass. On the other hand, for each satellite on the formation, there is an additional term obtained as the necessary feedback to implement the relative orbit control within the formation with the restriction that the dynamics of the centre of mass of the local formation is not perturbed. Following the previous analogy, this can be seen as a "differential mode", using relative navigation between the satellites on the same local formation. Consider the two followers and the chief in the previous section's application as a formation; thus, the control acceleration to achieve relative dynamics while preserving the formation's centre of mass must be computed. Let $\underline{a}_{0R}$, $\underline{a}_{1R}$ and $\underline{a}_{2R}$ be the relative terms of the control accelerations for the chief (index 0), follower 1 (index 1) and follower 2 (index 2). Because the SAR interferometry requirements are written in terms of the error between the chief and each of the followers, rather than the error between the followers, one can begin by stating the desired formation objectives:

$$-k_F\, \delta\underline{\alpha}_{01R} = B\left(\underline{a}_{1R} - \underline{a}_{0R}\right) \tag{18}$$

$$-k_F \cdot\, \delta\underline{\alpha}_{02R} = B\left(\underline{a}_{2R} - \underline{a}_{0R}\right) \tag{19}$$

where k_F is a proportional gain for the formation control and $\delta\underline{\alpha}_{01R}$ and $\delta\underline{\alpha}_{02R}$ are the relative orbital elements of each of the followers with respect to the desired Maneuver orbit (see [324]) written in both cases relative to the same chief:

$$\delta\underline{\alpha}_{01R} = T_0\left(\underline{\xi}_1 - \partial\underline{\xi}_1 - \underline{\xi}_0\right) \tag{20}$$

$$\delta\underline{a}_{02R} = T_0\left(\underline{\xi}_2 - \partial\underline{\xi}_2 - \underline{\xi}_0\right) \tag{21}$$

where $\underline{\xi}_0$, $\underline{\xi}_1$, $\underline{\xi}_2$ are the mean Ustinov parameters (see [14-15]) of the chief, follower 1 and follower 2 orbits respectively, while $\partial\underline{\xi}_1$ and $\partial\underline{\xi}_2$ are the desired deviation relative to the chief necessary to implement the mission orbit requirement for follower 1 and follower 2 respectively. The matrix T_0 is written in terms of the chief parameters and can be found in [323]]. Finally, the matrix B in (42)-(43) is the control input matrix of these relative orbit elements dynamics, which is assumed equal for all the satellites in the formation. These relative orbital elements dynamics are given as (see [324, 325]):

$$\frac{d\delta\underline{\alpha}_{01R}}{dt} = \underline{f}_{01R} + B\left(\underline{a}_{1R} - \underline{a}_{0R}\right) \tag{22}$$

$$\frac{d\delta\underline{\alpha}_{02R}}{dt} = \underline{f}_{02R} + B\left(\underline{a}_{2R} - \underline{a}_{0R}\right) \tag{23}$$

where $\underline{f}_{02R}$ and $\underline{f}_{02R}$ are considered very small disturbances, which can be partially compensated as a feed-forward term by the control law. The common input matrix B for a formation F_i will be determined by the orbit parameters of its centre of mass orbit, using its orbital elements as follows:

$$B = \frac{1}{\bar{a}\,\bar{n}} \begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ sin\,(\bar{\lambda}) & 2\cos\,(\bar{\lambda}) & 0 \\ -\cos\,(\bar{\lambda}) & 2sin\,(\bar{\lambda}) & 0 \\ 0 & 0 & \cos\,(\bar{\lambda}) \\ 0 & 0 & \cos\,(\bar{\lambda}) \end{bmatrix} \tag{24}$$

where $\bar{a}$, $\bar{n}$ and $\bar{\lambda}$ are the mean orbital elements of the centre of mass of the formation F_i, associated respectively with the semi-major axis, mean motion and mean argument of latitude. The columns of matrix B span the whole vector space R^6 every orbit, but locally only can generate a subspace of dimension 3. Therefore (42) and (43) cannot actually be met unless the left-hand sides belong to the column vector space of matrix B, but this can be solved in general by using the pseudo-inverse B^+ of the input matrix B :

$$\underline{a}_{1R} = \underline{a}_{0R} - k_F \cdot B^+ \cdot \delta\underline{\alpha}_{01R} \tag{25}$$

$$\underline{a}_{2R} = \underline{a}_{0R} - k_F \cdot B^+ \cdot \delta\underline{\alpha}_{02R} \tag{26}$$

The centre of mass constraint for the control accelerations is given as follows, for a chief with mass, and the followers with masses:

$$\underline{a}_{0R} \cdot m_0 + \underline{a}_{1R} \cdot m_1 + \underline{a}_{2R} \cdot m_2 = 0 \tag{27}$$

The linear equations (25)-(27) can be solved for the control vectors a_{0R}, a_{1R} and a_{2R} as follows:

$$\underline{a}_{1R} = -k_F \cdot B^+ \left(\frac{(m_0 + m_1)\,\delta\alpha_{01R} - m_2 \cdot \delta\alpha_{02R}}{m_0 + m_1 + m_2} \right) \tag{28}$$

$$a_{2R} = -k_F \cdot B^+ \left(\frac{(m_0 + m_2) \cdot \delta\alpha_{02R} - m_1\,\delta\alpha_{01R}}{m_0 + m_1 + m_2} \right) \tag{29}$$

$$a_{0R} = -k_F \cdot B^+ \left(\frac{m_1 \cdot (m_0 + m_1 - m_2) \cdot \delta\alpha_{01R} + m_2 \cdot (m_0 + m_2 - m_1) \cdot \delta\alpha_{02R}}{m_0 . (m_0 + m_1 + m_2)} \right) \tag{30}$$

This method can incorporate disturbance rejection, control saturation and fuel consumption management as made in [324], but the emphasis on the linear combination of the relative orbital elements is maintained. Notice that the control acceleration shown in the previous section's example for each of the followers did not specify the implementation completely, as there were undefined degrees of freedom. For instance, one could define zero relative control acceleration for the chief, as can be found in a non-cooperative leader-follower approach. Exploiting these degrees of freedom more generally allowing to preserve the centre of mass for relative control, and on the other hand, one can compute the control acceleration for the centre of mass in order to track the desired constellation objective as a common control acceleration $\underline{a}_C^{F_i}$ for a given formation F_i. Therefore, the total control accelerations to be implemented on each of the satellites of this formation F_i are as follows:

$$\underline{a}_0^{F_i} = \underline{a}_C^{F_i} + \underline{a}_{0R}^{F_i} \tag{31}$$

$$\underline{a}_1^{F_i} = \underline{a}_C^{F_i} + \underline{a}_{1R}^{F_i} \tag{32}$$

$$\underline{a}_2^{F_i} = \underline{a}_C^{F_i} + \underline{a}_{2R}^{F_i} \tag{33}$$

which is the control law for each formation F_i in the constellation of formations C. For a given objective for the centre of mass of the formation F_i in the constellation, it is defined as a relative orbital element $\delta\underline{\alpha}_C^{F_i}$ which determines the control term $\underline{a}_C^{F_i}$ as follows:

$$\underline{a}_C^{F_i} \;=\; -\,B_{Fi}^{+}\left(k_C \cdot \delta\underline{\alpha}_C^{F_i} + \underline{f}_C^{F_i}\right) \tag{34}$$

where the input matrix corresponds to the formation F_i which is explicitly stated in the notation. The term $\underline{f}_C^{F_i}$ may be used for feed-forward compensation of non-conservative dynamics, as aerodynamic drag or Solar radiation pressure, as a degree of freedom for the designer. In order to compute the constellation error, it is necessary to compute the desired orbit for the centre of mass, which can be performed on-board with a suitable orbit propagator, which should be modified/initialised considering the mission needs. As both control objectives O_C and O_F have different accuracy limits, the proposed separation helps to optimise the application of each of the laws on the specific time periods on which they may be more effective.

Note on the control law: In [65], several relative orbit control laws are formulated in terms of the linearised Clohessy-Wiltshire equation as:

$$\frac{d\underline{x}}{dt} \;=\; A^{cw}\,\underline{x} \;+\; B^{cw}\,\underline{u} \tag{35}$$

and the control is obtained as:

$$\underline{u} = -K(\underline{x})(\underline{x} - \underline{x}_d) \tag{36}$$

for a given desired coordinate $\underline{x}_d$. As the relation between the control and the error $(\underline{x} - \underline{x}_d)$ can be considered linear as in (55)-(56), the same approach can be implemented to determine the relative control component associated with the formation objective, restricted to determine a null deviation of the centre of the mass formation.

Note on the saturated control law: The thrust control authority must be selected to achieve the constellation objectives with a large enough margin. In this way, it is always possible to select a small enough gain k_F for the formation control which achieves stabilisation of the formation objective. As there might be time and propellant consumption restrictions, this gain and the thrust and satellite masses allocation in the formation should be selected carefully (see [324, 328]). In particular, the gain k_F could be selected specifically for each chief/follower pair in order to consider different features of each follower satellite and associated objective. However, to make the presentation simpler on (12)-(14), a unique gain k_F is chosen for all the followers.

5.5.3. Allocation of Satellite Masses on Each Formation

In the studies of a companion satellite for the L-band SAR Argentine MicroWave Observation Satellite SAOCOM mission [329, 330], the relation of masses between the chief and the follower was around ten times. It is reasonable to fix the same mass for the followers, i.e., $m_1 = m_2 = m_F$, and the chief mass is given as $m_0 = \beta m_F$ for $\beta \geq 1$. Moreover, it would be convenient to implement on the chief a thruster β times bigger in terms of force and propellant mass, for a given common propulsion technology and specific impulse. Under this mass model, the total mass of the constellation of formation would be $m_T = N_c (2 + \beta) m_F$, where N_c. is the number of formations of three satellites (one chief and two followers). The following particular cases can be identified by inspection of equations (28)-(30):

- $\beta \gg 1$: In this case, the required chief's control acceleration becomes negligible with respect to the control acceleration of the followers, which tends to be like a classical leader-follower topology on which the control is made by the follower only.
- $\beta = 1$: The required chief's control acceleration authority doubles the required control acceleration authority of each of the followers.

- $\beta = 2$: The required chief's control acceleration authority equals the required control acceleration authority of each of the followers.
- $\beta > 2$: The required chief's control acceleration authority is smaller than the required control acceleration authority of each of the followers.

In general, if there were N_{F_i} satellites on a formation F_i, the mass ratio for equal control acceleration authority for a chief with mass $m_0 = \beta m_F$ is given by $\beta = N_{F_i} - 1$. Moreover, notice that the case with $0 < \beta < 1$ would be feasible, but this case is not practical for a SAR formation, where the chief performs more tasks than the followers and thus requires more satellite mass.

Figure 70 and Figure 71 show the control acceleration evolution to implement the same formation reconfiguration as proposed for AT-InSAR, with $\beta = 2$ and $\beta = 10$ respectively. It is verified fact that for a larger value of β, the control authority required on the chief becomes reduced in comparison with the followers. This also has an impact on the DeltaV of each satellite, as shown in Figure 72. This could be taken for a trade-off on the specific constellation/formation system design under the particular restrictions and mission objectives, which is beyond the scope of this work.

Finally, notice that the implementation of this distributed control requires knowledge of the satellite masses, which are time-varying. One of the main uncertainties is knowing the satellite mass given by the propellant consumption. However, this becomes negligible by using very high specific impulse electric propulsion of several thousands of seconds. The feasibility of this specific impulse level can be verified with the Field Emission Electric Propulsion (FEEP) technology, which is available now as COTS products for small satellites (see [331, 332]).

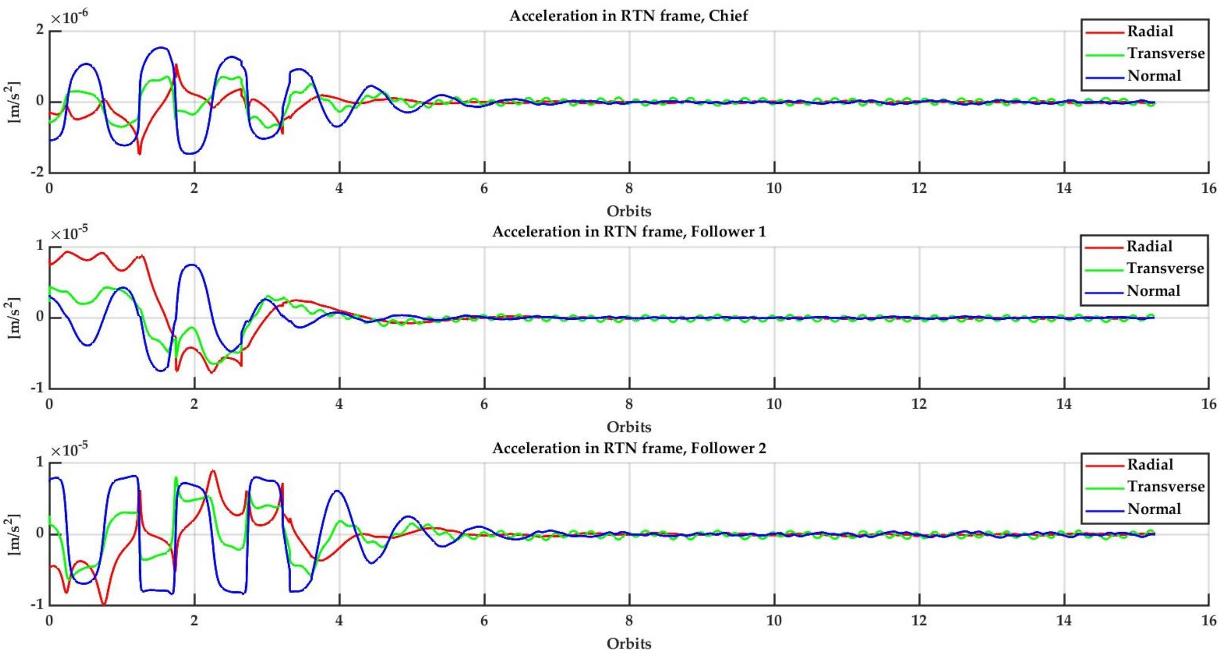


Figure 70: Control acceleration results with chief/follower mass ratio $\beta = 2$.

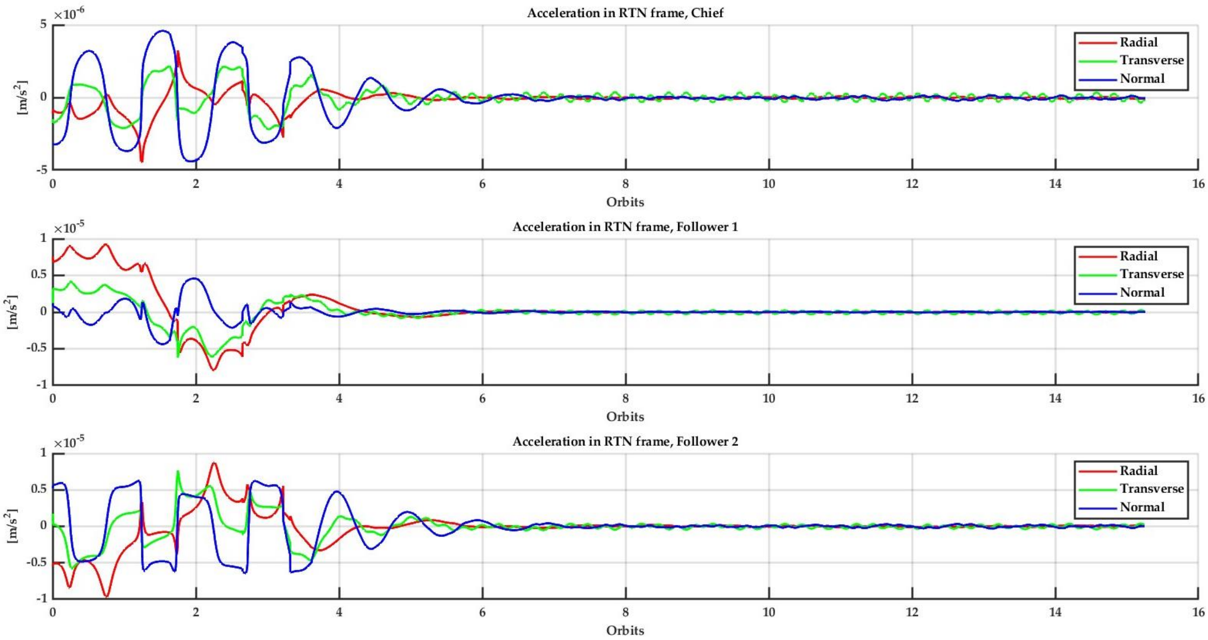


Figure 71: Control acceleration results with chief/follower mass ratio $\beta = 10$.

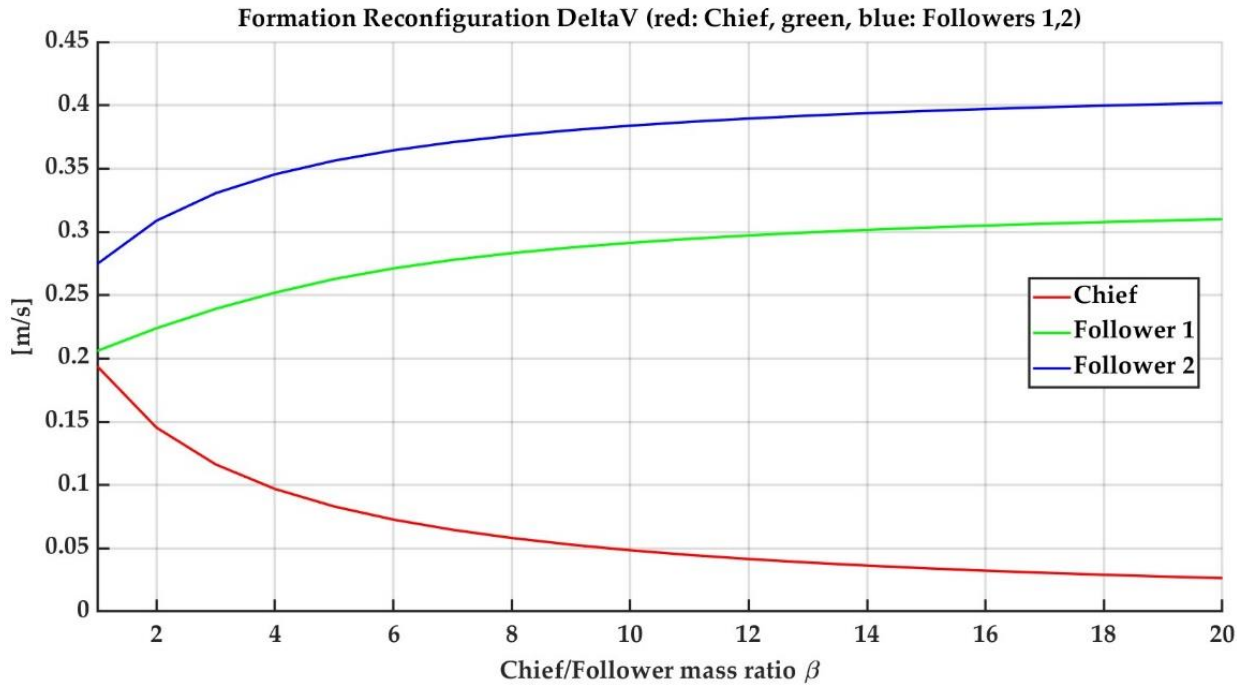


Figure 72: DeltaV on a formation reconfiguration as a function of the mass ratio β.

In order to simplify the implementation of the saturation, the saturation on the control acceleration differences is defined as follows:

$$\Delta \underline{a}_{1R} = sat\left(\underline{a}_{1R} - \underline{a}_{0R}\right) \tag{37}$$

$$\Delta \underline{a}_{2R} = sat\left(\underline{a}_{2R} - \underline{a}_{0R}\right) \tag{38}$$

Therefore, it can be found that under previous assumptions and two followers:

$$\underline{a}_{1R} = \Delta \underline{a}_{1R} \cdot \left(1 - \frac{1}{\beta+2}\right) + \Delta \underline{a}_{2R} \cdot \left(-\frac{1}{\beta+2}\right) \tag{39}$$

$$\underline{a}_{2R} = \Delta \underline{a}_{2R} \cdot \left(1 - \frac{1}{\beta+2}\right) + \Delta \underline{a}_{1R} \cdot \left(-\frac{1}{\beta+2}\right) \tag{40}$$

$$\underline{a}_{0R} = \frac{-1}{\beta+2}\left(\Delta \underline{a}_{1R} + \Delta \underline{a}_{2R}\right) \tag{41}$$

There is no real actuator saturation in (31) and (32), as the saturation is applied here to a difference between control accelerations on different satellites. However, one could use an estimate of the upper bounds on these maximum available differences, considering the margin to guarantee that the real actuator on the full expressions (25)-(27) does not reach any saturation limit.

5.5.4. Inter-Satellite Communications

The implementation of this control law requires communication between the satellites in each formation, i.e., ISL, in such a way that all the absolute positions are known at least by one of the satellites, while all of the satellites receive all the relative errors and the centre of mass acceleration command in order to implement the associated force. ISL allows for satellite-to-satellite communication on each iDSS formation F_i, and many possible implementations are shown by Liz Martinez et al. [333]. A direct solution is given by the Star topology, where the follower satellites of the formation communicate this navigation states to the chief, and hence this chief can broadcast this information to all the followers, including also its own navigation state and the relative navigation respect to each of the followers. Figure 44 shows this and other feasible topologies, with the full-duplex ISL being represented by double arrows. By including reactive components into the architecture, ISL allows the iDSS operations to be enhanced and data to be processed on-board the satellite for real-time operation. The ground station network and/or geostationary satellite service can be used to facilitate communications between satellites of different formations, which may be useful to perform constellation reconfigurations and process collision avoidance alarms from external objects [22]. As the communications become part of the control loop, a complete infrastructure to validate autonomous orbit control shall be able to emulate the inter-satellite links, as proposed in [334].

5.6 Conclusion

Autonomous orbit maintenance paves the way for TASO to become a reality in iDSS. Here it is shown that TASO is attainable with low-thrust electric propulsion for two main objectives: achieve and maintain the satellite orbit on the constellation, using absolute orbit navigation, and on the formation, using a more precise relative orbit navigation. In this way, each autonomous orbit control objective has a dedicated navigation type.

As a case study, iDSS mission for MDA is considered using distributed SAR instruments and demonstrated a formation geometry capable of tracking ship movements using single-pass AT-InSAR. In particular, a single pass and multi-baseline implementation was proposed using a safe three-satellite formation, which allows us to avoid temporal decorrelation and to have different velocity scales to track simultaneously. In order to improve the repeatability of these SAR products, a constellation of these formations is proposed. A demonstrated, this can be solved by autonomous orbit control using low thrusts compatible with electric propulsion. A particular formation mass distribution was analysed on which there is a chief whose mass is equal or greater than the mass of the followers by certain common factors. It was shown that for the combined formation/constellation control the relative importance of the control authority (in terms of manoeuvre total Delta V) of the chief decreases in relation of the equivalent figure for the followers, as this mass ratio increases.

A Constellation of Formations approach was proposed as a way to model the problem, and the solution's concept has been determined. The approach is based on the concept of a system of particles to describe each of the formations in the constellation in such a way that the relative control within the formation determines the formation flying, while there is a separate constellation control objective stated in terms of the centre of mass of each formation, i.e., a constellation of formation's centre of masses. Both objectives were solved with the same feedback control law structure using relative orbital elements obtained from the mean orbit elements of each of the spacecrafts. This requires an inter-satellite communication link between the satellites on the same formation for the formation flying feedback computation and the knowledge of the constellation objective in terms of mean orbital elements for the constellation feedback computation, which may be obtained on-board by the desired orbit propagation.

Finally, notice that with the recent evolution of inter-satellite communications, it is possible to augment this ISL capacity to share also the SAR data information to generate the interferogram on-board and, therefore, deliver it in near real-time to the user. In this way, as iDSS solutions become more readily available, a concept of operation with on-board single pass multibaseline interferometry computation will make it possible to deliver high-quality SAR data products faster to the user for effective maritime monitoring. Additional work must be addressed to perform more realistic simulations, including hardware in the loop to test GNSS navigation hardware, control nonlinearities, and possible inter-satellite links topologies, in order to complete the mission concept at the flight segment system level.

Chapter 6

Multidisciplinary Design and Optimisation of iDSS

This chapter focuses on the design and optimisation of the iDSS for EO specifically for Australia and Australasia, with the goal of providing persistent coverage over the region. The iDSS subsystem and mathematical model is presented and a Multidisciplinary Design Optimisation (MDO) is carried out to optimise the iDSS in terms of mass and coverage. The results are presented and discussed in this chapter.

6.1 MDO of iDSS

Outer space is dominated by small satellites, especially in LEO [192]. In an iDSS, several satellites or modules work collaboratively via ISL to achieve mission objectives using AI astrionics that are challenging for a single small satellite to achieve on its own. Previous research [193-196] focused on improving coverage and lowering costs by optimising the geometric arrangement of the satellite constellation, but the satellite subsystems and their characteristics were not considered in the optimisation. Additionally, iDSS platforms can be linked via ISL, allowing data sharing among the platforms. As demonstrated in Chapter 4, to ensure that only actionable and meaningful information is downlinked to the ground, this is accomplished by utilising cutting-edge astrionics with AI algorithms [217, 218]. An iDSS long-term success relies on significant budget reduction, which is possible only when the interconnection between the constellation and satellite subsystems is exploited to its full potential. As a result, the iDSS design that considers all disciplines with interdisciplinary interaction must be optimised. Due to its disciplinary restrictions, iDSS becomes a MDO and cannot be solved as a conventional Non-Linear Programming (NLP) problem [197, 198]. When solving a multidisciplinary design problem, it is important to examine the system's design for each discipline and how those disciplines interact with one another. MDO is a branch of engineering that deals with optimisation problems to determine the ideal solution in a specific design space while considering the constraints [197, 199-202]. When integrated, the optimisation of distinct subsystems may conflict with one another. As a result, the whole system must be optimised in its entirety. In an iDSS, modelling subsystem interactions complicate the optimisation problem since subsystem compatibility must be preserved while the objective function is minimised [197]. Martin et al., [201] described various MDO architectures and classified Hierarchical and Non-hierarchical MDO architectures. In a hierarchical architecture, each child element solely interacts with a parent element. However, in non-hierarchical structures, besides the parent-child interaction, there are other significant interactions among the child elements [203]. Since there are substantial interactions among some of the satellite subsystems, a non-hierarchical strategy is required in the current scenario. Depending on the problem formulation, non-hierarchical architectures are further divided into the Monolithic and Distributed categories. Monolithic architecture is formulated as a single optimisation function. Whereas, in distributed architecture, the problem is split into subproblems and reassembled to produce a combined solution.

In recent decades, several efforts have been made to compare different MDO architectures, and a conclusion has been established based on the specific situation and can be found in the literature [203-211]. These studies suggest that the success of architecture depends on the task at hand, and there is no such architecture ideal for all the applications. The research findings reveal that the chosen MDO architecture impacts the solution's optimality and the processing resources required. iDSS design optimisation using MDO techniques is seldom discussed in the literature.

OpenMDAO [212], a specialised framework for MDO optimisation, is used to represent multiple disciplines of iDSS and their interconnections. The use of such a framework eliminates most of the human factors in architecture programming. Therefore, the results are unbiased. The following are the main contributions:

- Design of an iDSS Constellation for an Australia/Australasia-specific EO mission for persistent coverage.
- The proposed MDO is discussed in terms of constraints, then optimised in terms of mass and coverage modified to be consistent with the iDSS design.

The chapter begins with a thorough analysis of the iDSS design challenge that defines the objectives and links across disciplines in section 6.2. Each iDSS subsystem and its corresponding analytical model are presented in Section 6.3. Section 6.4 provides the background for the iDSS design challenge for the examination of Multidisciplinary Feasible

(MDF) architecture. Results and analysis are presented done in Section 6.5, followed by the conclusion in section 6.6 where the optimisation outcomes for the MDO-MDF architectures investigated in this research are presented.

6.2 Problem Description

This research mainly focuses on designing and optimising an iDSS for persistent coverage over Australia. iDSS is a multidisciplinary system that includes the disciplines, as shown in Figure 73.

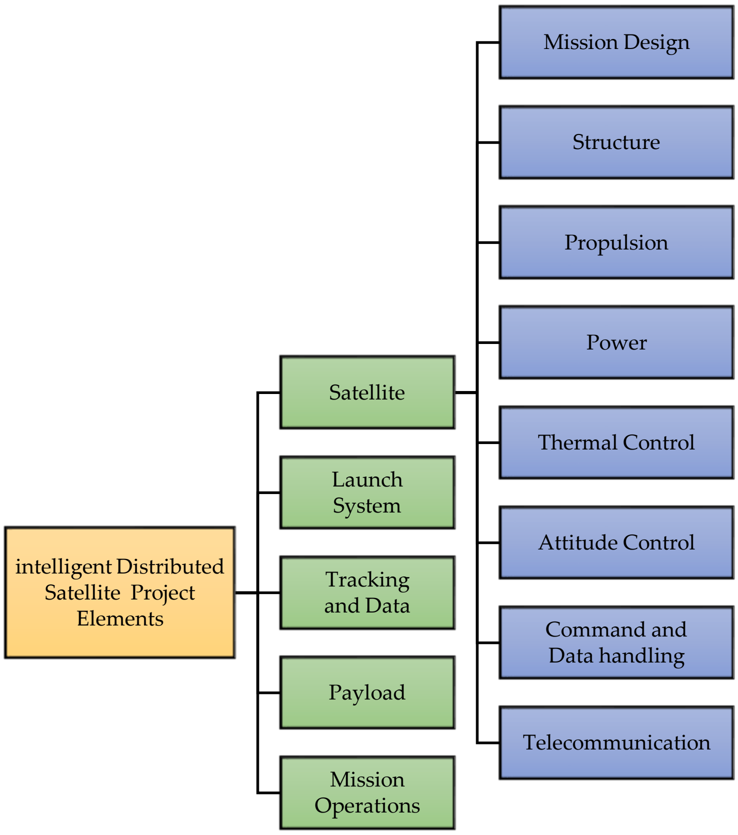


Figure 73: iDSS Project Elements.

The primary need for iDSS is to provide a more responsive and resilient option to address the growing needs of Australia. Only the Constellation, Power, Thermal, Structure and Payload subsystems are considered in this initial analysis. The attitude control, command and data handling, Telecommunication, and Propulsion subsystems are assumed to be readily available, and employing design estimation relationships, their mass and power budgets are assessed [215, 216]. The other elements, such as the Launch system, Tracking and data, and mission operations will be considered in future research. The MDO optimisation problem necessitates a set of design variables and subsystem inputs to produce subsystem states through the processing of the corresponding analysis model. The calculated subsystem states are either required for calculating the objective/constraints or by other subsystems (coupling). Figure 74 depicts the relationships between the disciplines of the modelled iDSS. The Telemetry, Tracking, and Control (TT&C) component of a spacecraft serves as a link between the spacecraft in the iDSS and the facilities the link to the ground. On-board Data Handling (OBDH) is responsible for processing the data on the satellite itself.

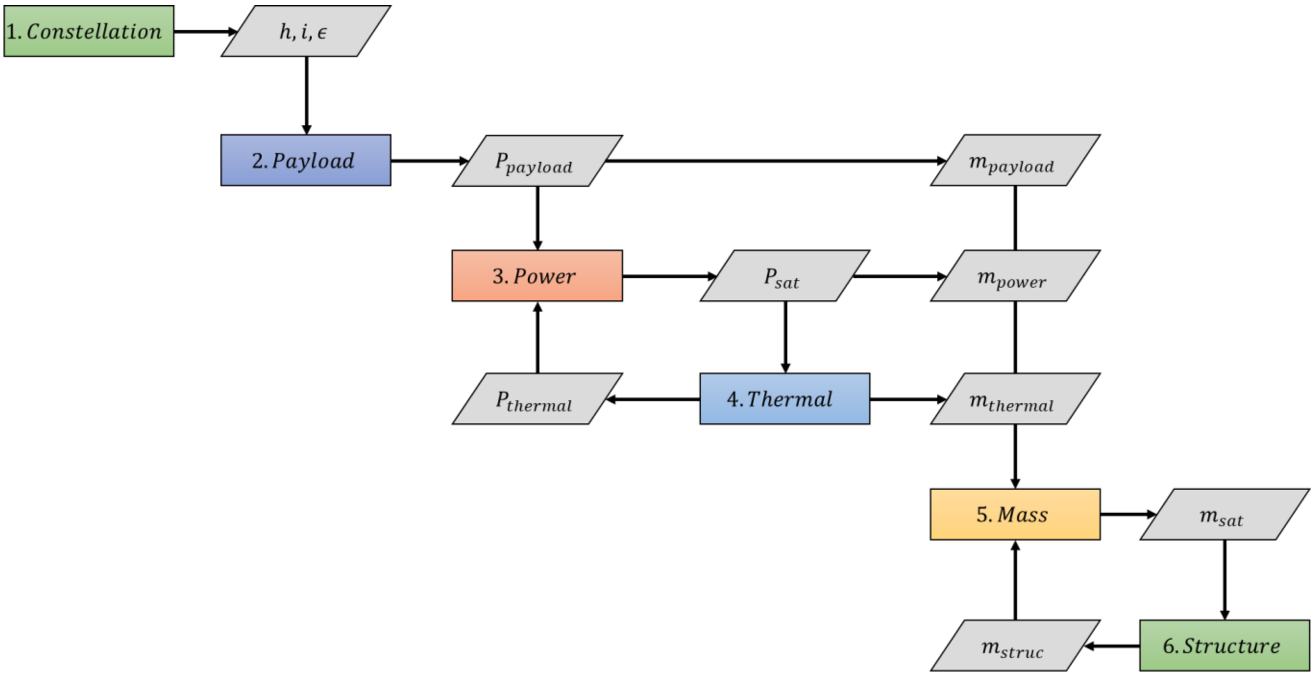


Figure 74: iDSS Subsystem couplings.

The handling of coupling variables is the distinction between the different MDO architectures. Although most of the design variables in the current problem are continuous, some of them are discrete. The optimisation problem changes to a Mixed Integer Non-Linear Problem (MINLP) when both types of variables are considered [219], whereas MINLP is extremely challenging to solve. Launch costs directly impact the number of satellites and orbital planes, which are independent of another subsystem. These discrete factors will simultaneously impact the final satellite design and computation time. For MINLP to a Non-Linear Problem Reduction (NLPR) [220], the number of satellites and orbital planes are fixed and solely optimises the elevation, inclination, and altitude angles in the context of constellations.

This research work proposes a constellation of LEO satellites [37, 47-49, 52, 221] for EO [222]. The constellation is made up of 25 satellites that are uniformly spaced across five orbital planes. All the participants in the proposed constellation are assumed to be similar and carry the same optical payload. In this iDSS optimisation problem, the major challenge is to offer full coverage over Australia. The optimisation challenge aims to reduce the overall mass of the iDSS system while keeping a variety of limitations in mind. The following equation is used to compute the mass m_{sys} of iDSS.

$$m_{sys} = N_S * (m_{struc} + m_{payload} + m_{power} + m_{thermal} + m_{remaining}) \tag{42}$$

where N_s denotes the number of satellites, m_{struc} denotes the structural mass of the spacecraft, $m_{payload}$ denotes the Payload mass, m_{power} denotes the mass of the power subsystem, $m_{thermal}$ denotes the mass of the thermal subsystem, and $m_{remaining}$ denotes the mass of the subsystems other than the ones mentioned above.

6.3 iDSS Subsystems

The mathematical framework for each discipline considered for the iDSS is presented in this section. The iDSS represents the parent element, whereas the constellation model, payload, thermal, power and structure represent the children's elements.

6.3.1 Constellation Model

Constellation Model refers to a group of satellites whose orbital parameters ensure an orderly sequence of handovers to obtain the desired coverage. Different types of coverage are available. a. Global, b. Zonal (Latitude belt), and c. Regional, as illustrated in Figure 75.

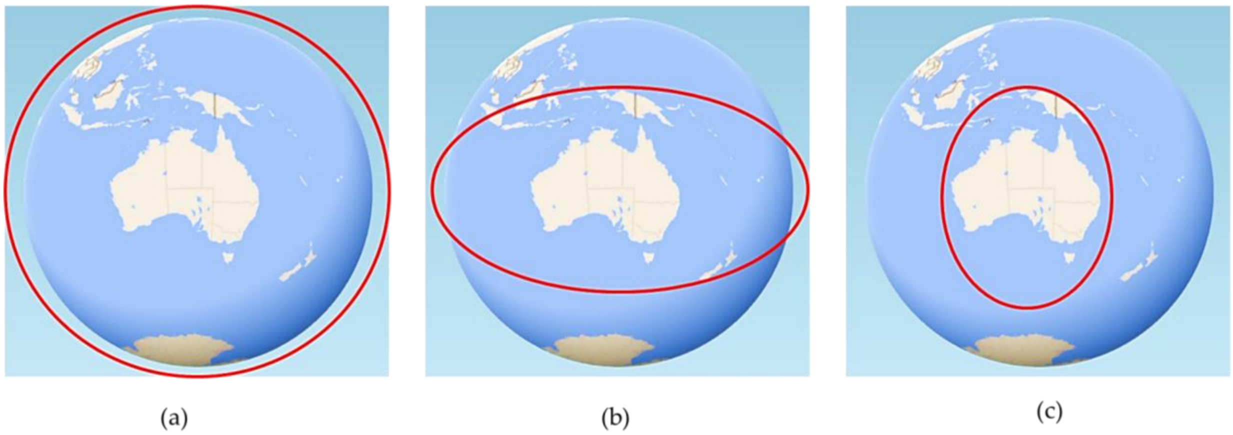


Figure 75: Distinct coverage modes (a) Global, (b) Zonal, (c) Regional.

In contrast to the monolithic satellite system, a constellation could provide persistent global or near-global coverage, meaning that at any given time, at least one satellite is visible from anywhere on Earth. Satellites are often situated in orbital planes that are complementary to one another and communicate with globally dispersed ground stations. Inter satellite communication may also be used to communicate with each other. The two main types of constellations: (i) Polar and Near-polar Constellations and (ii) Walker Constellation. Other constellation types are available in the literature, but the two mentioned above are the most important and widely used.

6.3.1.1 Polar and Near-Polar Constellations

The polar orbit is inclined at an angle of 90°, while the near-polar orbit constellation has an inclination similar to the polar orbits but can be tuned according to the orbit's specific requirements. The polar constellation model is interesting because it allows a purely geometrical solution, as shown in Figure 76. Walker [46, 223] looked at a variety of constellations, typically covering them with a street approach. Walker star patterns are near-polar constellations with just an orbital seam between ascending and descending planes. This is because so many orbits cross at the Poles, and the orbital planes intersect to form a star when viewed from one of the Poles.

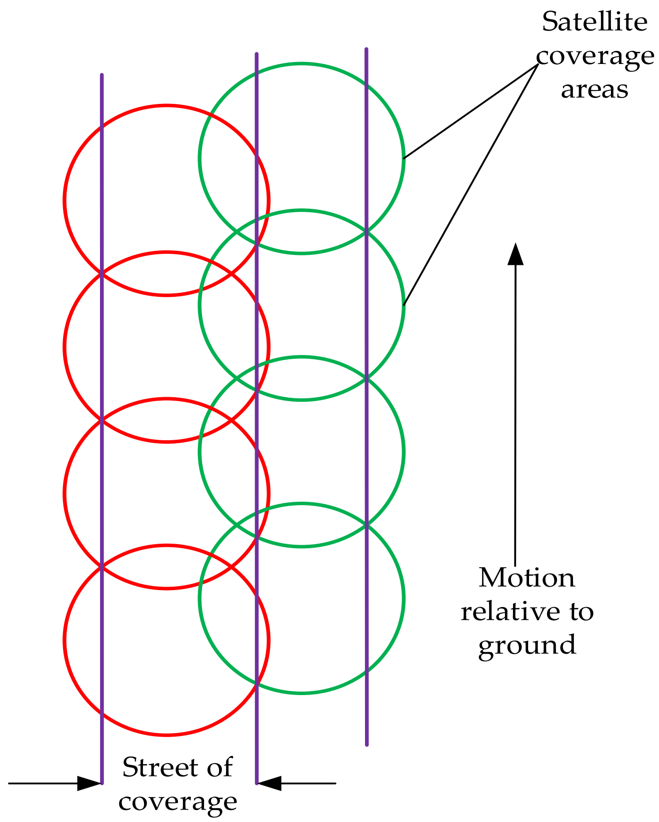


Figure 76: Street coverage of two satellites in a polar constellation, with red and green representing the satellites fields of view.

6.3.1.2 Walker Constellation

Walker proposed uniform constellations, as illustrated in Figure 77, with the inclination criterion relaxed to lower the needed number of satellites by reducing superfluous overlap at the poles. There is no way to identify a geometrical solution, but numerical analysis can reveal the total number of satellites (checking that coverage is ensured every time). The Delta constellation, sometimes called the Walker Delta or a Rosette, is one of the most well-known constellations [223].

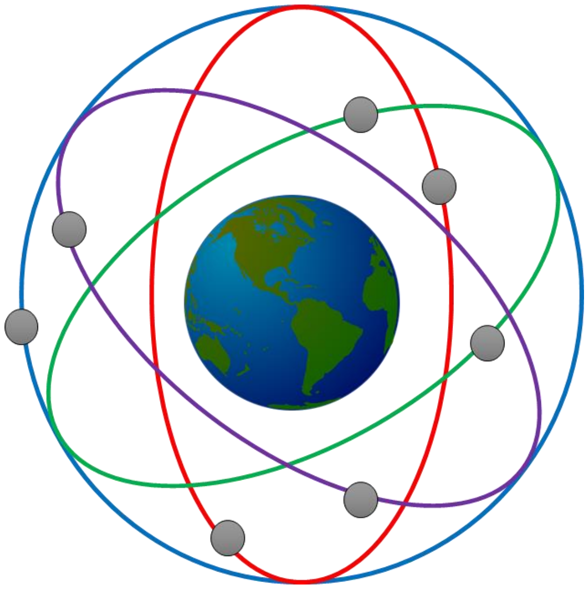


Figure 77: Walker delta constellation.

Aiming for effective coverage across the Australian continent, the Walker scenario will be appropriate for the constellation model. On the other hand, the Walker Delta design is appealing for the current research work because of its simplicity and economic feasibility [46, 224]. The parameters i, N_s, p, and f indicate the distribution of satellites in space, where i is the inclination, N_s is the total number of spacecrafts, p is the total number of orbital planes, and f is the phase difference between the participating spacecrafts in the neighbouring plane. The number of satellites in each orbit is given by $s = N_s/p$, where $p \mid N_s$. To avoid satellite collisions, the phase difference between the neighbouring spacecrafts of a specific plane is calculated using $f \times (360°)/N_s$, where f is an integer between 0 and $(p - 1)$ in this research.

The semi-major axis (a), eccentricity (e), inclination (i), the longitude of ascending node (Ω), the argument of perigee (ω), and true anomaly (υ) are the six Keplerian elements that make up a satellite's orbital parameters in three dimensions. Since the Walker Delta constellation is comprises of circular orbits, $e = 0$ and $\omega = 0$, a is equal to the radius of the orbit, and ν indicates the angle from the satellite's position vector to the ascending node. The ascending node's right ascension is expressed as $\Omega = (360°)/p$. At epoch $v_n = f_n$, where $n = 1, 2, \ldots, N_s$. The altitude (h), inclination (i) and elevation angle (ϵ) are optimisation design variables utilised by the constellation model. The orbital elements are utilised to establish the satellite's initial state (position and velocity). The initial state is then stretched around the Earth for a set amount of time. Finally, each satellite's coverage is calculated and updated. The steps are explained in detail below.

Satellite State Determination: At any given point in space, the satellite's position and velocity vectors determine its state $(\vec{Y})$. In the Perifocal coordinate system, PQW, the state $(\vec{Y}_{PQW})$, position $(\vec{r}_{PQW})$, and velocity $(\vec{v}_{PQW})$ vectors are represented by Equation (43-45), where μ is the standard gravitational parameter. The P axis is pointing towards perigee (ω), the Q axis points 90° degrees from P in the direction of satellite motion, and the W points perpendicular to the orbit. Thus, we have:

$$\vec{Y}_{PQW} = \begin{bmatrix} \vec{r}_{PQW} \\ \vec{v}_{PQW} \end{bmatrix} \tag{43}$$

$$\vec{r}_{PQW} = \begin{bmatrix} \dfrac{\text{acos}\,(v)}{1 + ecos(v)} \\ \dfrac{\text{asin}\,(v)}{1 + ecos(v)} \\ 0 \end{bmatrix} \tag{44}$$

$$\vec{v}_{PQW} = \begin{bmatrix} -\sqrt{\dfrac{\mu}{p}}\sin\,(v) \\ \sqrt{\dfrac{\mu}{p}}(\mathrm{e} + \cos(v)) \\ 0 \end{bmatrix} \tag{45}$$

Since ω = 0, the perifocal frame of reference is no longer used for calculations. Hence, the state variables are transformed into the Earth-Centred Inertial (ECI) system, IJK, where the I axis points to the Aries direction, the J axis points 90° east in the equatorial plane and the K axis passes through the north pole, using coordinate transformation.

Satellite State Propagation: Due to the gravitational interactions between the satellite and Earth, the movement of a satellite orbiting around the Earth is considered a two-body problem. Ideally, this can be expressed by simple equations of motion, but the presence of the perturbations complicates the problem: (i) Earth's non-homogeneity and oblateness, (ii) Third-body effects, (iii) atmospheric drag, and (iv) solar radiation pressure. In a real-world setting, the effect of satellite disturbances cannot be overlooked. The perturbing accelerations are taken into account in the two-body equation of motion using Cowell's Formulation:

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} + \vec{a}_p \tag{46}$$

where $\vec{a}_p$, is the perturbing acceleration, which corresponds to the total acceleration induced by external forces experienced by the satellite, and $\ddot{\vec{r}}$ is the resultant satellite acceleration. The number of perturbation sources considered in the problem determines the precise shape of $\vec{a}_p$. As part of this study, a simplified acceleration model that incorporates perturbations caused by the non-spherical central body is used. The perturbing acceleration experienced by the satellite is calculated using the gradient of the non-spherical Earth's gravitational potential, which is described using spherical harmonics [225]. The analysis is carried out considering the perturbations from the second (J_2), third (J_3), and fourth (J_4) zonal harmonics. Pertubations are taken into account because propulsion subsystem is not considered in the MDO; typically, orbit keeping strategies are always performed. Vallado [226] provides the components of the perturbing acceleration due to the J_2, J_3, and J_4 zonal harmonics and are linearly added to Equation 46. The initial state of the spacecraft is denoted by a first-order system by combining equation (12) with Cowell's second-order equation of motion:

$$\dot{\vec{Y}} = \begin{bmatrix} \vec{v} \\ -\frac{\mu}{r^3}\vec{r} + \vec{a}_p \end{bmatrix} \tag{47}$$

Equation 47 is the modified Cowell's formulation and can be solved using numerical integration methods.

Coverage Analysis: As a satellite observes a region on Earth, it projects a circular or rectangular imprint on the surface. The instantaneous coverage of the satellite is the distance between the satellite and a target point in the satellite field of view region (imprint region) at a given time. The value of Earth Central Angle λ is determined by Equation 48, where (Θ_s, Λ_s), denotes the latitudes and longitudes of the sub-satellite point and (Θ_t, Λ_t) denotes the latitudes and longitudes of the target [227].

$$cos\lambda = sin\Theta_s sin\Theta_t + cos\Theta_s cos\Theta_t cos|\Lambda_s - \Lambda_t| \quad (48)$$

Then, using the design variable ϵ from Equation 19, the nadir η is computed, which is then utilised to get the maximum earth central angle λ_{max}.

$$Sin\,\eta_{max} = cos\,\epsilon_{min}\left[\frac{R_E}{R_E + h}\right] \quad (49)$$

$$\lambda_{max} = 90° - \epsilon_{min} - \eta_{max} \quad (50)$$

Where R_E is the radius of the earth, h is the altitude, η_{max} is the maximum nadir angle, ϵ_{min} is the minimum elevation angle. For this research, a total of 465 grid points were chosen as objectives, dispersed across the specified region (Australia), as illustrated in Figure 78.

Figure 78: Selected data points.

For each grid point, the requirement for coverage, $\lambda < \lambda_{max}$ is evaluated. The following equation determines the entire temporal coverage:

$$C = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} T_{ij}}{nm} \quad (51)$$

where C is the constellation's coverage performance, n denotes the total number of time points analysed, m denotes the number of grid points, and T_{ij} denotes the coverage matrix.

6.3.2 Mission Payload

The system design is driven by the payload, which is the most significant subsystem of a satellite. Satellite design must consider factors such as payload size, weight, and power requirements early on. Typically, the payload for constellation missions is pre-defined and must be adequately launched in orbit. In the current scenario, however, the payload design is optimised. At this stage, knowing the exact value of the payload parameters is uncertain. As a result, practical estimating approaches to determine its approximate value [216, 224, 227-229] is employed. The size of the payload is determined using the calculations below, which are based on the satellite's altitude:

$$f = \frac{hd_x}{X/N_{samp}} \tag{52}$$

$$D = \frac{Bf}{Qd_x} \tag{53}$$

where h is the altitude, f is the focal length, D is the diameter of the aperture, d_x is the width of the cross-track detector, X is the resolution of the cross-track ground pixel, N_{samp} is cross-track detector samples per pixel, B is the operating wavelength, and Q indicates the imaging quality factor. Sub-scaling from a reference payload is used to determine the payload's mass $m_{payload}$, and power $P_{payload}$ based on the estimated aperture diameter $D_{payload}$ [216]:

$$R = \frac{D_{payload}}{D_0} \tag{54}$$

$$m_{payload} \approx KR^3 W_0 \tag{55}$$

$$P_{payload} \approx KR^3 P_0 \tag{56}$$

where R is the aperture ratio, $D_0, W_0,$ and P_0 are the reference payload's aperture diameter, mass, and power, and K is the scaling factor, which is 2 when R is less than 0.5; otherwise, 1 is considered. Here one should guarantee that the satellite footprints overlap to maximise total system coverage. Thus, satellite swaths must be larger than node crossings near the Equator. In this way, the orbits can cover a larger area at higher latitudes. The satellite's swath is determined by $2\lambda_{max}$. The perpendicular spacing between the orbits is used to calculate successive node crossings, as shown in the following equation:

$$S = sin^{-1}\,(sin(\Delta L)sin(i)) \tag{57}$$

where i is the orbital inclination angle, and ΔL is the longitudinal shift per orbit. The following part provides the list of sensors with their classification and characteristics. These models are organised to carry out the work in the most effective way possible. Figure 79 and

Figure 80 present a typical strategy for selecting the EO sensor and the AI algorithm. Given that there are two types of sensing approaches, active and passive, the classification is based on imaging and non-imaging sensors for the EO application. Imaging sensors can be further divided into three categories: optical, thermal, and radar sensors. Table 21-24 provides the detailed categorization of these sensors. The potential AI algorithms and the use cases in EO are outlined in Table 25. According to what has been reported, a variety of sensors and AI algorithms are currently available. It is essential to establish a formulation technique with which the choice of the algorithm may be made for the various applications. In space operations, it is impossible to keep to a specific type of AI because AI constantly evolves, and the models that makeup AI integrate within projects.

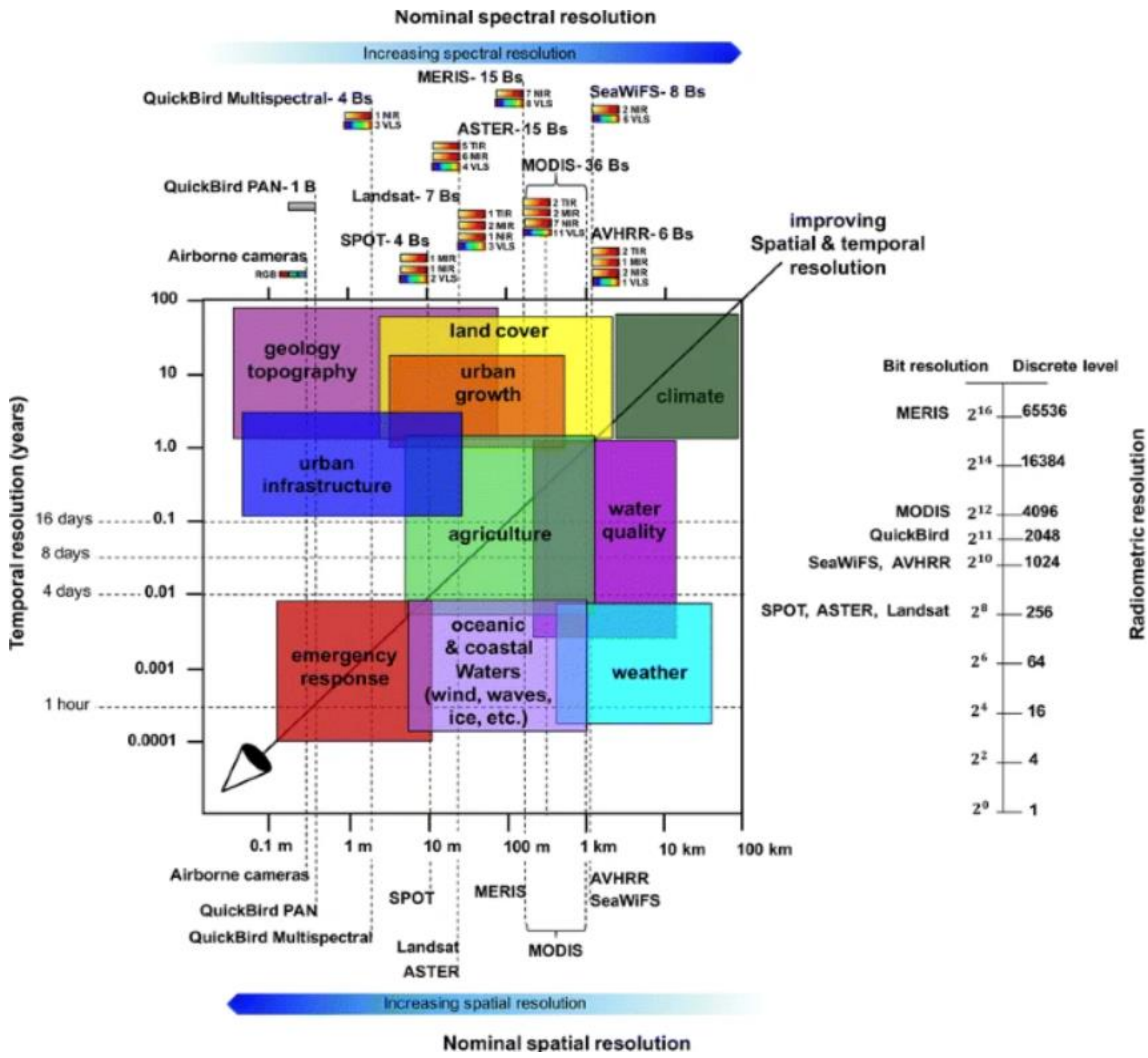


Figure 79: EO specific sensor application resolution requirements [66].

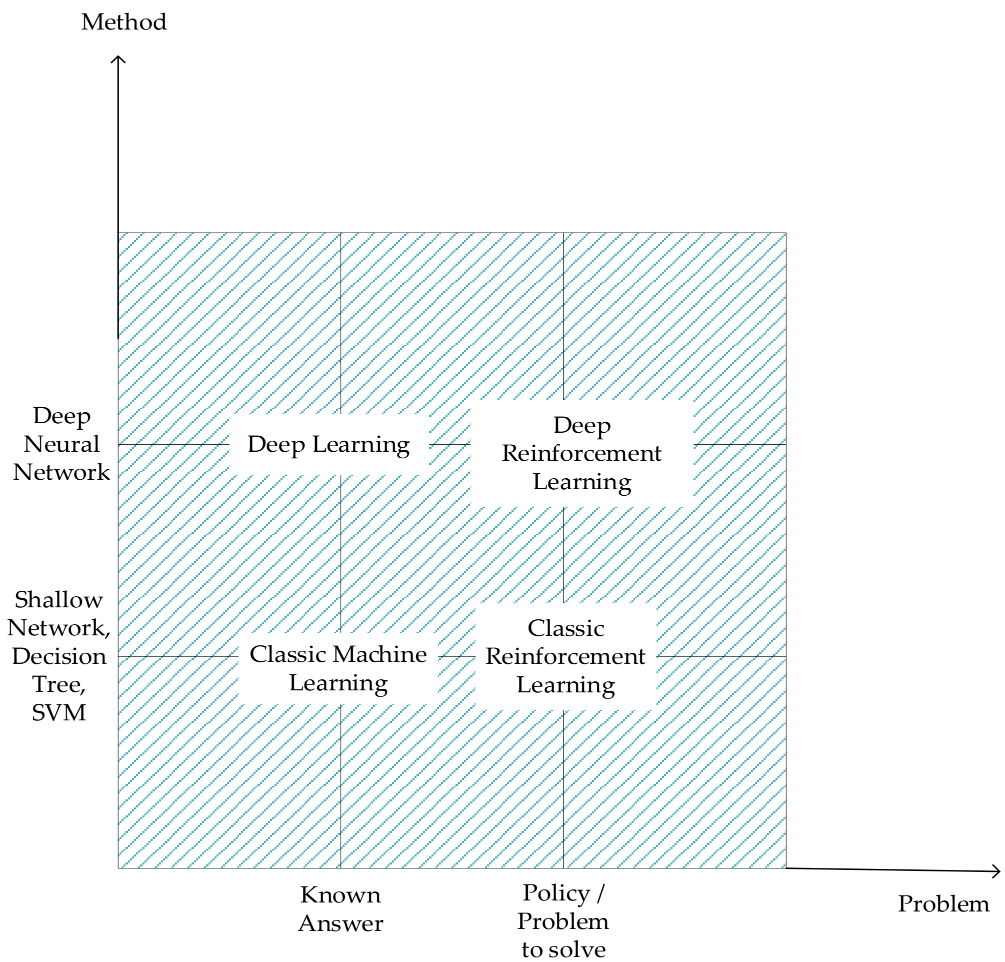


Figure 80: Classification standard of AI algorithm [67].

Table 21: Thermal sensor types and its applications [232].

Sensor	Operational wave band	Definition	Satellites sensors	Applications
IR imaging radiometer	UV, mid-to-far-infrared, or microwave	Measures the intensity of electromagnetic radiation	ASTER	Volcanological, mineralogical, and hydrothermal studies, forest fires, glacier, limnological and climatological studies and DEM
Imaging spectroradiometer	Infrared	Measure the intensity of radiation in multiple spectrums	MODIS, ASAS, IRIS	Sea surface temperature, cloud characteristics, ocean color, vegetation, trace chemical species in the atmosphere
Infrared imaging camera	Mid-far infrared	Measure reflected energy from the surface	-	Volcanology, determining thunderstorm intensity, identifying fog and low clouds

Table 22: Optical sensor types and its applications [232].

Sensor	Panchromatic systems	Multispectral systems	Hyperspectral systems
Spectral range (nm)	~430–720	~430–720 ~750–950	~470–2000
Satellites	QuickBird, SPOT, IKONOS	SPOT, QuickBird, IKONOS	TRW Lewis, EO-1

Spectral band	Monospectral, black and white, gray-scale image	Several spectral bands	10 to 100 of spectral bands
Spatial resolution	Submeter	Up to 1–2 m	Up to 2 m
Applications	Earth observation and reconnaissance applications	Red-green-blue (true color): visual analysis; Green-red-infrared: vegetation and camouflage detection; Blue-NIR-MIR: visualizing water depth, vegetation coverage, soil moisture content, and the presence of fires, all in a single image	(i) Agriculture; (ii) eye care; (iii) food processing; (iv) mineralogy; (v) surveillance; (vi) physics; (vii) astronomy; (viii) chemical imaging; (ix) environment
Advantages	High applicability in (i) imaging multiple targets; (ii) mosaic strips to large area; (iii) stereo and tristereo acquisition; (iv) linear feature acquisition, such as coastlines, pipelines, roads, and borders		
Disadvantages	Affected by sun illumination and cloud coverage. Polar areas with seasonal changes in sun illumination and the equatorial belt with persistent cloud coverage		

Table 23: Radar types and its applications [232].

Sensor	Operational wave band	Definition	Satellites sensors
Ka	40–27	0.75–1.11	Usually for astronomical observations
K	27–18	1.11–1.67	Used for radar, satellite communications, astronomical observations, automotive radar
Ku	18–12	1.67–2.5	Typically used for satellite communications
X	12.5–8	2.4–3.75	Widely used for military reconnaissance, mapping and surveillance
C	4–8	3.75–7.5	Penetration capability of vegetation or solids is limited and restricted to the top layers. Useful for sea-ice surveillance
S	4–2	7.5–15	Used for medium-range meteorological applications, for example, rainfall measurement, airport surveillance
L	2–1	15–30	Penetrates vegetation to support observation applications over vegetated surfaces and for monitoring ice sheet and glacier dynamics
P	1–0.3	30–100	So far, only for research and experimental applications. Significant penetration capabilities regarding vegetation canopy, sea ice, soil, and glaciers

Table 24: Non-Imaging sensor types and its applications [232].

Sensor	Operational wave band	Definition	Application
Radiometer	Ultraviolet, IR, microwave	To measure the amount of electromagnetic energy present within a specific wavelength range	Calculating various surface and atmospheric parameters

Altimeter	IR, microwave/radiowave, sonic	To measure the altitude of an object above a fixed level	Mapping ocean-surface topography and the hills and valleys of the sea surface
Spectrometer	Visible, IR, microwave	To measure the spectral content of the incident electromagnetic radiation	Multispectral and hyperspectral imaging
Spectro-radiometer	Visible, IR, microwave	To measure the intensity of radiation in multiple spectrums	Monitoring sea surface temperature, cloud characteristics, ocean color, vegetation, trace chemical species in the atmosphere
LIDAR	Ultraviolet, visible, NIR	To measure distance and intensity	Ocean, land, 3D topographic mapping
		Doppler LIDAR: measure the wave number for speed; Polarization effects of LIDAR: shape	Meteorology, cloud measurements, wind profiling and air quality monitoring
Sodar	Acoustic	As a wind profiler, sodar systems measure wind speeds at various heights above the ground and the thermodynamic structure of the lower layer of the atmosphere	Meteorology: atmospheric research, wind monitoring (typically in a range from 50 to 200 m above ground level)

Table 25: AI algorithms and potential sensors choices.

<table>
<tr><th colspan="2">AI Type</th><th>Potential Use Cases</th><th>Widely used AI Algorithms</th><th>Widely used EO Sensors (Imaging sensor)</th></tr>
<tr><td rowspan="5">Machine Learning</td><td>Supervised Learning</td><td rowspan="5">• Prognostic and Diagnostic
• Image Classification
• Forecasting
• Prediction
• Feature Selection
• Structure Discovery
• Segmentation
• Land Cover Classification
• Contrastive Sensor Fusion
• Crop Classification
• Image Scene Classification
• Road Surface Extraction
• Crop Type Classification
• Wildfire Fuel Mapping
• Wildfire Detection
• Hazardous Mapping
• Monitoring Volcanoes
• Mining
• Ship Detection
• Oil Spill Detection
• Sea Ice Monitoring
• Forest Monitoring
• Soil Moisture
• Critical Infrastructure
• High Precision Agriculture
• Weather Forecasting
• Study Of Regional Vegetation Coverage
• Wide-area Weather
• Cloud Patterns.</td><td rowspan="5">• Support Vector Machine
• Association Rule Learning Algorithms
• Bayesian Algorithms
• Artificial Neural Networks
• Clustering Algorithms
• Decision-trees
• Deep Neural Networks
• Dimensionality Reductions
• Ensemble Methods
• Instance-based Algorithms
• Regression
• Regularization
• Random Forest
• Logistic Regression
• Regression Decision Tree
• K-means
• K-nearest Neighbor
• Principal Component Analysis
• Linear Discriminant Analysis
• Graph-graph Based Method
• Heuristic Approach
• Monte Carlo Method
• Direct Policy Approach</td><td rowspan="5">• Panchromatic
• Multispectral
• Hyperspectral
• Radar
• Thermal IR</td></tr>
<tr><td>Unsupervised Learning</td></tr>
<tr><td>Semi-Supervised Learning</td></tr>
<tr><td>Reinforcement Learning</td></tr>
<tr><td>Deep Learning</td></tr>
</table>

6.3.3 Thermal Subsystem

Spacecraft orbiting Earth have several heat sources, which is shown in Figure 81. The major thermal radiation sources are the Sun, Albedo, Earth Infrared Radiation, Power on-board, and Re-emitted Radiation. Internal power dissipation in electrical components is another example (Joule effect). Aerothermal flux must also be considered during launch and re-entry, which is not depicted in the diagram. Internal heat dissipation and Earth's outgoing radiation are the only heat sources that thermal impact the spacecraft during the eclipse, and the spacecraft will begin to cool. As a result, the temperature of the satellite fluctuates cyclically along its orbit, rising and lowering during solar eclipses. Deep Space, which should be perceived as a dark body emitting at 3K, acts as the main contributor to cold.

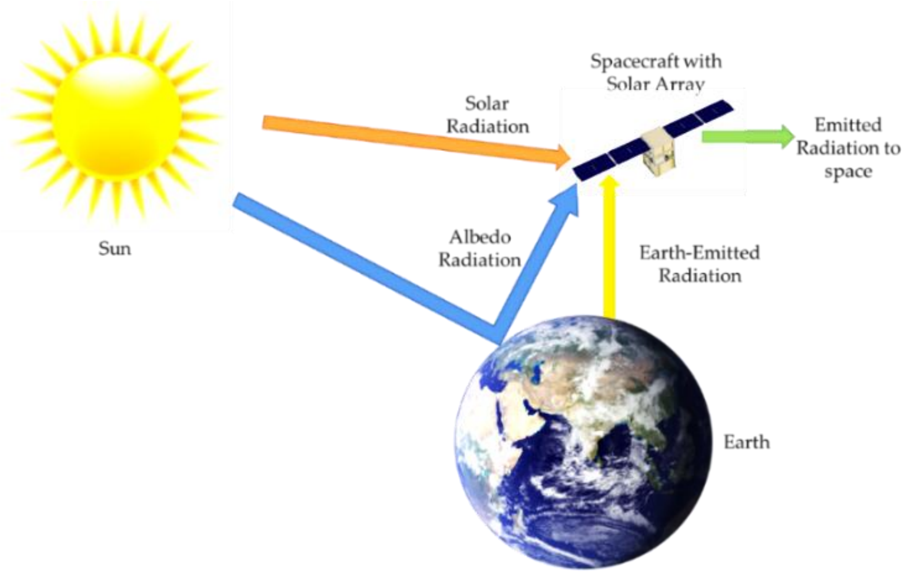


Figure 81: Spacecraft Thermal Environment.

The Sun radiation, the Earth's thermal radiation, and Albedo influence the satellite thermally in orbit. To keep the electronics in their operating range, the temperature within the satellite must be kept constant. Radiators positioned in the sun-facing orientation release excess heat accumulated inside the satellite into space. A typical orbit cycle of the satellite is depicted in Figure 82. The external environment and internal heat generation are the most frequent heat sources. Insulation and controlled heat rejection from radiators are the most common heat sinks. Initially, the satellite is assumed to be in steady-state equilibrium. The thermal balance equation for a spacecraft in orbit is expressed using Equation 58 [201]. The first element of the equation represents the capacitive term, and it takes into consideration the energy stored inside the structure as a function of its temperature variation. The parameters m and c represent the spacecraft's mass and specific heat, respectively. The incoming and outgoing fluxes owing to the Sun (Q_s), the Earth (Q_e), the Albedo (Q_a), the power generated on-board (P), and the energy re-emitted by the satellite surface (Q_r) combine to form this term [227, 233] as follows:

$$
\begin{gathered}
Q_{source} = Q_{sink} \\
mc\,\frac{dT}{dt} = Q_s + Q_e + Q_a + P - Q_r \\
Q_{external} + Q_{internal} = Q_{Radiator} + Q_{MLI}
\end{gathered} \tag{58}
$$

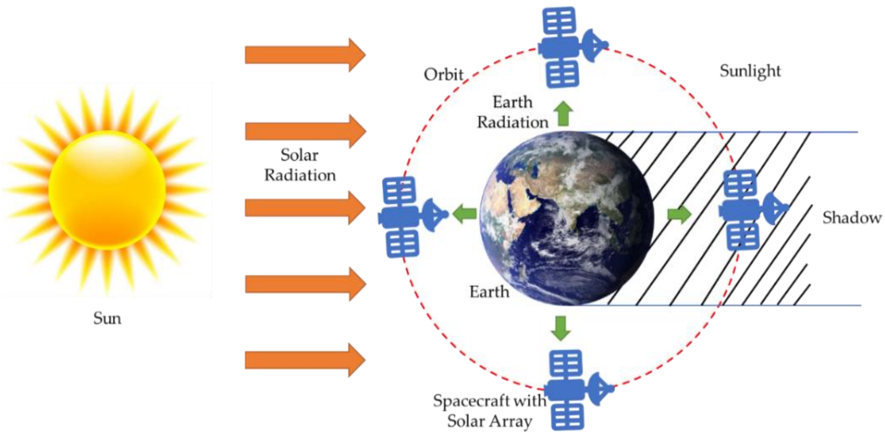


Figure 82: An illustration of the spacecraft's thermal environment in orbit.

Here it was assumed that the Sun is the primary source of radiation and that any external sources are insignificant. Except for the faces where the radiators are placed, the entire satellite is encased in Multi-Layer Insulation (MLI). Heat leaks from MLI are minimal and not included in the calculations. As a result, the heat balance equation is rewritten using the Stefan-Boltzmann law as follows:

$$\alpha S_0 A_R + Q_{internal} = \varepsilon A_R \sigma T^4 \tag{59}$$

where α is the material's absorptivity, ε is the material's emissivity, S_0 is the solar constant, σ is the Stefan-Boltzmann constant, T is the temperature of the spacecraft, A_R is the area of the radiator, and $Q_{internal}$ is the internal heat generation. The temperature T is computed for hot and cold situations using Equation 59 and A_R as the design variable. In the hot scenario, $Q_{internal}$ is 60% of the satellite power, while in the cold situation, it corresponds to 40% of the total satellite power. In the hot case, the temperature must not exceed 340K; in the cold case, it must not fall below 263K. Finally, Equation 60 and Equation 61 are used to calculate the mass and power of the thermal subsystem in which ρ_R is the areal radiator density. Thus, we have:

$$m_{thermal} = A_R \, \rho_R \tag{60}$$

$$p_{thermal} = \varepsilon\sigma A_R T^4 \tag{61}$$

6.3.4 Power Subsystem

The power subsystem generates the satellite's electrical power, which uses solar panels to do so. In addition to generating power and storing it, the power subsystem is responsible for delivering it to each subsystem and controlling it as needed. The size of the satellite and the area of solar panels have an impact on power generation. The eclipse's length determines the rechargeable battery's size and capacity used to store the generated energy. [227]. A typical architecture is shown in Figure 83.

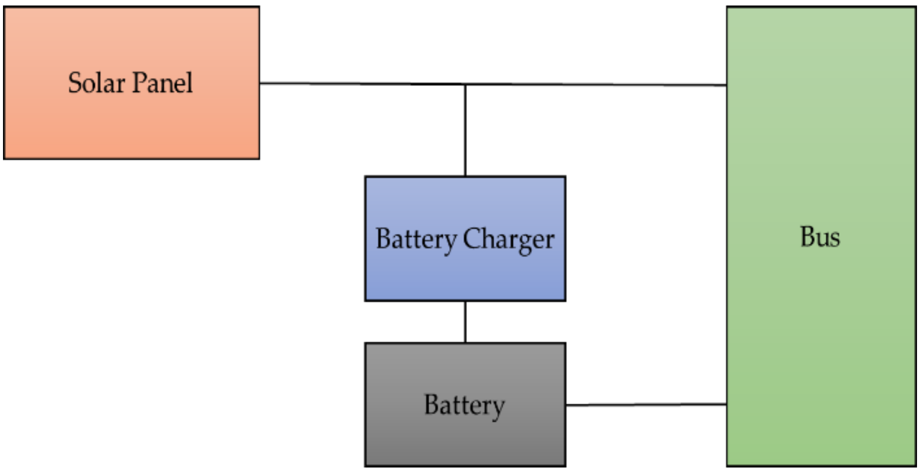


Figure 83: Power system architecture.

Solar Panel Sizing: Solar panels must be sized so that they generate more than the required power. The quantity of power required by the satellite determines the size of the solar panel. The satellite's power requirements are calculated as follows:

$$P_{req} = \frac{P_{payload}T_{payload} + P_{thermal}T_e + P_{batt}T_e + P_{others}T}{T - T_e} \tag{62}$$

Here, $P_{payload}$ is the required payload power in time $T_{payload}$, $P_{thermal}$ is the power required by the thermal subsystem during an eclipse T_e, P_{batt} is the power required for the battery to charge, and P_{others} is the overall power required by the remaining during the orbital period

T. The power provided by the solar arrays [234] is dependent on a number of factors, as shown in the following equation:

$$P_{gen} = S_0 X_i X_s X_e X_0 A_s \eta F_c (\beta_p \Delta T + 1) cos(\chi) \quad (63)$$

The solar constant is $S_0 = 1367\, W/m^2$, the correction factors are $X_i = 0.95$, $X_s = 0.9637$, $X_e = 1$ and $X_0 = 0.98$. A_s is the area of the solar panel, F_c is the solar array loss coefficient, β_p is the power temperature coefficient, η represents photoelectric conversion efficiency, and χ is the sun vector divergence angle from the solar array normal in the worst-case scenario (full-hot). The surplus power is computed using required and generated power as follows [234]:

$$P_{surplus} = (1 - dy)^{Lt} P_{gen} - (1 + 5\%) P_{req} \quad (64)$$

wherein d_y is just the annual solar panel power deterioration and L_t is the total duration of the mission.

Battery Sizing: The solar panel generates no power in the eclipse phase. Maintaining the satellite's power source necessitates the use of a rechargeable battery. The battery's discharge capacity, C, is determined by the eclipse duration and the amount of power required during the eclipse. The battery's Depth-of-Discharge (DOD) is 80 percent of its rated capacity, or C_{rated}. The area of the solar panels and projected battery capacity are used as design variables to calculate the mass of the power subsystem (m_{power}):

$$m_{power} = \rho_s A_s + C_{rated} \cdot \frac{V_{DB}}{\mu_b} \quad (65)$$

Where, ρ_s is the areal density of the solar array, V_{DB} is the battery voltage and μ_b is the battery's specific energy.

6.3.5 Structure Subsystem

The satellite structure protects the satellite subsystems in the launch and space environments. The structural elements can be treated as a separate subsystem for design and analysis purposes. This subsystem is in constant touch with the launch vehicle during launch and is subjected to significant static and dynamic loads. A satellite's load-carrying capacity is determined by its strength, and stiffness can be increased by careful material selection and suitable reinforcing. However, the satellite's weight must be kept as low as possible to lower the launch cost. The satellite in consideration for the current work is a semi-monocoque cuboid with identical lengths in both the X and Y directions, as shown in Figure 84.

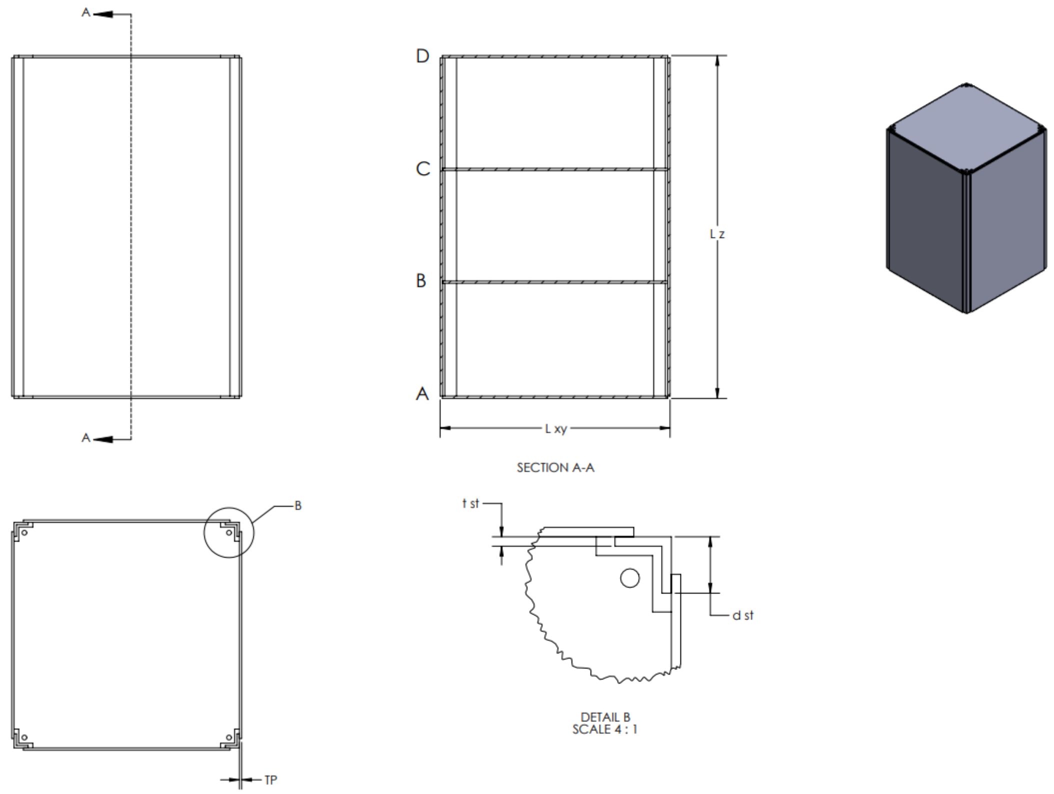


Figure 84: Satellite structural arrangement.

In the launcher, the launch adapter is attached to the outside of the bottom tray. The payload and other subsystems are stored in trays. The number of structural design elements is fixed with optimal dimensions and the material chosen for the spacecraft is space-grade aluminium alloy AL7075 T6. The material parameters of the product are presented in Table 26. The structural design of the spacecraft in this work is inspired by analytical structural design approaches from various works of literature [235-237]. Geometric dimensions and launch loads are the design variables and restrictions for structural optimisation, respectively. The size of the launch payload varies depending on the launchers. The launch loads corresponding to the launcher are tabulated in Table 27.

Table 26: Material properties of AL7075T6.

$E\ (GPa)$	ν	$G\ (GPa)$	$\rho\ (kg/m^3)$	$\sigma\ (MPa)$	$\tau\ (MPa)$
71.7	0.33	26.9	2810	503	331

Table 27: Launch loads.

Launch load	Longitudinal	Lateral
Acceleration (g)	± 10g	± 7.5g
Frequency (Hz)	≥ 50Hz	± 45Hz

Static Model: Static models or time-invariant satellite models are used to evaluate the structure under quasi-static limits imposed by the launcher. In the initial calculations, the satellite is assumed to be a cantilever beam locked by the launch adapter at the base. The satellite is subjected to a maximum axial force of 10g and a uniform lateral load of 7.5g. The maximum normal stress, σ_{max} and maximum shear stress, τ_{max}, are then computed using Equations 66-68, in which t_p is the thickness of the side panels, I_x is the satellite moment of inertia, M_{max} is the maximum bending moment, and V_{max} is the maximum shear force. A_{sat} is the cross-sectional area of the satellite, L_{XY} is the satellite dimension in X and Y directions. By replacing lateral acceleration a_{lat} and longitudinal accelerations a_{long} in $F = m_{sat} * a$, the lateral load, F_{lat} and longitudinal load, F_{long} are produced. The computed stress should be smaller than the material's yield strength, as shown in Table 26. Therefore, we have:

$$A_{sat} = 4[A_b + t_p(L_{XY} - t_p)] \tag{66}$$

$$\sigma_{max} = \frac{M_{max} h_p}{2I_x} + \frac{F_{long}}{A_{sat}} \tag{67}$$

$$\tau_{max} = \frac{V_{max} Q}{I_x L_{XY}} \tag{68}$$

Dynamic Model: The satellite must be able to bear both static and dynamic loads. Traditionally, the design is verified for static loads followed by dynamic loads. As part of our optimisation process, a set of design factors is iteratively evaluated against both static and dynamic loads. As part of our optimisation process, design factors are iteratively evaluated against static and dynamic loads. The dynamic model uses a spring-mass system with four degrees of freedom. The lumped masses for trays A, B, C, and D are represented by the masses $m_1, m_2, m_3,$ and m_4. The launch adapters as well as structural pieces that connect the trays, work as springs. The following are the equations of motion in the longitudinal and lateral directions, respectively:

$$\begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix} \begin{bmatrix} \ddot{z}_1 \\ \ddot{z}_2 \\ \ddot{z}_3 \\ \ddot{z}_4 \end{bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 & 0 & 0 \\ -k_2 & k_2 + k_3 & -k_3 & 0 \\ 0 & -k_3 & k_3 + k_4 & -k_4 \\ 0 & 0 & -k_4 & k_4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{69}$$

$$\begin{bmatrix} I_m & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix} \begin{bmatrix} \ddot{\varphi} \\ \ddot{x}_2 \\ \ddot{x}_3 \\ \ddot{x}_4 \end{bmatrix} + \begin{bmatrix} k_\varphi & 0 & 0 & 0 \\ 0 & k_4 + k_5 & -k_6 & 0 \\ 0 & -k_6 & k_6 + k_7 & -k_7 \\ 0 & 0 & -k_7 & k_7 \end{bmatrix} \begin{bmatrix} \varphi \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{70}$$

The launch adapter's longitudinal and lateral stiffnesses are k_1, k_ϕ. The longitudinal stiffness of the structural components between the trays A-B, B-C, and C-D is k_{2-4}, while the lateral stiffness of the structural elements between the trays A-B, B-C, and C-D is k_{5-7} . I_m is the satellite's mass moment of inertia. The solution of the eigenvalue problem $([K] - \omega_n^2[M]) = 0$ corresponds to the angular velocity of the satellite. $f_n = \frac{\omega_n}{2}$ is used to calculate the first natural frequency. Table 27 shows the frequency restrictions of the launcher. The calculated frequency must be greater than the launcher restrictions.

6.4 MDO Architecture

MDO problems are typically constrained nonlinear programming problems as long as the disciplinary boundaries are absent. They require determining the values of design parameters that maximise or minimise a particular design objective function while remaining constrained by constraints on the design. In a given system, the designer determines the design goals, limitations, and sometimes even the variables to be changed. Modelling the behaviour of a single element inside the system uses a disciplined analysis. Each disciplinary research is typically presented as a computer program, with complexity levels varying between empirical curve-fit statistics and a very detailed physics-based simulation. There are three different MDO architectures, as shown in Figure 85 [201].

a) **Monolithic:** Generally, MDO problems are solved by transforming them into single optimisation problems. The methodologies utilized to ensure the multidisciplinary feasibility of the optimal design are what distinguish architectures [201].
b) **Distributed Architectures:** The architectures that divide a large optimisation problem into smaller problems, each of which has the same solution when reassembled. [201].

c) **Hybrid architecture:** It refers to an architecture that combines features from two or more other architectures in a way that handles different discipline analyses or optimisations differently. Using MDF and Individual Discipline Feasible (IDF), a hybrid monolithic architecture could be developed by resolving the coupling of some disciplines within an MDA while resolving the remaining coupling variables through constraints. The usage of hybrid architectures is an area of MDO that has yet to be fully explored [201].

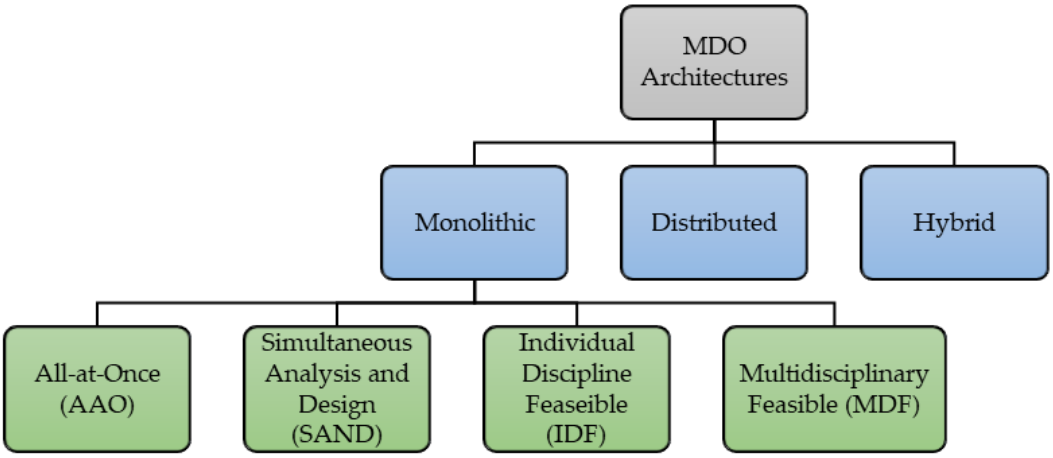


Figure 85: MDO Architecture classification.

According to Matins [201], distributed MDO architectures can be classified into three categories: MDF, IDF, or Simultaneous Analysis and Design (SAND) based on their monolithic counterparts due to the various approaches to handling the state and the coupling variables. It is similar to the classifications discussed previously in which for each variable eliminated from the problem statement, an equality constraint must also be eliminated from the optimisation problem. Even when distributed architectures are designed in isolation, a classification based on monolithic structures makes it a lot easier to see the links between them. A distributed architecture's problem formulation can often be easily derived from a monolithic architecture by adding certain parts and assumptions and following a specific decomposition scheme. This is because a distributed architecture always begins with a monolithic architecture and this classification can also be considered as a framework for developing new distributed architectures [199-201]. This architecture classification is depicted in Figure 86. Arrows indicate known links between the architectures. They included the "core" architectures in the diagram due to the enormous number of adaptations produced for some distributed architectures, such as introducing surrogate models and variations to address multi-objective problems [201].

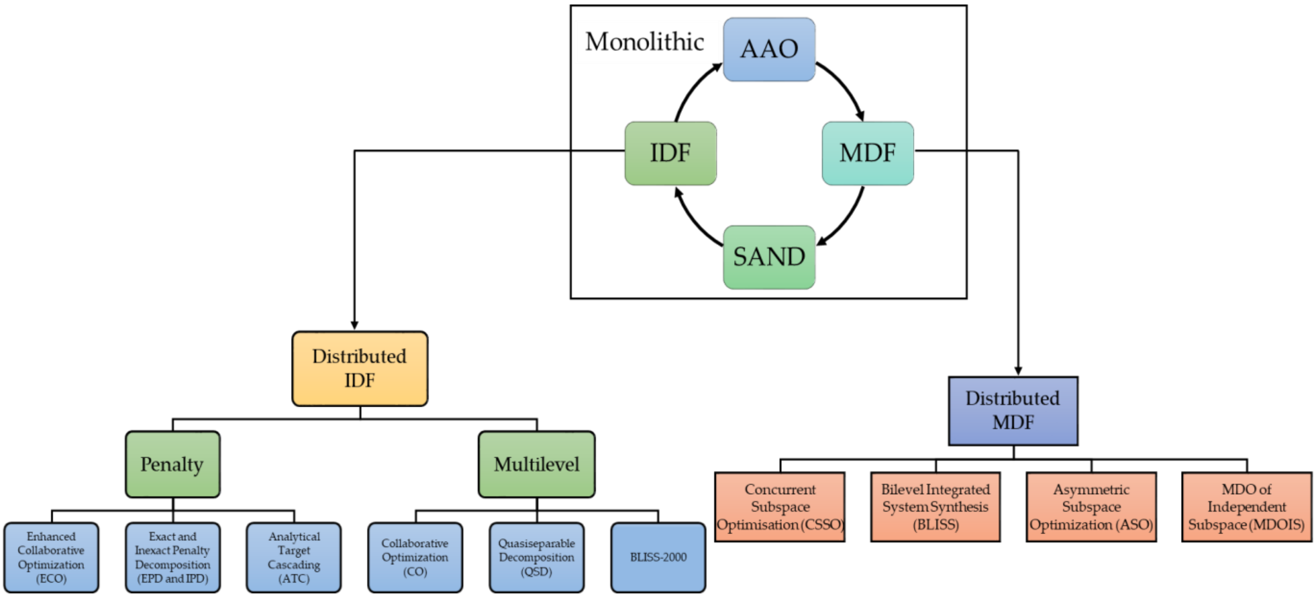


Figure 86: Distributed MDO architecture classification. Adapted from [201].

For our research work, MDF is selected because if the optimisation process is terminated early, MDF returns a system design that always satisfies the consistency constraints [197, 198]. As shown in Table 28, the MDO problem is formulated using a single objective function (minimise iDSS mass) and continuous design variables. The MDF architecture is used to overcome the problem. The process of handling the multidisciplinary coupling is the fundamental distinction between the architectures. Sequential Least-Squares Quadratic Programming (SLSQP) optimiser [238] is used, with the optimiser's convergence tolerance set to 1×10^{-3}. MDF requires a solver to handle the coupling. The section below provides the problem formulations in MDF architecture and their corresponding extended Design Structure Matrix (XDSM) diagram for each iDSS model. XDSM's architecture is described using expressions proposed by Ref [201] to describe operation sequences and data interactions. The conventions used in XDSM are as follows:

(i). The rounded rectangle represents the optimiser that controls the entire optimisation.

(ii). Rectangular-shaped nodes indicate the diagonally placed discipline modules.

(iii). The Parallelogram-shaped nodes represent the data and results.

(iv). The thick grey lines and thin black lines indicate the data flow and process flow.

(v). The input to the module is represented by the data flow in the vertical direction, whereas the data flow in the horizontal direction denotes the output from the module.

(vi). In addition to the thin black lines, a numbering system is also incorporated to indicate the process flow.

(vii). The process flow direction starts from module-0 and continues in sequential order up to module-n.

(viii). The process i is followed by process j until a specified condition is met is represented by $i \rightarrow j$.

(ix). The initial guesses $x^{(0)}$, variables at their optimum x^* and discipline-specific variables are placed in the outer nodes.

Table 28: Optimisation design variables.

Variable	Symbol	Unit	Range	Initial Guess
Altitude	h	km	[1200,1300]	1250
Inclination	i	deg	[45,55]	47
Elevation Angle	ϵ	deg	[15,25]	15
Length in X & Y direction	L_{xy}	m	[0.6,1.7]	0.8
Length in Z direction	L_z	m	[1,2]	1.2
Thickness of panel	t_p	m	[0.001,0.005]	0.005
L-bar width	d_{st}	m	[0.02,0.05]	0.03
L-bar Thickness	t_{st}	m	[0.001,0.005]	0.005
Length Ratio between plates A and B	AB	$-$	[0.2,0.5]	0.325
Length Ratio between plates B and C	BC	$-$	[0.25,0.375]	0.25
Area of Solar Panel	A_s	m^2	[1,5]	2
Area of Radiator	A_r	m^2	[0.1,2]	1.06

The MDF architecture is useful since it provides a viable interdisciplinary solution at each iteration. However, the optimisation must consider the practical design approach for this to be the case. The optimiser is placed on top of the Multidisciplinary Analysis (MA) modules in the MDF architecture. This means that a viable multidisciplinary solution is present in each MDF iteration. The disciplinary analysis models are iterated using the design variables (x) provided to the MA modules as inputs until a stable set of coupling variables (y) is generated. The design and the resulting coupling variables compute the objective and constraints. The MA is solved using standard iterative solvers such as block Gauss-Seidel and Newton solvers. Figure 87 shows the XDSM of the iDSS problem implementation in MDF architecture [201]. MDF architecture's general mathematical formulation is provided in the following equation:

$$\begin{aligned} & min\, f(x, y(x, y)) \\ w.r.t. \quad & x \\ s.t. \quad & g_i(x_0, x_i, y_i(x_0, x_i, y_i)) \geq 0 \; for \; i = 1, \ldots, N \end{aligned} \tag{71}$$

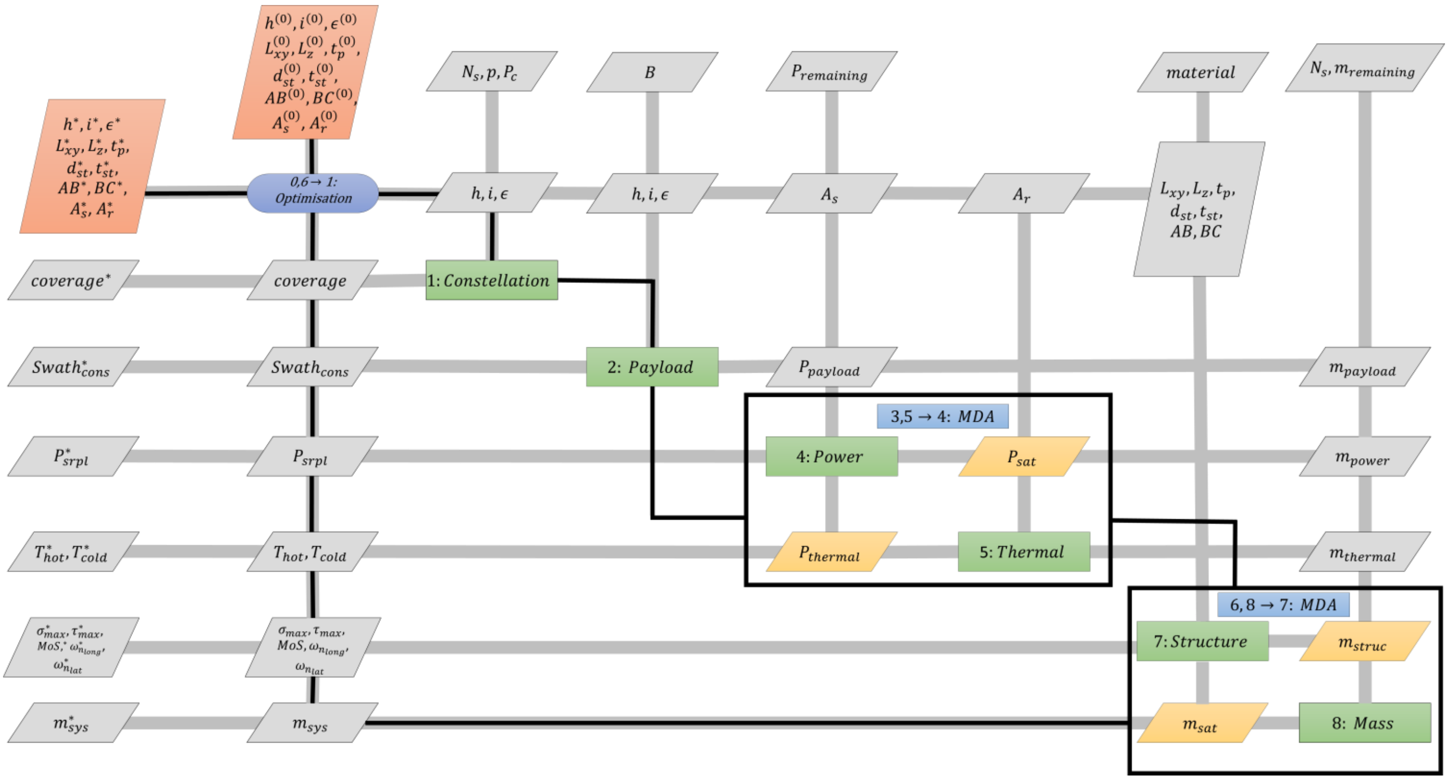


Figure 87: MDF XDSM representation.

6.5 Results and Discussions

The iDSS problem has been exacerbated and optimised, and Free-Flyer software is used to perform a coverage evaluation. The iDSS design problem for Australia was planned and addressed in MDF architecture, and the SLSQP optimiser was employed to optimise the solution once it was complete. To address the coupling between the disciplines in MDF, a solver is required. A Linear Direct solver is also required for the computation derivatives of the Nonlinear Block Gauss–Seidel Method (NLGBS). An N^2 diagram, sometimes referred to as an N-squared diagram, is a diagram that takes the form of a matrix and depicts the functional or physical interfaces that exist between the various elements of a system. It is implemented to methodically locate, define, tabulate, design, and analyze functional and physical interfaces. It applies to system interfaces and interfaces between hardware and/or software. The N^2 variable in the linear, non-linear, and all variables in the model are presented in Figure 88, Figure 89 and Figure 90. Figure 88 illustrates the N^2 diagram of the linear solvers connected with the model of the parameters that are being considered, and Figure 89 illustrates the N^2 diagram of the model; the only difference between the two is the solver. The clear relationship between all the variables connected to the model is illustrated in Figure 90, which is helpful for both the simulation and the optimisation processes.

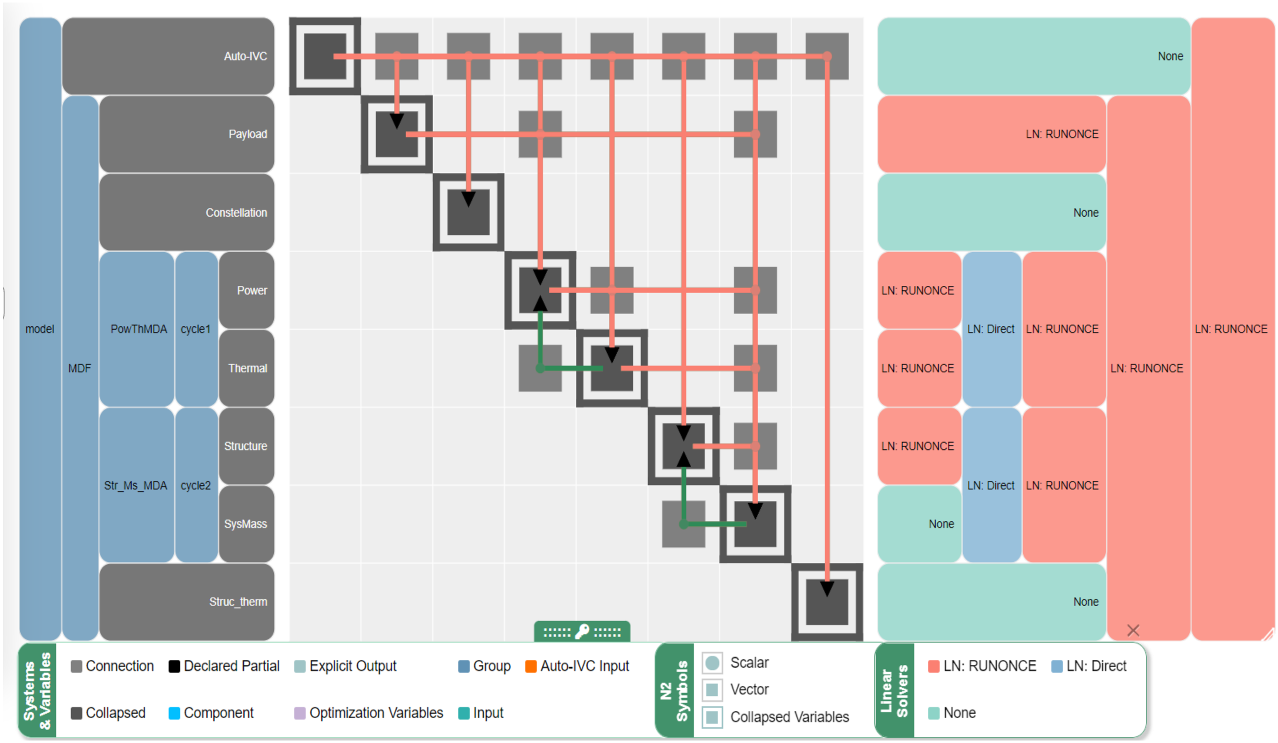


Figure 88: N^2 of linear solvers.

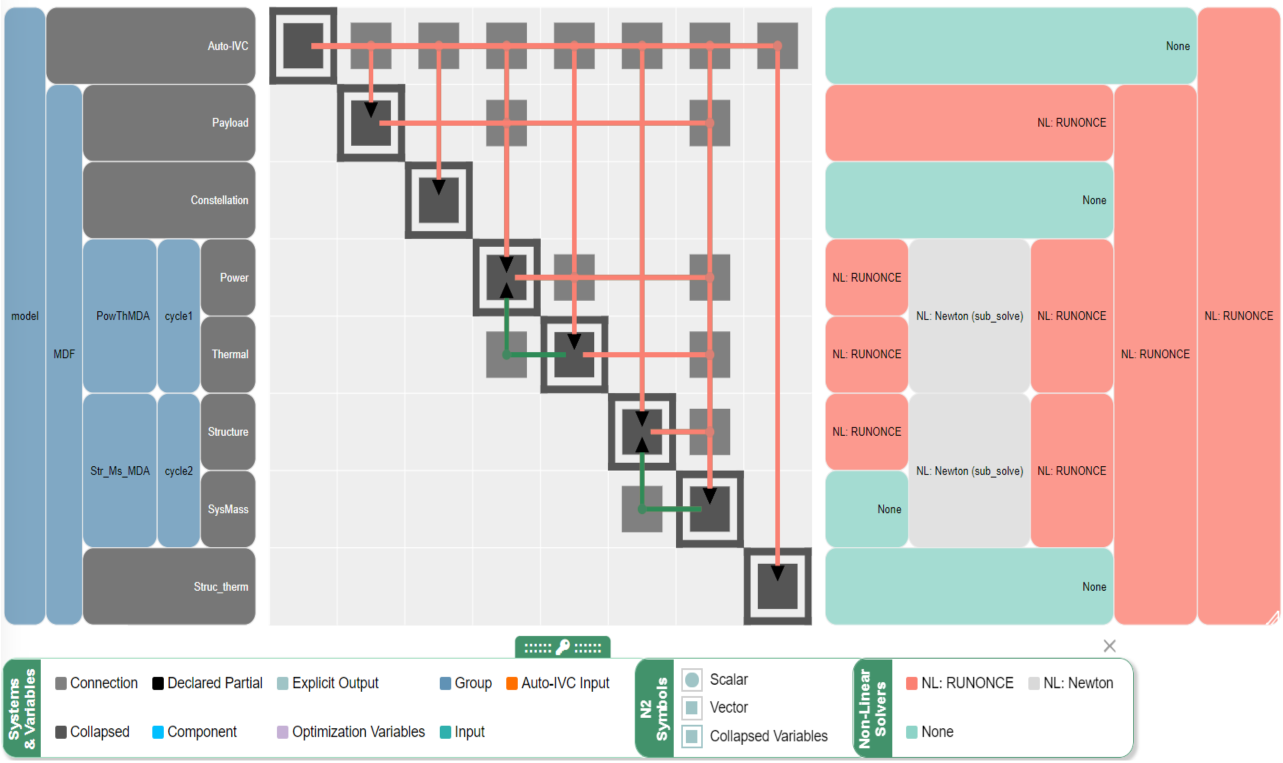


Figure 89: N^2 of non-linear solvers.

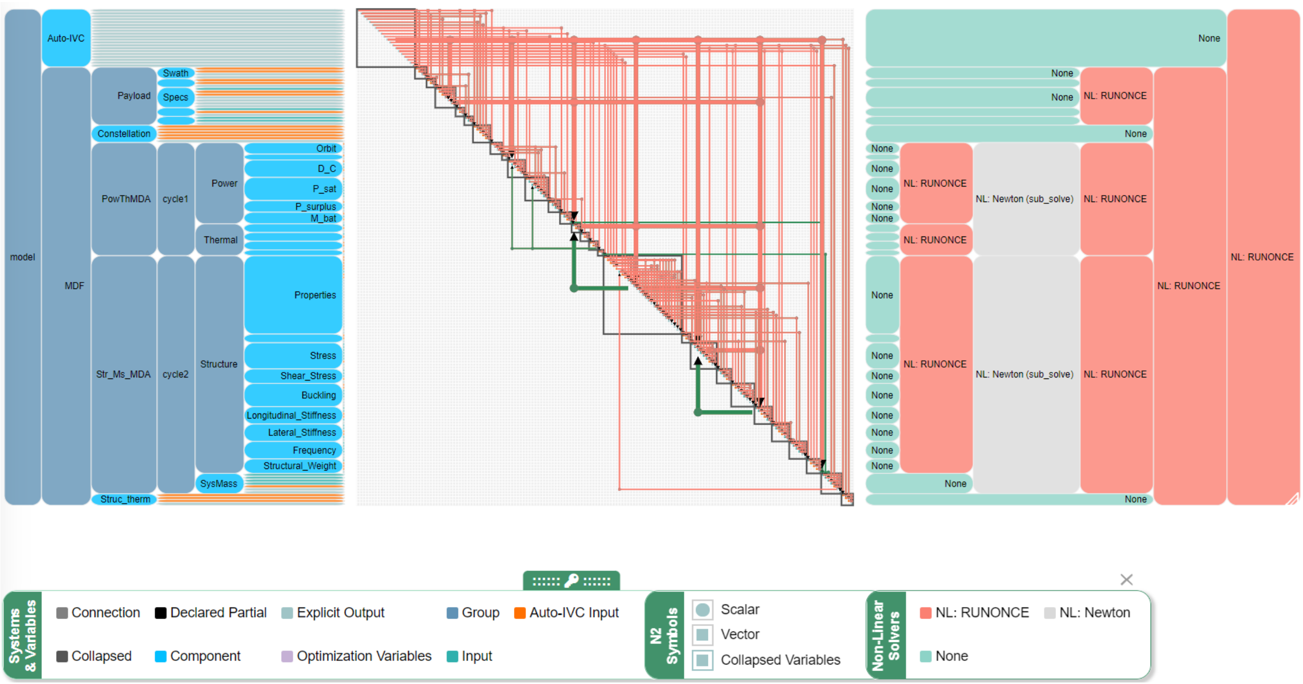


Figure 90: N^2 of all the variables in the model.

The optimisation was carried out using a COREi7 8th Gen Intel processor. The number of functions evaluated/called during optimisation indicates the processing power required by the architecture. The calls to calculate derivatives are likewise counted as part of the functional call estimates for each subsystem. The list of function calls of each subsystem for the MDF architecture is tabulated in Table 29.

Table 29: Function evaluation counts.

Subsystem	Function calls
Constellation	1014
Payload	1056
Power	2151
Thermal	2099
Structure	2490
Mass	2470
Total	11260

The optimisation results of the MDF problem are tabulated in Table 28. Due to careful consideration of design variable ranges and iDSS coverage computation, the computation time required is significantly shorter than the time generally required to solve a problem of this size. In the real world, the design variables have a wide range of values, and the temporal coverage is estimated throughout the mission's duration. However, the results obtained are sufficient for the MDF architecture. The information shown in Table 30 reveals that the total mass of the proposed satellite constellations is approximately 4668.63 kg, whereas the mass of a single satellite is about 187 kg. The simulations are performed with the help of the data currently at our disposal, but the results can be improved even further if more precise data is used. The literature and our simulations of the optimised result have led to a conclusion that for Australia, an inclination of 45 degrees is optimal to achieve the highest possible temporal coverage with the given constraints, which is approximately 70%. The other results are consistent with the constraint maintained based on the evidence available in the literature. Furthermore, it is clear that, from an optimisation standpoint, constellation design and payload significantly influence the MDO.

Table 30: Optimisation results.

Symbol	Unit	MDF Optimisation Results
$\boldsymbol{m_{sys}}$	kg	4668.632
$\boldsymbol{m_{sat}}$	kg	186.82
$\boldsymbol{h}$	km	1200.4

$\boldsymbol{i}$	deg	45
$\boldsymbol{\epsilon}$	deg	16.8
$\boldsymbol{L_{xy}}$	m	0.6
$\boldsymbol{L_z}$	m	1
$\boldsymbol{t_p}$	m	0.005
$\boldsymbol{d_{st}}$	m	0.03
$\boldsymbol{t_{st}}$	m	0.005
$\boldsymbol{AB}$	$-$	0.25
$\boldsymbol{BC}$	$-$	0.2502
$\boldsymbol{A_s}$	m^2	2.3641
$\boldsymbol{A_r}$	m^2	0.879
$\boldsymbol{P_{thermal}}$	W	55.89
$\boldsymbol{P_{payload}}$	W	213.52
$\boldsymbol{P_{satellite}}$	W	568.324
Coverage	%	70
Execution time	h	~6

FreeFlyer [239] is a package of COTS software that stands out as the most effective tool of its kind since it provides users with access to a comprehensive programming language for solving all types of astrodynamics problems and is free for academic use. The mission is analysed to validate our results even further. The mission simulation has been carried out with a start date of January 1st, 2020. Figure 91 illustrates the coverage for Australia for various situations. Clearly, the proposed constellation achieves constant monitoring; the pink points denote the time seen or observed; the cone indicates the Field of View (FOV) of the EO payload. The spatial coverage of the suggested walker constellation is 88.36%, and the Acquisition of Signal and Loss of Signal (AOS/LOS) are illustrated in Figure 91. Figure 92 depicts coverage on the poles; lower coverage is evident there. Figure 93 represents the entire mission coverage.

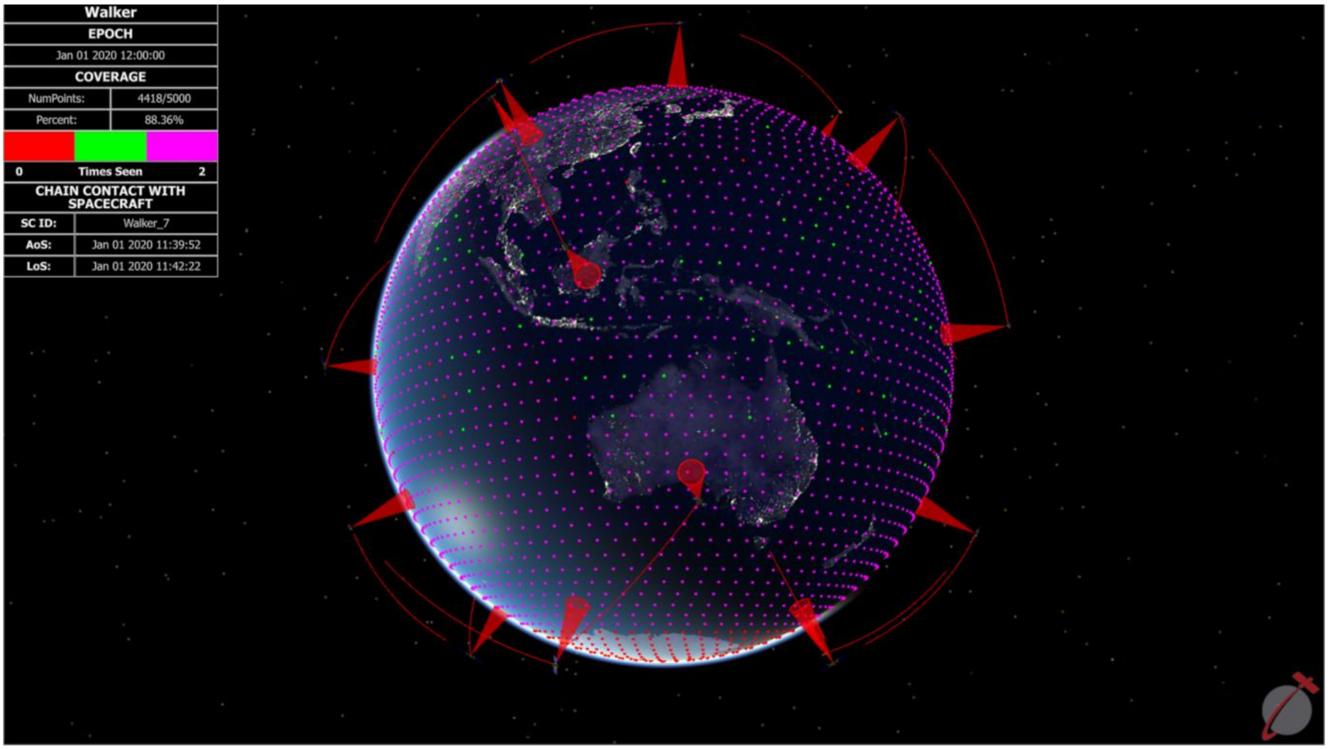


Figure 91: Coverage of Australia.

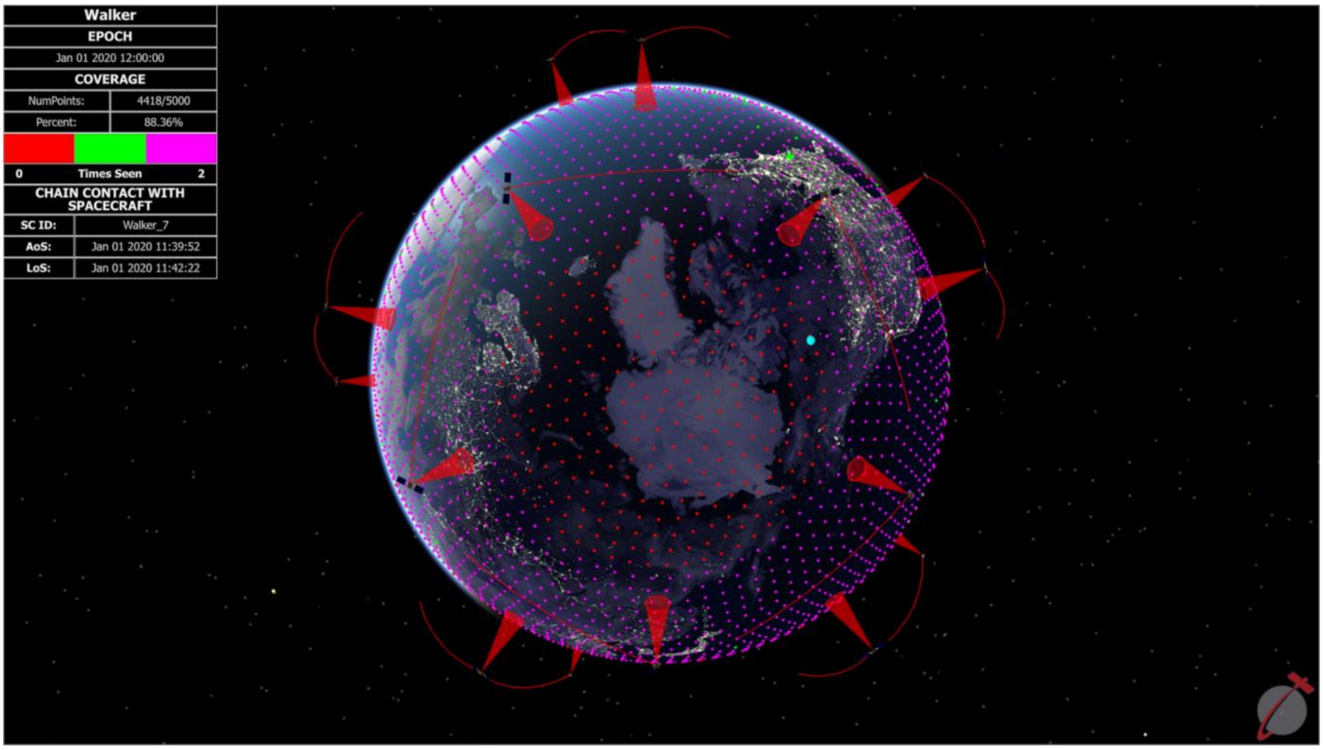


Figure 92: Coverage in Poles.

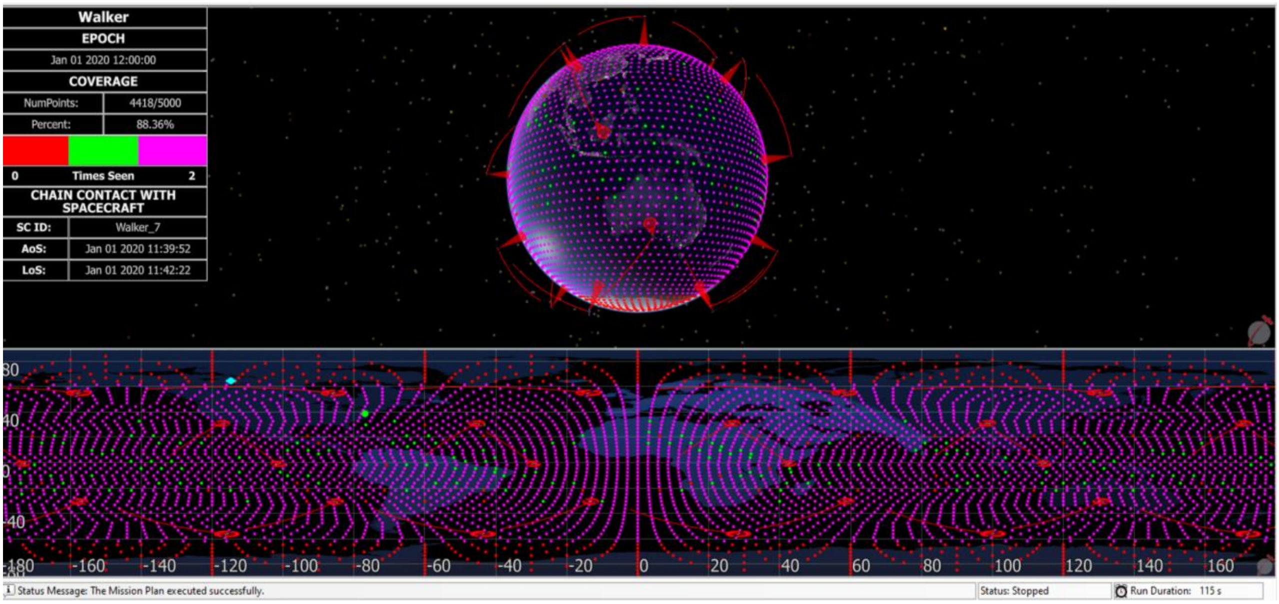


Figure 93: Coverage of the entire mission.

The resultant constellation parameters provide satisfactory spatial coverage. The spatial coverage is further enhanced by adding more satellites to the constellation and repeating the optimisation process.

6.6 Conclusion

iDSS represents a paradigm shift compared to monolithic satellites both in a standalone and in a constellation layout. To unleash the potential of this emerging technology, it is critically important to evolve the design methodology and the associated models compared to the traditional satellite system engineering best practices.

The research presented in this chapter developed and optimised an iDSS constellation for the EO mission. For maximum coverage, mission adaptability and to improve the revisit frequency, iDSS are used, the same has been modelled as an MDO problem. The MDF design of the design problem was then optimised, and the iDSS problem was then incorporated. Later, in the iDSS design process, straightforward analytical models are insufficient.

To create a more comprehensive design, all subsystems must be considered, and high-fidelity simulation models must be used. This increases the complexity and cost of the optimisation procedure. In the context of engineering design, it may be useful to be able to produce a superior design that is not always mathematically optimal. When this occurs, MDF architecture is useful since it consistently produces a practicable multidisciplinary solution, provided that the workable design approach is taken into account during optimisation. Other discipline models and components will be included in future studies to enhance the iDSS model. The problem will be transformed into a mixed-integer non-linear problem having discrete variables for optimisation based on the knowledge learned about their behaviour. The other architecture available in the literature will be compared with the results to determine which is more appropriate and efficient for the application. In the upcoming research, optimisers that are available as COTS software will be employed to obtain higher precision.

Chapter 7

Conclusion and Recommendation for Future Research

This chapter summarises and concludes the doctoral research work and provides future research directions in this area. The concluding remarks are drawn to address the research objective.

7.1 Conclusion

The thesis summarizes doctoral research results on Trusted Autonomous Satellite Operations (TASO) for Distributed Satellite Systems (DSS). The research focused on the functionality and potentialities of Artificial Intelligence (AI) to develop and enhance the autonomy level of Earth Observation (EO) missions. Several factors influenced the choice of domain, methods, and case studies, which can be understood in light of the research group in which this research was conducted, as well as the Australian EO space strategy and SmartSat CRC research priorities.

Even while the increased efficacy in operational capacities was one of the early drivers of autonomous operation in the space domain across the industry, it is evident that the immense potential of the TASO extends even further. In the context of implementing state-of-the-art DSS, a review of the whole body of literature in space operations identified major gaps that need to be addressed, notably with regard to trusted autonomy. In this regard, a novel concept known as intelligent DSS (iDSS) was presented to address both the system's mission astrionics and service astrionics aspects by employing the predictive and reactive components of AI within the architecture. It is envisaged that iDSS solutions will go beyond the traditional satellite operating role and become components crucial to the mission and safety-critical in the next generation of trusted autonomous space flight systems. On-board data processing, in which a combination of real-time measurements from distributed sensor networks are used to process the data on-board the satellite and facilitate the resultant reconfiguration of systems and re-planning of mission activities, is at the forefront of this transformation. In addition, iDSS systems are distinguished by their innovative capabilities of real-time decision-making and the dynamic management of missions.

The following is a summary of the findings from this investigation in terms of the research objectives that were accomplished:

1. **Conduct a thorough and in-depth analysis of the AI and DSS current state-of-the-art to find new requirements for TASO.**

A comprehensive review of the current state-of-the-art in DSS was performed to identify existing findings in the sector as well as the prerequisites associated with developing iDSS for trusted autonomous systems. This was done in light of the recognized potential, and to capitalise on it. This review's scope was expanded so that it would also look into the different DSS architectures and provide a unified classification. A comprehensive investigation at the use of TASO in space, as well as the ever-changing human-machine interaction. Then a review of the AI methods that have been implemented in space, along with an analysis of the potential applications of AI in space operations, beginning with the various segments, applications, and orbits. The review concluded that the most significant barrier to the actualisation of the iDSS concept was not associated with the limitations of the existing technologies or methods but rather the engineering approach to design and develop iDSS systems, in particular TASO. A solution was proposed for employing iDSS in space operations. Design case studies were carried out to verify this comprehensive approach of iDSS operation and their potential is demonstrated.

2. **Identify AI inference techniques for wildfire detection and develop an iDSS for real-time/near real-time disaster management. Finally, identify and implement mission management and reconfiguration options to ensure an acceptable level of operational capability for wildfire management.**

In this case study, on-board data processing of hyperspectral data from the PRISMA Mission was considered, and the same has been used to detect wildfires on-board the satellite to provide real-time alerts during the catastrophe event. A One-Dimensional (1D) Convolution Neural Network (CNN) has been used to process data in the space segment. Commercial-Off-The-Shelf (COTS) astrionics, i.e., hardware accelerators, are used to demonstrate this capability. The proposed model has been tested in the DSS for real-time disaster management, which will increase AOI coverage while decreasing revisit time. Since the DSS are connected via Inter Satellite Links (ISL) and the AI on-board, the data can be processed and communicated within the DSS network, enabling intelligent DSS (iDSS) operations. This thesis proposed a Low-Earth Orbit (LEO) iDSS constellation for real-time/near real-time wildfire monitoring. Because the iDSS is always connected via ISL, active AOCS is not always required; rather, only when one of the constellation's satellites detects a wildfire, it can communicate with the remaining nearby satellites and perform active reconfiguration to achieve the mission objective.

3. **Develop an iDSS mission for monitoring Australia's Maritime with autonomous orbit control.**

Australia is surrounded on all sides by water, and its maritime resources are one of its most valuable assets. As a result, strong and robust maritime security arrangements are required to contribute to Intelligence, Surveillance, and Reconnaissance (ISR) operations and MDA, which can be accomplished using iDSS technologies. It is demonstrated here that TASO is feasible with low-thrust electric propulsion for two main goals: achieving and maintaining satellite orbits on the constellation using absolute orbit navigation and, on the formation, using a more precise relative orbit navigation. As a result, each autonomous orbit control objective has its own navigation type. The potential application of this approach for Synthetic Aperture Radar (SAR) Along-Track (AT) interferometry with multiple baselines is demonstrated by selecting the relative orbit elements in such a way that the formation is robust in the sense of passive collision avoidance.

4. **Develop a Multidisciplinary Design Optimisation (MDO) methodology for iDSS to ensure persistent coverage over Australia.**

This case study was designing and developing an iDSS to provide persistent coverage across Australia/Australalasia, as there is no indigenous satellite constellation of iDSS specific to Australia. This was accomplished using the free data that was available online. In order to carry out comprehensive surveillance over Australia as part of an EO mission, an iDSS constellation was designed, developed, and optimised. Due to the fact that the frequency of wildfire breakouts has significantly increased over the past few years due to climate change, the EO operation consists of an optical payload (Hyperspectral) for the detection of wildfires, was chosen. iDSS are utilised to achieve maximum coverage as well as to enhance the revisit frequency. This problem has been modelled as an iDSS Multidisciplinary Design Optimisation (MDO) problem. The results of the optimisation were provided, and a simulation of the coverage in both spatial and temporal aspects was carried out and presented.

7.1 Recommendation for Future Research

The contributions of this thesis serve as a foundation for the future work and complete operation of iDSS systems as key enablers for on-board autonomy and mission performance assurance in TASO. These contributions include a critical review of the state-of-the-art in DSS and the operation of iDSS in TASO, the development of a generalised methodology for the design and integration of iDSS systems, as well as its application to the case studies that were considered. In particular, initiatives for the continued development of the fundamental iDSS components of distributed sensor networks and intelligent and dynamic decision-making based on situational awareness should be addressed by future studies. The following areas have been identified and recommended for future research in the field of iDSS:

i. **Utilising the other optical sensor data for the wildfire management:** In order to make use of both the spectral and spatial information contained within the PRSIMA data, additional research is required. It is possible to conduct transfer learning and assess the applicability of the model that has been developed by applying it to other hyperspectral data sets. In addition, a hardware-in-the-loop test could be carried out in order to compare the results of the simulation with the results of the experiment.

ii. **Exploring the use case of iDSS for other disaster events:** Additional study can be done to investigate the application of iDSS to other kinds of natural disasters, such as the detection of floods or volcanic eruptions, for example.

iii. **Exploring the iDSS use and applicability in space exploration missions:** Further investigation into the use case of iDSS in the interplanetary mission is something that can be done. The iDSS will be of great assistance for opportunistic science, the monitoring of rare events, and other similar pursuits.

iv. **iDSS Mission Planning and Scheduling Operations:** Additional research can be done in the autonomous mission planning and scheduling for the iDSS operation. This research can be based on the iDSS potential to dynamically adapt to the external environment.

v. **Resilient components in iDSS:** Future work can be done to address the iDSS's resilience component, which needs to be taken into account. The remaining satellites in the iDSS can therefore reconfigure themselves to continue the mission and successfully complete it even if one of them degrades. AI integration into the iDSS framework of the cyber-security measures should also be taken into account.

vi. **Heterogeneous iDSS and Data fusion:** Probability exists for additional research on the integration and processing of both hyperspectral and SAR data from heterogeneous iDSS. This can be difficult as different systems frequently employ distinct data processing and monitoring techniques. Various types of integrated models, such as sensor correction and calibration, are also in demand. This requires the formation of a team of developers with diverse backgrounds in areas like data fusion, subsystem modelling, etc. Regardless of the nature of the application, iDSS development requires a substantial amount of data and experience.

vii. **Incorporation of AI in the iDSS Design and Optimisation:** Further research is required to enhance the presented iDSS model with all subsystem models and components with high-fidelity simulation models should be included. Moreover, AI and surrogate models should be included in the design phase to automate the design process.

Chapter 8

References

[1] L. Feruglio, "Artificial intelligence for small satellites mission autonomy," Ph.D., ING-IND/05 - Aerospace Plants and Systems, Politecnico di Torino, Online, 2017.

[2] J. L. Moigne, J. C. Adams, and S. Nag, "A New Taxonomy for Distributed Spacecraft Missions," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 13, pp. 872-883, 2020, doi: 10.1109/JSTARS.2020.2964248.

[3] J. L. Moigne, M. M. Little, and M. Cole, "New Observing Strategy (NOS) for Future Earth Science Missions," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium,* pp. 5285-5288, 2019.

[4] N. Cramer *et al.*, "Design and Testing of Autonomous Distributed Space Systems," presented at the Small Satellite Conference, 2021. [Online]. Available: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?filename=0&article=5003&context=smallsat&type=additional.

[5] B. A. Corbin, "The value proposition of distributed satellite systems for space science missions," 2015.

[6] M. Grasso, A. Renga, G. Fasano, M. D. Graziano, M. Grassi, and A. Moccia, "Design of an end-to-end demonstration mission of a Formation-Flying Synthetic Aperture Radar (FF-SAR) based on microsatellites," *Advances in Space Research,* 2021.

[7] E. Joffre *et al.*, "LISA: Heliocentric formation design for the laser interferometer space antenna mission," *Advances in Space Research,* vol. 67, no. 11, pp. 3868-3879, 2021/06/01/ 2021, doi: https://doi.org/10.1016/j.asr.2020.09.034.

[8] L. Plice, A. D. Perez, and S. West, "HelioSwarm: Swarm Mission Design in High Altitude Orbit for Heliophysics," 2019.

[9] R. T. Rajan *et al.*, "Applications and Potentials of Intelligent Swarms for magnetospheric studies," *Acta Astronaut,* vol. 193, pp. 554-571, 2022/04/01/ 2022, doi: https://doi.org/10.1016/j.actaastro.2021.07.046.

[10] D. Selva, A. Golkar, O. Korobova, I. Cruz, P. Collopy, and O. de Weck, "Distributed Earth Satellite Systems: What Is Needed to Move Forward?," *Journal of Aerospace Information Systems,* vol. 14, pp. 1-26, 08/08 2017, doi: 10.2514/1.I010497.

[11] R. Preston, "Distributed Satellite Constellations Offer Advantages Over Monolithic Systems. ," no. Santa Monica, CA: RAND Corporation, 2004. [Online]. Available: https://www.rand.org/pubs/research_briefs/RB92.html.

[12] E. J. M. Angelita C. Kelly, "The A-Train: NASA's Earth Observing System (EOS) Satellites and other Earth Observation Satellites," *4th IAA Symposium on Small Satellites for Earth Observation,* vol. IAA-B4-1507P, 2003.

[13] R. Sabatini, "IEEE Distinguished Lecture: Aerospace Cyber-Physical and Autonomous Systems," ed, 2020.

[14] J. Utzmann *et al.*, "Space-based space surveillance and tracking demonstrator: mission and system design," 2014.

[15] M. K. Ben-Larbi *et al.*, "Towards the automated operations of large distributed satellite systems. Part 1: Review and paradigm shifts," *Advances in Space Research,* vol. 67, no. 11, pp. 3598-3619, 2021/06/01/ 2021, doi: https://doi.org/10.1016/j.asr.2020.08.009.

[16] M. K. Ben-Larbi *et al.*, "Towards the automated operations of large distributed satellite systems. Part 2: Classifications and tools," *Advances in Space Research,* vol. 67, no. 11, pp. 3620-3637, 2021/06/01/ 2021, doi: https://doi.org/10.1016/j.asr.2020.08.018.

[17] R. T. Rajan, A. Salmeri, D. Haken, J. Cohen, and C. Turner, "APIS: Applications and Potentials of Intelligent Swarms for magnetospheric studies," in *Proceedings of 71st International Astronautical Congress-The Cyberspace Edition*, 2020: International Astronautical Federation.

[18] M. Rosso, A. Sebastianelli, D. Spiller, P. Mathieu, and S. Ullo, "On-Board Volcanic Eruption Detection through CNNs and Satellite Multispectral Imagery," *Remote Sensing,* vol. 13, p. 3479, 09/02 2021, doi: 10.3390/rs13173479.

[19] R. M. Millan *et al.*, "Small satellites for space science: A COSPAR scientific roadmap," *Advances in Space Research,* vol. 64, no. 8, pp. 1466-1517, 2019/10/15/ 2019, doi: https://doi.org/10.1016/j.asr.2019.07.035.

[20] M. Campbell and K. Böhringer, "Intelligent Satellite Teams Space Systems," 05/03 1999.

[21] K. Ranasinghe *et al.*, "Advances in Integrated System Health Management for Mission-essential and Safety-critical Aerospace Applications," *Progress in Aerospace Sciences,* vol. 128, p. 100758, 11/18 2021, doi: 10.1016/j.paerosci.2021.100758.

[22] L. Enrico, H. Samuel, A. Andoh, G. Alessandro, and S. Roberto, "Autonomous Trajectory Optimisation for Intelligent Satellite Systems and Space Traffic Management," *Acta Astronaut,* vol. 194, pp. 185-201, 2022, doi: https://doi.org/10.1016/j.actaastro.2022.01.027.

[23] S. Hilton *et al.*, *Human-Machine System Design for Autonomous Distributed Satellite Operations*. 2020.

[24] M. Vasile, E. Minisci, and K. Tang, "Computational Intelligence in Aerospace Science and Engineering [Guest Editorial]," *IEEE Computational Intelligence Magazine,* vol. 12, pp. 12-13, 11/01 2017, doi: 10.1109/MCI.2017.2742866.

[25] S. Ramasamy and R. Sabatini, "A Unified Approach to Cooperative and Non-Cooperative Sense-and-Avoid," in *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, Denver, CO, USA, 2015: IEEE, pp. 765-773, doi: 10.1109/ICUAS.2015.7152360.

[26] G. Dubos and J. Saleh, "Comparative cost and utility analysis of monolith and fractionated spacecraft using failure and replacement Markov models," *Acta Astronautica - ACTA ASTRONAUT,* vol. 68, pp. 172-184, 02/28 2011, doi: 10.1016/j.actaastro.2010.07.011.

[27] G. F. Dubos and J. H. Saleh, "Comparative cost and utility analysis of monolith and fractionated spacecraft using failure and replacement Markov models," *Acta Astronaut,* vol. 68, no. 1, pp. 172-184, 2011, doi: 10.1016/j.actaastro.2010.07.011.

[28] M. Mosleh, K. Dalili, and B. Heydari, "Distributed or Monolithic? A Computational Architecture Decision Framework," *IEEE Systems Journal,* vol. 12, no. 1, pp. 125-136, 2018, doi: 10.1109/JSYST.2016.2594290.

[29] S. Nag, "Design and evaluation of distributed spacecraft missions for multi-angular Earth observation," 2015.

[30] A. Poghosyan *et al.*, *Unified Classification for Distributed Satellite Systems*. 2018.
[31] R. Langlois, "Modularity in technology and organization," *Journal of Economic Behavior and Organization,* vol. 49, pp. 19-37, 2002.
[32] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences,* vol. 103, no. 23, p. 8577, 2006, doi: 10.1073/pnas.0601602103.
[33] A. Kharrazi, "Resilience," in *Encyclopedia of Ecology (Second Edition)*, B. Fath Ed. Oxford: Elsevier, 2019, pp. 414-418.
[34] K. Ulrich, "The role of product architecture in the manufacturing firm," *Research Policy,* vol. 24, no. 3, pp. 419-440, 1995/05/01/ 1995, doi: https://doi.org/10.1016/0048-7333(94)00775-3.
[35] K. Hölttä-Otto and O. de Weck, "Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints," *Concurrent Engineering,* vol. 15, no. 2, pp. 113-126, 2007/06/01 2007, doi: 10.1177/1063293X07078931.
[36] D. A. Gianetto and B. Heydari, "Network Modularity is essential for evolution of cooperation under uncertainty," *Scientific Reports,* vol. 5, no. 1, p. 9340, 2015/04/07 2015, doi: 10.1038/srep09340.
[37] B. Heydari, M. Mosleh, and K. Dalili, "From Modular to Distributed Open Architectures: A Unified Decision Framework," *Syst. Eng.,* vol. 19, no. 3, pp. 252–266, 2016, doi: 10.1002/sys.21348.
[38] M. Mosleh, K. Dalili, and B. Heydari, "Distributed or Monolithic? A Computational Architecture Decision Framework," *IEEE Systems Journal,* vol. PP, 08/02 2016, doi: 10.1109/JSYST.2016.2594290.
[39] *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT Press, 1999.
[40] C. Araguz, A. Alvaro, I. d. Portillo, K. Root, E. Alarcón, and E. Bou-Balust, "On autonomous software architectures for distributed spacecraft: A Local-Global Policy," in *2015 IEEE Aerospace Conference*, 7-14 March 2015 2015, pp. 1-9, doi: 10.1109/AERO.2015.7119182.
[41] C. Araguz, E. Bou-Balust, and E. Alarcón, "Applying autonomy to distributed satellite systems: Trends, challenges, and future prospects," *Systems Engineering,* vol. 21, no. 5, pp. 401-416, 2018, doi: 10.1002/sys.21428.
[42] R. Preston, "Distributed Satellite Constellations Offer Advantages Over Monolithic Systems.," no. Santa Monica, CA: RAND Corporation, 2004. [Online]. Available: https://www.rand.org/pubs/research_briefs/RB92.html.
[43] M. R. Schoeberl, "The afternoon constellation: a formation of Earth observing systems for the atmosphere and hydrosphere," vol. 1, ed: IEEE, 2002, pp. 354-356 vol.1.
[44] D. Wischert *et al.*, "Conceptual Design of a Mars Constellation for Global Communication Services using Small Satellites," presented at the 71st International Astronautical Congress (IAC), Dubai, United Arab Emirates, 2020.
[45] I. del Portillo Barrios, B. Cameron, and E. Crawley, "A technical comparison of three low earth orbit satellite constellation systems to provide global broadband," *Acta Astronaut,* vol. 159, 03/01 2019, doi: 10.1016/j.actaastro.2019.03.040.
[46] J. G. Walker, "Satellite constellations," *Journal of the British Interplanetary Society,* vol. 37, p. 559, 1984.
[47] J. Guo, D. Maessen, and E. Gill, "Fractionated spacecraft: The new sprout in distributed space systems," presented at the 60th International Astronautical

Congress, Daejeon, Republic of Korea., 2009. [Online]. Available: https://www.researchgate.net/publication/241882204_Fractionated_spacecraft_The_new_sprout_in_distributed_space_systems.
[48] O. Brown and P. Eremenko, "The Value Proposition for Fractionated Space Architectures," *Sciences,* vol. 4, p. 23, 09/01 2006, doi: 10.2514/6.2006-7506.
[49] O. Brown and P. Eremenko, "Fractionated Space Architectures: A Vision for Responsive Space," 04/01, 2006. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA504007.
[50] J. Lafleur and J. Saleh, "GT-FAST: A Point Design Tool for Rapid Fractionated Spacecraft Sizing and Synthesis," in *AIAA SPACE 2009 Conference & Exposition*, (AIAA SPACE Forum: American Institute of Aeronautics and Astronautics, 2009.
[51] I. d. Portillo, E. Bou, E. Alarcón, M. Sanchez-Net, D. Selva, and Á. Á, "On scalability of Fractionated Satellite Network architectures," in *2015 IEEE Aerospace Conference*, 7-14 March 2015 2015, pp. 1-13, doi: 10.1109/AERO.2015.7119143.
[52] B. Yaglioglu and J. Wang, "Cluster Flying Configuration Evaluation in the Case of Fractionated Spacecraft Architecture," 07/10 2010.
[53] W. Yao, X. Chen, Y. Zhao, and M. van Tooren, "A Fractionated Spacecraft System Assessment Tool Based on Lifecycle Simulation Under Uncertainty," *Chinese Journal of Aeronautics,* vol. 25, no. 1, pp. 71-82, 2012/02/01/ 2012, doi: https://doi.org/10.1016/S1000-9361(11)60364-6.
[54] A. Golkar and I. Lluch i Cruz, "The Federated Satellite Systems paradigm: Concept and business case evaluation," *Acta Astronaut,* vol. 111, pp. 230-248, 2015, doi: 10.1016/j.actaastro.2015.02.009.
[55] I. Lluch and A. Golkar, "Design Implications for Missions Participating in Federated Satellite Systems," *Journal of Spacecraft and Rockets,* vol. 52, no. 5, pp. 1361-1374, 2015, doi: 10.2514/1.A33172.
[56] J. A. Ruiz-de-Azua *et al.*, "Proof-of-Concept of a Federated Satellite System Between Two 6-Unit CubeSats for Distributed Earth Observation Satellite Systems," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 28 July-2 Aug. 2019 2019, pp. 8871-8874, doi: 10.1109/IGARSS.2019.8900099.
[57] R. Akhtyamov, R. Vingerhoeds, and A. Golkar, "Identifying Retrofitting Opportunities for Federated Satellite Systems," *Journal of Spacecraft and Rockets,* vol. 56, pp. 620-629, 05/01 2019, doi: 10.2514/1.A34196.
[58] J. An *et al.*, "Path Planning for Self-Collision Avoidance of Space Modular Self-Reconfigurable Satellites," *Aerospace,* vol. 9, no. 3, p. 141, 2022. [Online]. Available: https://www.mdpi.com/2226-4310/9/3/141.
[59] H. Bloom and D. S. Wilson, "The Lucifer Principle: A Scientific Expedition into the Forces of History," *Foreign Affairs,* vol. 74, 01/01 1995, doi: 10.2307/20047215.
[60] F. Heylighen, "Stigmergy as a universal coordination mechanism I: Definition and components," *Cognitive Systems Research,* vol. 38, pp. 4-13, 2016/06/01/ 2016, doi: https://doi.org/10.1016/j.cogsys.2015.12.002.
[61] S. Cannon, J. Daymude, D. Randall, and A. Richa, *A Markov Chain Algorithm for Compression in Self-Organizing Particle Systems*. 2016, pp. 279-288.
[62] S. Engelen, E. Gill, and C. Verhoeven, "Systems engineering challenges for satellite swarms," *2011 Aerospace Conference,* pp. 1-8, 2011.

[63] Z. Chen and Y. Zeng, "A Swarm Intelligence Networking Framework for Small Satellite Systems," *Communications and Network,* vol. 05, pp. 171-175, 01/01 2013, doi: 10.4236/cn.2013.53B2033.
[64] N. Olsen *et al.*, "The Swarm Satellite Constellation Application and Research Facility (SCARF) and Swarm data products," *Earth, Planets and Space,* vol. 65, no. 11, p. 1, 2013/11/22 2013, doi: 10.5047/eps.2013.07.001.
[65] S. Mathavaraj and R. Padhi, *Satellite Formation Flying: High Precision Guidance Using Optimal and Adaptive Control Techniques*. Singapore: Springer Singapore Pte. Limited, 2021.
[66] D. Wang, B. Wu, and E. K. Poh, *Satellite formation flying : relative dynamics, formation design, fuel optimal maneuvers and formation maintenance* (Intelligent Systems, Control and Automation: Science and Engineering, 87). Singapore: Springer Singapore, 2017.
[67] J. A. Roberts, "Satellite formation flying for an interferometry mission," 2005.
[68] CNES, "PRISMA: Testing new formation-flying technologies," 2016. [Online]. Available: https://prismaffiord.cnes.fr/en/PRISMA/index.htm
[69] K. Thangavel *et al.*, "A Distributed Satellite System for Multibaseline AT-InSAR: Constellation of Formations for Maritime Domain Awareness Using Autonomous Orbit Control," *Aerospace,* vol. 10, no. 2, p. 176, 2023. [Online]. Available: https://www.mdpi.com/2226-4310/10/2/176.
[70] J.-G. Meß, F. Dannemann, and F. Greif, *Techniques of Artificial Intelligence for Space Applications - A Survey*. 2019.
[71] A. Jonsson, R. Morris, and L. Pedersen, *Autonomy in Space Exploration: Current Capabilities and Future Challenges*. 2007, pp. 1-12.
[72] N. J. Nilsson, "Shakey The Robot," AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, 323, 1984. [Online]. Available: http://www.ai.sri.com/pubs/files/629.pdf
[73] E. Vassev and M. Hinchey, "Engineering Requirements for Autonomy Features," in *Software Engineering for Collective Autonomic Systems: The ASCENS Approach*, M. Wirsing, M. Hölzl, N. Koch, and P. Mayer Eds. Cham: Springer International Publishing, 2015, pp. 379-403.
[74] M. Freed, P. Bonasso, M. Ingham, D. Kortenkamp, B. Pell, and J. Penix, "Trusted Autonomy for Spaceflight Systems," in *1st Space Exploration Conference: Continuing the Voyage of Discovery*.
[75] R. Proud, J. Hart, and R. Mrozinski, "Methods for Determining the Level of Autonomy to Design into a Human Spaceflight Vehicle: A Function Specific Approach," p. 15, 09/01 2003.
[76] F. Kucinskis and M. Ferreira, "Taking the ECSS Autonomy Concepts One Step Further," in *SpaceOps 2010 Conference*.
[77] R. Boyce and D. Griffin, "Future Trusted Autonomous Space Scenarios," in *Foundations of Trusted Autonomy*, H. A. Abbass, J. Scholz, and D. J. Reid Eds. Cham: Springer International Publishing, 2018, pp. 355-364.
[78] N. Steiner and P. Athanas, *Hardware autonomy and space systems*. 2009, pp. 1-13.
[79] J. Valasek and T. C. Lieuwen, *Advances in computational intelligence and autonomy for aerospace systems* (Computational intelligence and autonomy for aerospace systems). Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2018.

[80] E. Allouis, Y. Gao, A. Kisdi, R. Ward, and D. Jones, "UK RAS White Papers - Space Robotics & Autonomous Systems: Widening the horizon of space exploration," 06/15 2016.
[81] J. Beer, A. Fisk, and W. Rogers, "Toward a Framework for Levels of Robot Autonomy in Human-Robot Interaction," *Journal of Human-Robot Interaction,* vol. 3, p. 74, 06/01 2014, doi: 10.5898/JHRI.3.2.Beer.
[82] Y. Lim, A. Gardi, and R. Sabatini, "UAS Human Factors and Human-Machine Interface Design," 2020, pp. 23-48.
[83] Y. Lim, S. Ramasamy, A. Gardi, T. Kistan, and R. Sabatini, "Cognitive Human-Machine Interfaces and Interactions for Unmanned Aircraft," *J Intell Robot Syst,* 10/13 2017, doi: 10.1007/s10846-017-0648-9.
[84] S. Bijjahalli, R. Sabatini, and A. Gardi, "Advances in intelligent and autonomous navigation systems for small UAS," *Progress in Aerospace Sciences,* vol. 115, p. 100617, 2020/05/01/ 2020, doi: https://doi.org/10.1016/j.paerosci.2020.100617.
[85] N. Pongsakornsathien *et al.*, "Sensor Networks for Aerospace Human-Machine Systems," *Sensors,* vol. 19, p. 3465, 08/08 2019, doi: 10.3390/s19163465.
[86] Y. Lim, A. Gardi, R. Sabatini, S. Ramasamy, and R. Bolia, "Avionics Human-Machine Interfaces and Interactions for Manned and Unmanned Aircraft," *Progress in Aerospace Sciences,* 08/01 2018, doi: 10.1016/j.paerosci.2018.05.002.
[87] Y. Lim, V. Bassien-Capsa, S. Ramasamy, J. Liu, and R. Sabatini, "Commercial Airline Single-Pilot Operations: System Design and Pathways to Certification," *IEEE Aerospace and Electronic Systems Magazine,* vol. 32, pp. 4-21, 09/19 2017, doi: 10.1109/MAES.2017.160175.
[88] W. Xu, M. J. Dainoff, L. Ge, and Z. Gao, "From Human-Computer Interaction to Human-AI Interaction: New Challenges and Opportunities for Enabling Human-Centered AI," *arXiv preprint arXiv:2105.05424,* 2021.
[89] S. Chien and R. Morris, "Space Applications of Artificial Intelligence," *Ai Magazine,* vol. 35, pp. 3-6, 12/01 2014, doi: 10.1609/aimag.v35i4.2551.
[90] A. Argan, M. Tavani, and A. Trois, "The on-board data processing of the AGILE satellite," *Rendiconti Lincei. Scienze Fisiche e Naturali,* vol. 30, 10/19 2019, doi: 10.1007/s12210-019-00846-0.
[91] P. Buist and B.-J. Vollmuller, "On-board Data Processing for Earth and Atmospheric Observations," *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES, AEROSPACE TECHNOLOGY JAPAN,* vol. 12, pp. Tn_25-Tn_30, 01/01 2014, doi: 10.2322/tastj.12.Tn_25.
[92] T. Itoh *et al.*, "The on-board data processing system for scientific satellite - EXOS-C," 09/01 1980.
[93] M. Tragni, C. Abbattista, L. Amoruso, L. Cinquepalmi, F. Bgongiari, and W. Errico, *On-board Payload Data Processing from Earth to Space Segment*. 2013.
[94] Firdaus, Y. Arkeman, A. Buono, and I. Hermadi, "Satellite image processing for precision agriculture and agroindustry using convolutional neural network and genetic algorithm," *IOP Conference Series: Earth and Environmental Science,* vol. 54, p. 012102, 01/01 2017, doi: 10.1088/1755-1315/54/1/012102.
[95] W. G.G and K. Charalabos, *Application of Artificial Intelligence in Land Use Mapping from Satellite Imagery*. 1990.
[96] D. Hemanth, *Artificial Intelligence Techniques for Satellite Image Analysis*. 2020.

[97] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep Convolutional Neural Networks for Hyperspectral Image Classification," *Journal of Sensors,* vol. 2015, p. 258619, 2015/07/30 2015, doi: 10.1155/2015/258619.

[98] G. J.-P. Schumann, L. Giustarini, M. Zare, and B. Gaffinet, "Call to action: Pushing scientific and technological innovation to develop an efficient AI flood mapper for operational SAR satellites," in *EGU General Assembly Conference Abstracts*, 2021, pp. EGU21-5943.

[99] J. Sharma, P. Varma, Y. Verma, and M. Tank, "Automated Land Classification using AI/ML."

[100] D. A. Zaidenberg, A. Sebastianelli, D. Spiller, B. Le Saux, and S. L. Ullo, "Advantages and Bottlenecks of Quantum Machine Learning for Remote Sensing," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021: IEEE, pp. 5680-5683.

[101] A. Nowakowski *et al.*, "AI Opportunities and Challenges for Crop Type Mapping Using Sentinel-2 and Drone Data," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021: IEEE, pp. 258-261.

[102] A. Nowakowski *et al.*, "Crop type mapping by using transfer learning," *International Journal of Applied Earth Observation and Geoinformation,* vol. 98, p. 102313, 2021.

[103] D. Spiller, L. Ansalone, F. Carotenuto, and P. P. Mathieu, "Crop Type Mapping Using Prisma Hyperspectral Images and One-Dimensional Convolutional Neural Network," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021: IEEE, pp. 8166-8169.

[104] A. Sebastianelli, D. A. Zaidenberg, D. Spiller, B. L. Saux, and S. L. Ullo, "On Circuit-based Hybrid Quantum Neural Networks for Remote Sensing Imagery Classification," *arXiv preprint arXiv:2109.09484,* 2021.

[105] L. Shi, J. Zhang, D. Zhang, T. Igbawua, and Y. Liu, "Developing a dust storm detection method combining Support Vector Machine and satellite data in typical dust regions of Asia," *Advances in Space Research,* vol. 65, no. 4, pp. 1263-1278, 2020/02/15/ 2020, doi: https://doi.org/10.1016/j.asr.2019.11.027.

[106] P. V. R. Ferreira *et al.*, "Reinforcement Learning for Satellite Communications: From LEO to Deep Space Operations," *IEEE Communications Magazine,* vol. 57, no. 5, pp. 70-75, 2019, doi: 10.1109/MCOM.2019.1800796.

[107] P. V. R. Ferreira *et al.*, "Multi-objective reinforcement learning-based deep neural networks for cognitive space communications," in *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*, 27-28 June 2017 2017, pp. 1-8, doi: 10.1109/CCAAW.2017.8001880.

[108] T. M. Hackett, S. G. Bilén, P. V. R. Ferreira, A. M. Wyglinski, R. C. Reinhart, and D. J. Mortensen, "Implementation and On-Orbit Testing Results of a Space Communications Cognitive Engine," *IEEE Transactions on Cognitive Communications and Networking,* vol. 4, no. 4, pp. 825-842, 2018, doi: 10.1109/TCCN.2018.2878202.

[109] F. Fourati and M.-S. Alouini, "Artificial intelligence for satellite communication: A review," *arXiv preprint arXiv:2101.10899,* 2021.

[110] H. Dahrouj *et al.*, "An Overview of Machine Learning-Based Techniques for Solving Optimization Problems in Communications and Signal Processing," *IEEE Access,* 2021.

[111] A. Anzagira, W. W. Edmonson, and D. N. Amanor, "LED-Based Visible Light Inter-Satellite Communication for Distributed Space Systems," *IEEE Journal on Miniaturization for Air and Space Systems,* 2021.

[112] P. Guan, X.-J. Liu, and J.-Z. Liu, "Adaptive fuzzy sliding mode control for flexible satellite," *Engineering Applications of Artificial Intelligence,* vol. 18, no. 4, pp. 451-459, 2005/06/01/ 2005, doi: https://doi.org/10.1016/j.engappai.2004.11.003.

[113] W. MacKunis, F. Leve, P. M. Patre, N. Fitz-Coy, and W. E. Dixon, "Adaptive neural network-based satellite attitude control in the presence of CMG uncertainty," *Aerospace Science and Technology,* vol. 54, pp. 218-228, 2016/07/01/ 2016, doi: https://doi.org/10.1016/j.ast.2016.04.022.

[114] D. Lee, "Guidance, navigation and control system for autonomous proximity operations and docking of spacecraft," 01/01 2009.

[115] S. Lim, M. Stoeckle, B. J. Streetman, and M. Neave, "Markov Neural Network For Guidance, Navigation and Control," in *AIAA Scitech 2020 Forum*, (AIAA SciTech Forum: American Institute of Aeronautics and Astronautics, 2020.

[116] Q. Yao, "Neural adaptive attitude tracking control for uncertain spacecraft with preassigned performance guarantees," *Advances in Space Research,* 2021/10/18/ 2021, doi: https://doi.org/10.1016/j.asr.2021.10.022.

[117] Q. Yao, "Adaptive fuzzy neural network control for a space manipulator in the presence of output constraints and input nonlinearities," *Advances in Space Research,* vol. 67, no. 6, pp. 1830-1843, 2021/03/15/ 2021, doi: https://doi.org/10.1016/j.asr.2021.01.001.

[118] M. Zheng, J. Luo, and Z. Dang, "Feedforward neural network based time-varying state-transition-matrix of Tschauner-Hempel equations," *Advances in Space Research,* 2021/10/20/ 2021, doi: https://doi.org/10.1016/j.asr.2021.10.023.

[119] R. Lungu, M. Lungu, and C. Efrim, "Adaptive control of DGMSCMG using dynamic inversion and neural networks," *Advances in Space Research,* vol. 68, no. 8, pp. 3478-3494, 2021/10/15/ 2021, doi: https://doi.org/10.1016/j.asr.2021.06.018.

[120] E. Sin, S. Nag, V. Ravindra, A. Li, and M. Arcak, *Attitude Trajectory Optimization for Agile Satellites in Autonomous Remote Sensing Constellation*. 2021.

[121] S. Abdelghafar, A. Darwish, and A. E. Hassanien, "Intelligent Health Monitoring Systems for Space Missions Based on Data Mining Techniques," 2020, pp. 65-78.

[122] C.-P. Fritzen, "Vibration-Based Structural Health Monitoring – Concepts and Applications," *Key Engineering Materials - KEY ENG MAT,* vol. 293, 01/01 2005, doi: 10.4028/www.scientific.net/KEM.293-294.3.

[123] S. Fuertes, G. Picart, J.-Y. Tourneret, L. Chaari, A. Ferrari, and C. Richard, *Improving Spacecraft Health Monitoring with Automatic Anomaly Detection Techniques*. 2016.

[124] Nasa.gov, "Integrated Vehicle Health Management (IVHM)." [Online]. Available: https://www.nasa.gov/centers/ames/research/humaninspace/humansinspace-ivhm.html.

[125] K. Penta and D. Khemani, *Satellite Health Monitoring Using CBR Framework*. 2004, pp. 732-747.

[126] M. Tipaldi and B. Bruenjes, *Spacecraft Health Monitoring and Management Systems*. 2014, pp. 68-72.

[127] L. Elerin, C. Learoyd, and B. Wilson, "Applying neural networks and other AI techniques to fault detection in satellite communication systems," in *Neural Networks*

for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop, 1997: IEEE, pp. 617-625.

[128] A. Malik, B. Moster, and C. Obermeier, *Exoplanet Detection using Machine Learning*. 2020.

[129] T. Estlin, F. Fisher, D. Mutz, and S. Chien, *Automated planning for a Deep Space Communications Station*. 1999, pp. 1410-1417 vol.2.

[130] F. Fisher, T. Estlin, D. Mutz, and S. Chien, "An AI Approach to Ground Station Automomy for Deep Space Communications," vol. 440, p. 191, 07/31 1999.

[131] F. Fisher, R. Knight, B. Engelhardt, S. Chien, and N. Alejandre, *A planning approach to monitor and control for deep space communications*. 2000, pp. 311-320 vol.2.

[132] D. Izzo, C. Sprague, and D. Tailor, "Machine learning and evolutionary techniques in interplanetary trajectory design," 02/01 2018.

[133] M. Agnan and J. Vannitsen, "Scaling uncertainties on asteroid characteristics to prepare datasets for machine learning," *Advances in Space Research*, vol. 68, no. 8, pp. 3225-3232, 2021/10/15/ 2021, doi: https://doi.org/10.1016/j.asr.2021.06.007.

[134] B. Gaudet, R. Linares, and R. Furfaro, "Deep reinforcement learning for six degree-of-freedom planetary landing," *Advances in Space Research*, vol. 65, no. 7, pp. 1723-1741, 2020/04/01/ 2020, doi: https://doi.org/10.1016/j.asr.2019.12.030.

[135] R. Sherwood, S. Chien, R. Castaño, and G. Rabideau, *Autonomous Planning and Scheduling on the TechSat 21 Mission*. 2002, pp. 213-224.

[136] A. A. Bataleblu and J. Roshanian, "Robust trajectory optimization of space launch vehicle using computational intelligence," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, 25-28 May 2015 2015, pp. 3418-3425, doi: 10.1109/CEC.2015.7257318.

[137] D. S. Kolosa, "A Reinforcement Learning Approach to Spacecraft Trajectory Optimization," 2019.

[138] K. Parmar and D. Guzzetti, "Interactive imitation learning for spacecraft path-planning in binary asteroid systems," *Advances in Space Research*, vol. 68, no. 4, pp. 1928-1951, 2021/08/15/ 2021, doi: https://doi.org/10.1016/j.asr.2021.04.023.

[139] S. Yin, J. Li, and L. Cheng, "Low-thrust spacecraft trajectory optimization via a DNN-based method," *Advances in Space Research*, vol. 66, no. 7, pp. 1635-1646, 2020/10/01/ 2020, doi: https://doi.org/10.1016/j.asr.2020.05.046.

[140] O. Jung, J. Seong, Y. Jung, and H. Bang, "Recurrent neural network model to predict re-entry trajectories of uncontrolled space objects," *Advances in Space Research*, vol. 68, no. 6, pp. 2515-2529, 2021/09/15/ 2021, doi: https://doi.org/10.1016/j.asr.2021.04.041.

[141] I. P. Rodrigues, P. A. S. Oliveira, A. M. Ambrosio, and R. A. J. Chagas, "Modeling satellite battery aging for an operational satellite simulator," *Advances in Space Research*, vol. 67, no. 6, pp. 1981-1999, 2021/03/15/ 2021, doi: https://doi.org/10.1016/j.asr.2020.12.031.

[142] S. Hilton, F. Cairola, A. Gardi, R. Sabatini, N. Pongsakornsathien, and N. Ezer, "Uncertainty Quantification for Space Situational Awareness and Traffic Management," *Sensors*, vol. 19, p. 4361, 10/09 2019, doi: 10.3390/s19204361.

[143] S. Hilton, A. Gardi, F. Cairola, and R. Sabatini, *A Sensor-Centric Approach to Space Traffic Management*. 2019, pp. 584-589.

[144] S. Hilton, R. Sabatini, A. Gardi, H. Ogawa, and P. Teofilatto, "Space Traffic Management: Towards Safe and Unsegregated Space Transport Operations," *Progress in Aerospace Sciences,* vol. 105, 02/01 2019, doi: 10.1016/j.paerosci.2018.10.006.

[145] E. Blasch, D. Shen, G. Chen, and J. Sung, "Multisource AI scorecard table analysis of AMIGO," in *Sensors and Systems for Space Applications XIV*, 2021, vol. 11755: International Society for Optics and Photonics, p. 117550B.

[146] S. Nag, D. D. Murakami, N. A. Marker, M. T. Lifson, and P. H. Kopardekar, "Prototyping Operational Autonomy for Space Traffic Management," *Acta Astronautica,* vol. 180, pp. 489-506, 2021.

[147] S. J. Siddiqi, F. Naeem, S. Khan, K. S. Khan, and M. Tariq, "Towards AI-enabled traffic management in multipath TCP: A survey," *Computer Communications,* 2021.

[148] J. Zhang, D. Ye, J. D. Biggs, and Z. Sun, "Finite-time relative orbit-attitude tracking control for multi-spacecraft with collision avoidance and changing network topologies," *Advances in Space Research,* vol. 63, no. 3, pp. 1161-1175, 2019/02/01/ 2019, doi: https://doi.org/10.1016/j.asr.2018.10.037.

[149] K. Thangavel, A. Gardi, S. Hilton, A. Afful, and R. Sabatini, *Towards Multi-Domain Traffic Management*. 2021.

[150] J. Foust, "SpaceX's space-Internet woes: Despite technical glitches, the company plans to launch the first of nearly 12,000 satellites in 2019," *IEEE Spectrum,* vol. 56, pp. 50-51, 01/01 2019, doi: 10.1109/MSPEC.2019.8594798.

[151] T. Yairi, N. Takeishi, T. Oda, Y. Nakajima, N. Nishimura, and N. Takata, "A Data-Driven Health Monitoring Method for Satellite Housekeeping Data Based on Probabilistic Clustering and Dimensionality Reduction," *IEEE Transactions on Aerospace and Electronic Systems,* vol. PP, pp. 1-1, 02/20 2017, doi: 10.1109/TAES.2017.2671247.

[152] A. Young, C. Kitts, M. Neumann, I. Mas, and M. Rasay, "Initial Flight Results for an Automated Satellite Beacon Health Monitoring Network," 01/01 2010.

[153] G. Landis, S. Bailey, and R. Tischler, "Causes of Power-Related Satellite Failures," *2006 IEEE 4th World Conference on Photovoltaic Energy Conference,* vol. 2, pp. 1943-1945, 2006.

[154] P. Grandjean, T. Pesquet, A. M. M. Muxi, and M. C. Charmeau, "What on-Board Autonomy Means for Ground Operations: An Autonomy Demonstrator Conceptual Design," in *Space OPS 2004 Conference*.

[155] A. Börner *et al.*, "Sensor Artificial Intelligence and its Application to Space Systems -- A White Paper," 2020.

[156] O. Kodheli *et al.*, *Satellite Communications in the New Space Era: A Survey and Future Challenges*. 2020.

[157] L. Pellaco, N. Singh, and J. Jaldén, *Spectrum Prediction and Interference Detection for Satellite Communications*. 2019.

[158] A. Vazquez *et al.*, *On the Use of AI for Satellite Communications*. 2020.

[159] M. Bappy, R. Huq, and S. Siddique, *AI-OBC: Conceptual Design of a Deep Neural Network based Next Generation Onboard Computing Architecture for Satellite Systems*. 2019.

[160] V. Kothari, E. Liberis, and N. Lane, *The Final Frontier: Deep Learning in Space*. 2020.

[161] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, 2009.

[162] S. Voß, "MetaheuristicsMetaheuristics," in *Encyclopedia of Optimization*, C. A. Floudas and P. M. Pardalos Eds. Boston, MA: Springer US, 2009, pp. 2061-2075.

[163] S. Voß, "Meta-heuristics: The state of the art," in *Workshop on Local Search for Planning and Scheduling*, 2000: Springer, pp. 1-23.

[164] X.-S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*. 2010.

[165] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences,* vol. 237, pp. 82-117, 2013/07/10/ 2013, doi: https://doi.org/10.1016/j.ins.2013.02.041.

[166] S. Almufti, "Historical survey on metaheuristics algorithms," vol. 7, pp. 1-12, 11/17 2019, doi: 10.14419/ijsw.v7i1.29497.

[167] S. Harifi, J. Mohammadzadeh, M. Khalilian, and S. Ebrahimnejad, "Giza Pyramids Construction: an ancient-inspired metaheuristic algorithm for optimization," *Evolutionary Intelligence,* 2020/07/13 2020, doi: 10.1007/s12065-020-00451-3.

[168] G. Dhiman and A. Kaur, "Optimizing the Design of Airfoil and Optical Buffer Problems Using Spotted Hyena Optimizer," *Designs,* vol. 2, p. 28, 08/01 2018, doi: 10.3390/designs2030028.

[169] A. Hegazy, M. Makhlouf, and G. Eltaweel, "Dimensionality Reduction Using an Improved Whale Optimization Algorithm for Data Classification," *International Journal of Modern Education and Computer Science,* vol. 10, pp. 37-49, 07/08 2018, doi: 10.5815/ijmecs.2018.07.04.

[170] X.-S. Yang, "Firefly Algorithms for Multimodal Optimization," Berlin, Heidelberg, 2009: Springer Berlin Heidelberg, in Stochastic Algorithms: Foundations and Applications, pp. 169-178.

[171] D. Spiller, "Optimal control problems solved via swarm intelligence," 2018.

[172] H. H. Hoos and T. Stützle, "1 - INTRODUCTION," in *Stochastic Local Search*, H. H. Hoos and T. Stützle Eds. San Francisco: Morgan Kaufmann, 2005, pp. 13-59.

[173] M. Mandloi and V. Bhatia, "Chapter 12 - Symbol Detection in Multiple Antenna Wireless Systems via Ant Colony Optimization," in *Handbook of Neural Computation*, P. Samui, S. Sekhar, and V. E. Balas Eds.: Academic Press, 2017, pp. 225-237.

[174] M. Dorigo, M. Birattari, and T. Stützle, "Ant Colony Optimization," *Computational Intelligence Magazine, IEEE,* vol. 1, pp. 28-39, 12/01 2006, doi: 10.1109/MCI.2006.329691.

[175] Y. Eren, İ. B. Küçükdemiral, and İ. Üstoğlu, "Chapter 2 - Introduction to Optimization," in *Optimization in Renewable Energy Systems*, O. Erdinç Ed. Boston: Butterworth-Heinemann, 2017, pp. 27-74.

[176] J. Kennedy, R. C. Eberhart, and Y. Shi, "chapter seven - The Particle Swarm," in *Swarm Intelligence*, J. Kennedy, R. C. Eberhart, and Y. Shi Eds. San Francisco: Morgan Kaufmann, 2001, pp. 287-325.

[177] Y. Zhang, S. Wang, and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," *Mathematical Problems in Engineering,* vol. 2015, p. 931256, 2015/10/07 2015, doi: 10.1155/2015/931256.

[178] D. Spiller, R. Melton, and F. Curti, "Inverse dynamics particle swarm optimization applied to constrained minimum-time maneuvers using reaction wheels," *Aerospace Science and Technology,* vol. 75, 04/01 2018, doi: 10.1016/j.ast.2017.12.038.

[179] D. Spiller and F. Curti, *Inverse-dynamics Particle Swarm Optimization for Real Time Optimal Control: Challenges and Opportunities*. 2018.

[180] J. Kennedy and R. Eberhard, "Particle swarm optimization," *Phys. Rev., B.*, vol. 13, pp. 5344-5348, 01/01 2010.

[181] M. Clerc, *Particle Swarm Optimization*, 1. Aufl. ed. London: London: Wiley-ISTE, 2006.

[182] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 27 Nov.-1 Dec. 1995 1995, vol. 4, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[183] K. Parsopoulos and M. Vrahatis, *Particle Swarm Optimization and Intelligence: Advances and Applications*. 2010.

[184] M. Calvo, J. I. Montijano, and L. Randez, "A fifth-order interpolant for the Dormand and Prince Runge-Kutta method," *Journal of Computational and Applied Mathematics*, vol. 29, no. 1, pp. 91-100, 1990/01/10/ 1990, doi: https://doi.org/10.1016/0377-0427(90)90198-9.

[185] X. Zhang, Z. Chen, and Y. Liu, "Chapter 3 - The Material Point Method," in *The Material Point Method*, X. Zhang, Z. Chen, and Y. Liu Eds. Oxford: Academic Press, 2017, pp. 37-101.

[186] H. Shakhatreh *et al.*, "Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges," *IEEE Access*, vol. 7, pp. 48572-48634, 2019, doi: 10.1109/ACCESS.2019.2909530.

[187] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Computer Science*, vol. 132, pp. 679-688, 2018/01/01/ 2018, doi: https://doi.org/10.1016/j.procs.2018.05.069.

[188] J. P. A. Y. V. N. Y. A. Govinda Rao Locharla, "Review of the Convolution Neural Network Architectures for Deep Learning," *International Journal of Advanced Science and Technology*, vol. 29, no. 04, pp. 2251 – 2262, 05/13 2020. [Online]. Available: http://sersc.org/journals/index.php/IJAST/article/view/19246.

[189] G. Furano *et al.*, "Towards the Use of Artificial Intelligence on the Edge in Space Systems: Challenges and Opportunities," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 12, pp. 44-56, 2020, doi: 10.1109/MAES.2020.3008468.

[190] S. Di Mascio, A. Menicucci, E. Gill, G. Furano, and C. Monteleone, "Leveraging the Openness and Modularity of RISC-V in Space," *Journal of Aerospace Information Systems*, pp. 1-19, 08/05 2019, doi: 10.2514/1.I010735.

[191] G. Furano, A. Tavoularis, and M. Rovatti, "AI in space: applications examples and challenges," in *2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 19-21 Oct. 2020 2020, pp. 1-6, doi: 10.1109/DFT50435.2020.9250908.

[192] T. G. Reid, A. M. Neish, T. F. Walter, and P. K. Enge, "Leveraging commercial broadband leo constellations for navigating," in *Proceedings of the 29th International Technical Meeting of the Satellite Division of the Institute of Navigation (Ion Gnss+ 2016). 29th International Technical Meeting of the Satellite Division of the Institute of Navigation (Ion Gnss+ 2016), Portland, Oregon*, 2016, vol. 12, pp. 2016-16.09.

[193] W. A. CROSSLEY∗ and E. A. Williams, "Simulated annealing and genetic algorithm approaches for discontinuous coverage satellite constellation design," *Engineering Optimization+ A35*, vol. 32, no. 3, pp. 353-371, 2000.

[194] E. Frayssinhes, "Investigating new satellite constellation geometries with genetic algorithms," in *Astrodynamics Conference*, 1996, p. 3636.

[195] Q. He and C. Han, "Satellite constellation design with adaptively continuous ant system algorithm," *Chinese Journal of Aeronautics,* vol. 20, no. 4, pp. 297-303, 2007.
[196] T. Savitri, Y. Kim, S. Jo, and H. Bang, "Satellite constellation orbit design optimization with combined genetic algorithm and semianalytical approach," *International Journal of Aerospace Engineering,* vol. 2017, 2017.
[197] R. Pandi Perumal, H. Voos, F. Dalla Vedova, and H. Moser, "Comparison of Multidisciplinary Design Optimization Architectures for the design of Distributed Space Systems," in *Proceedings of the 71st International Astronautical Congress 2020*, 2020.
[198] R. P. Perumal, "Development of a Decision Support System for Incorporating Risk Assessments during the System Design of Microsatellites - Pandi Perumal Raja," Doctoral dissertation, University of Luxembourg, https://orbilu.uni.lu/handle/10993/48261, 2021.
[199] N. M. Alexandrov and M. Y. Hussaini, "Multidisciplinary design optimization: State of the art," 1997.
[200] J. Sobieszczanski-Sobieski, "Multidisciplinary design optimization: an emerging new engineering discipline," in *Advances in structural optimization*: Springer, 1995, pp. 483-496.
[201] J. R. Martins and A. B. Lambe, "Multidisciplinary design optimization: a survey of architectures," *AIAA journal,* vol. 51, no. 9, pp. 2049-2075, 2013.
[202] T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree, "Kriging models for global approximation in simulation-based multidisciplinary design optimization," *AIAA journal,* vol. 39, no. 12, pp. 2233-2241, 2001.
[203] B. Chell, S. Hoffenson, C. J. Philippe, and M. R. Blackburn, "Comparing Filtering Multifidelity Optimization Strategies With a Simulation-Based Multidisciplinary Aircraft Model," *Journal of Mechanical Design,* vol. 143, no. 8, p. 081702, 2021.
[204] N. F. Brown and J. R. Olds, "Evaluation of multidisciplinary optimization techniques applied to a reusable launch vehicle," *Journal of Spacecraft and Rockets,* vol. 43, no. 6, pp. 1289-1300, 2006.
[205] J. Gray, K. T. Moore, T. A. Hearn, and B. A. Naylor, "Standard platform for benchmarking multidisciplinary design analysis and optimization architectures," *AIAA journal,* vol. 51, no. 10, pp. 2380-2394, 2013.
[206] K. Hulme and C. Bloebaum, "A simulation-based comparison of multidisciplinary design optimization solution strategies using CASCADE," *Structural and Multidisciplinary Optimization,* vol. 19, no. 1, pp. 17-35, 2000.
[207] S. Kodiyalam, *Evaluation of methods for multidisciplinary design optimization (MDO), Phase I*. National Aeronautics and Space Administration, Langley Research Center …, 1998.
[208] S. Kodiyalam and C. Yuan, *Evaluation of methods for multidisciplinary design optimization (MDO), Part II*. Citeseer, 2000.
[209] R. Perez, H. Liu, and K. Behdinan, "Evaluation of multidisciplinary optimization approaches for aircraft conceptual design," in *10th AIAA/ISSMO multidisciplinary analysis and optimization conference*, 2004, p. 4537.
[210] N. P. Tedford and J. R. Martins, "Benchmarking multidisciplinary design optimization algorithms," *Optimization and Engineering,* vol. 11, no. 1, pp. 159-183, 2010.

[211] S.-I. Yi, J.-K. Shin, and G. Park, "Comparison of MDO methods with mathematical examples," *Structural and Multidisciplinary Optimization,* vol. 35, no. 5, pp. 391-402, 2008.

[212] J. S. Gray, J. T. Hwang, J. R. Martins, K. T. Moore, and B. A. Naylor, "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization,* vol. 59, no. 4, pp. 1075-1104, 2019.

[213] S. Hilton, A. Gardi, R. Sabatini, N. Ezer, and S. Desai, "Human-Machine System Design for Autonomous Distributed Satellite Operations," in *IEEE/AIAA 39th Digital Avionics Systems Conference, DASC 2020*, San Antonio, TX, USA, 2020: IEEE.

[214] S. Hilton, R. Sabatini, A. Gardi, H. Ogawa, and P. Teofilatto, "Space traffic management: towards safe and unsegregated space transport operations," (in English), *Progress in Aerospace Sciences,* Review vol. 105, pp. 98-125, Feb 2019, doi: 10.1016/j.paerosci.2018.10.006.

[215] Y.-K. Chang, K.-L. Hwang, and S.-J. Kang, "SEDT (System Engineering Design Tool) development and its application to small satellite conceptual design," *Acta Astronautica,* vol. 61, no. 7-8, pp. 676-690, 2007.

[216] J. R. Wertz, D. F. Everett, and J. J. Puschell, *Space mission engineering: the new SMAD*. Microcosm Press, 2011.

[217] K. Thangavel, D. Spiller, R. Sabatini, and P. Marzocca, "On-board Data Processing of Earth Observation Data Using 1-D CNN," presented at the SmartSat CRC Conference 2022, 12-13 September 2022.

[218] D. Spiller, K. Thangavel, S. Thottuchirayil Sasidharan, S. Amici, and R. Sabatini, "Wildfire segmentation analysis from edge computing for on-board real-time alerts using hyperspectral imagery," 2022.

[219] H. Tambunan and H. Mawengkang, "Solving Mixed Integer Non-Linear Programming Using Active Constraint," *Global Journal of Pure and Applied Mathematics,* vol. 12, no. 6, pp. 5267-5281, 2016.

[220] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics,* vol. 2, no. 2, pp. 164-168, 1944.

[221] B. S. Schwarz, "Fractionated satellites: a systems engineering analysis," Doctoral, Engineering and the Environment, University of Southampton, 2014. [Online]. Available: https://eprints.soton.ac.uk/370544/

[222] L. Qiao, C. Rizos, and A. Dempster, "Design and Analysis of Satellite Orbits for the Garada Mission," 01/01 2011.

[223] J. G. Walker, "Circular orbit patterns providing continuous whole earth coverage," ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (UNITED KINGDOM), 1970.

[224] J. R. Wertz, "Mission geometry; orbit and constellation design and management," *Space Technology Library,* 2001.

[225] Y. Kozai, "The motion of a close earth satellite," *The Astronomical Journal,* vol. 64, p. 367, 1959.

[226] D. A. Vallado, *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001.

[227] W. J. Larson and J. R. Wertz, "Space mission analysis and design," Torrance, CA (United States); Microcosm, Inc., 1992.

[228] R. Sabatini and M. Richardson, "Novel Atmospheric Extinction Measurement Techniques for Aerospace Laser System Applications," *Infrared Physics & Technology,* vol. 56, pp. 30–50, 01/31 2013, doi: 10.1016/j.infrared.2012.10.002.
[229] R. Sabatini, M. Richardson, H. Jia, and D. Zammit-Mangion, "Airborne Laser Systems for Atmospheric Sounding in the Near Infrared," *Proc SPIE,* p. 33, 05/01 2012, doi: 10.1117/12.915718.
[230] N. Kadhim, M. Mourshed, and M. Bray, "Advances in remote sensing applications for urban sustainability," *Euro-Mediterranean Journal for Environmental Integration,* vol. 1, no. 1, p. 7, 2016/10/18 2016, doi: 10.1007/s41207-016-0007-4.
[231] F. Kucinskis and M. Ferreira, "Taking the ECSS Autonomy Concepts One Step Further," *SpaceOps 2010 Conference,* 04/25 2010, doi: 10.2514/6.2010-2364.
[232] Z. Lingli, S. Juha, L. Jingbin, H. Juha, K. Harri, and H. Henrik, "A Review: Remote Sensing Sensors," in *Multi-purposeful Application of Geospatial Data*, B. R. Rustam, H. Sabina, and H. Z. Mahfuza Eds. Rijeka: IntechOpen, 2017, p. Ch. 2.
[233] K. Thangavel and M. Parisse, *AN ISOTHERMAL ANALYSIS OF 6U POCKET CUBE SATELLITE*. 2019.
[234] R. Shi, L. Liu, T. Long, Y. Wu, and G. G. Wang, "Multidisciplinary modeling and surrogate assisted optimization for satellite constellation systems," *Structural and Multidisciplinary Optimization,* vol. 58, no. 5, pp. 2173-2188, 2018.
[235] E. F. Bruhn, *Analysis and design of flight vehicle structures*. Tri-state offset company, 1965.
[236] A. Ravanbakhsh and S. Franchini, "Multiobjective optimization applied to structural sizing of low cost university-class microsatellite projects," *Acta Astronautica,* vol. 79, pp. 212-220, 2012.
[237] T. P. Sarafin and W. J. Larson, "Spacecraft structures and mechanisms: from concept to launch," 1995.
[238] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta numerica,* vol. 4, pp. 1-51, 1995.
[239] S. Loft, S. Bolland, M. S. Humphreys, and A. Neal, "A Theory and Model of Conflict Detection in Air Traffic Control: Incorporating Environmental Constraints," (in English), *Journal of Experimental Psychology: Applied,* vol. 15, no. 2, pp. 106-124, 2009, doi: 10.1037/a0016118.
[240] R. S. Tol, "The economic effects of climate change," *Journal of economic perspectives,* vol. 23, no. 2, pp. 29-51, 2009.
[241] J. Lubchenco and T. R. Karl, "Extreme weather events," *Phys. Today,* vol. 65, no. 3, p. 31, 2012.
[242] G. Kallis, "Droughts," *Annual review of environment and resources,* vol. 33, pp. 85-118, 2008.
[243] J. Han, H. Dai, and Z. Gu, "Sandstorms and desertification in Mongolia, an example of future climate events: A review," *Environmental Chemistry Letters,* vol. 19, no. 6, pp. 4063-4073, 2021.
[244] R. Lindsey. "Climate change: global sea level." http://arizonaenergy.org/News_17/News_Sep17/ClimateChangeGlobalSeaLevel.html (accessed 24th December 2022.

[245] C. C. Lee, "Utilizing synoptic climatological methods to assess the impacts of climate change on future tornado-favorable environments," *Natural hazards,* vol. 62, no. 2, pp. 325-343, 2012.

[246] A. Robock, "Volcanic eruptions and climate," *Reviews of geophysics,* vol. 38, no. 2, pp. 191-219, 2000.

[247] R. Xu *et al.*, "Wildfires, global climate change, and human health," *New England Journal of Medicine,* vol. 383, no. 22, pp. 2173-2181, 2020.

[248] J. Vukomanovic and T. Steelman, "A systematic review of relationships between mountain wildfire and ecosystem services," *Landscape Ecology,* vol. 34, no. 5, pp. 1179-1194, 2019.

[249] S. E. Finlay, A. Moffat, R. Gazzard, D. Baker, and V. Murray, "Health impacts of wildfires," *PLoS currents,* vol. 4, 2012.

[250] M. A. Tanase, C. Aponte, S. Mermoz, A. Bouvet, T. Le Toan, and M. Heurich, "Detection of windthrows and insect outbreaks by L-band SAR: A case study in the Bavarian Forest National Park," *Remote Sensing of Environment,* vol. 209, pp. 700-711, 2018/05/01/ 2018, doi: https://doi.org/10.1016/j.rse.2018.03.009.

[251] B. Pradhan, M. Dini Hairi Bin Suliman, and M. Arshad Bin Awang, "Forest fire susceptibility and risk mapping using remote sensing and geographical information systems (GIS)," *Disaster Prevention and Management: An International Journal,* vol. 16, no. 3, pp. 344-352, 2007, doi: 10.1108/09653560710758297.

[252] P. Guth, T. Chester, O. Zeyn, Leary, and J. Shotwell, "FIRE LOCATION FROM A SINGLE OSBORNE FIREFINDER AND A DEM," 10/07 2021.

[253] K. Bouabdellah, H. Noureddine, and S. Larbi, "Using Wireless Sensor Networks for Reliable Forest Fires Detection," *Procedia Computer Science,* vol. 19, pp. 794-801, 2013/01/01/ 2013, doi: https://doi.org/10.1016/j.procs.2013.06.104.

[254] P. Barmpoutis, P. Papaioannou, K. Dimitropoulos, and N. Grammalidis, "A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing," (in eng), *Sensors (Basel),* vol. 20, no. 22, Nov 11 2020, doi: 10.3390/s20226442.

[255] F. Bu and M. S. Gharajeh, "Intelligent and vision-based fire detection systems: A survey," *Image and Vision Computing,* vol. 91, p. 103803, 2019/11/01/ 2019, doi: https://doi.org/10.1016/j.imavis.2019.08.007.

[256] S. A. D. Spiller, and L. Ansalone, "Transfer learning analysis for wildfire segmentation using PRISMA hyperspectral imagery and convolutional neural networks," presented at the IEEE WHISPERS, Rome, Italy, 2022.

[257] D. Spiller, L. Ansalone, S. Amici, A. Piscini, and P. Mathieu, "Analysis and Detection of Wildfires by Using Prisma Hyperspectral Imagery," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. XLIII-B3-2021, pp. 215-222, 2021, doi: 10.5194/isprs-archives-XLIII-B3-2021-215-2021.

[258] C. Boshuizen, J. Mason, P. Klupar, and S. Spanhake, "Results from the planet labs flock constellation," 2014.

[259] K. Thangavel, D. Spiller, R. Sabatini, P. Marzocca, and M. Esposito, "Near Real-time Wildfire Management Using Distributed Satellite System," *IEEE Geoscience and Remote Sensing Letters,* pp. 1-1, 2022, doi: 10.1109/LGRS.2022.3229173.

[260] C. Ulivieri and L. Anselmo, "Multi-sun-synchronous (MSS) orbits for earth observation," *Astrodynamics 1991,* pp. 123-133, 1992.

[261] R. Bernstein, "Digital image processing of earth observation sensor data," *IBM Journal of research and development,* vol. 20, no. 1, pp. 40-57, 1976.

[262] R. Showstack, "Sentinel satellites initiate new era in earth observation," ed: Wiley Online Library, 2014.

[263] S. B. Shah, T. Grübler, L. Krempel, S. Ernst, F. Mauracher, and S. Contractor, "REAL-TIME WILDFIRE DETECTION FROM SPACE – A TRADE-OFF BETWEEN SENSOR QUALITY, PHYSICAL LIMITATIONS AND PAYLOAD SIZE," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.,* vol. XLII-2/W16, pp. 209-213, 2019, doi: 10.5194/isprs-archives-XLII-2-W16-209-2019.

[264] M. Esposito *et al.*, *Highly Integration of Hyperspectral, Thermal And Artificial Intelligence for The ESA Phisat-1 Mission*. 2019.

[265] M. Pastena, "Φsat-1: The first ESA Earth Observation Directorate cubesat mission," presented at the 4th Cubesat Industry Days, Netherlands, 2019.

[266] P. Xu *et al.*, "On-Board Real-Time Ship Detection in HISEA-1 SAR Images Based on CFAR and Lightweight Deep Learning," *Remote Sensing,* vol. 13, no. 10, p. 1995, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/10/1995.

[267] D. Viegas, "Fire behaviour and fire-line safety," *Ann. Medit. Burns Club,* vol. 6, no. 179-85, p. 1998, 1993.

[268] M. A. Finney, *FARSITE, Fire Area Simulator--model development and evaluation* (no. 4). US Department of Agriculture, Forest Service, Rocky Mountain Research Station, 1998.

[269] E. Pastor, L. Zárate, E. Planas, and J. Arnaldos, "Mathematical models and calculation systems for the study of wildland fire behaviour," *Progress in Energy and Combustion Science,* vol. 29, no. 2, pp. 139-153, 2003.

[270] A. Alkhatib, "Forest Fire Monitoring," 2018.

[271] F. Tedim *et al.*, "Defining Extreme Wildfire Events: Difficulties, Challenges, and Impacts," *Fire,* vol. 1, no. 1, p. 9, 2018. [Online]. Available: https://www.mdpi.com/2571-6255/1/1/9.

[272] J. H. Scott, "Introduction to wildfire behavior modeling," *National Interagency Fuel, Fire, & Vegetation Technology Transfer,* 2012.

[273] P. Yu, R. Xu, M. J. Abramson, S. Li, and Y. Guo, "Bushfires in Australia: a serious health emergency under climate change," *The Lancet Planetary Health,* vol. 4, no. 1, pp. e7-e8, 2020.

[274] A. M. Gill, "Biodiversity and bushfires: an Australia-wide perspective on plant-species changes after a fire event," *Australia's biodiversity–responses to fire: plants, birds and invertebrates. Environment Australia Biodiversity Technical Paper,* vol. 1, pp. 9-53, 1999.

[275] L. R, A. C, R. Guarini, E. Lopinto, C. L, and A. R. Pisani, *THE PRISMA HYPERSPECTRAL MISSION*. 2016.

[276] R. Tricot, "Venture capital investments in artificial intelligence," 2021, doi: doi:https://doi.org/10.1787/f97beae7-en.

[277] J. G. Canadell *et al.*, "Multi-decadal increase of forest burned area in Australia is linked to climate change," *Nature Communications,* vol. 12, no. 1, p. 6921, 2021/11/26 2021, doi: 10.1038/s41467-021-27225-4.

[278] P. T. Buergelt and R. Smith, "Chapter 6 - Wildfires: An Australian Perspective," in *Wildfire Hazards, Risks and Disasters*, J. F. Shroder and D. Paton Eds. Oxford: Elsevier, 2015, pp. 101-121.

[279] B. N. Tran, M. A. Tanase, L. T. Bennett, and C. Aponte, "High-severity wildfires in temperate Australian forests have increased in extent and aggregation in recent decades," *PLOS ONE,* vol. 15, no. 11, p. e0242484, 2020, doi: 10.1371/journal.pone.0242484.
[280] M. Haque, M. Azad, M. Y. Hossain, T. Ahmed, M. Uddin, and M. Hossain, "Wildfire in Australia during 2019-2020, Its Impact on Health, Biodiversity and Environment with Some Proposals for Risk Management: A Review," *Journal of Environmental Protection,* vol. 12, pp. 391-414, 01/01 2021, doi: 10.4236/jep.2021.126024.
[281] R. Guarini *et al.*, *Prisma Hyperspectral Mission Products*. 2018, pp. 179-182.
[282] D. Spiller, L. Ansalone, S. Amici, A. Piscini, and P. P. Mathieu, "ANALYSIS AND DETECTION OF WILDFIRES BY USING PRISMA HYPERSPECTRAL IMAGERY," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.,* vol. XLIII-B3-2021, pp. 215-222, 2021, doi: 10.5194/isprs-archives-XLIII-B3-2021-215-2021.
[283] D. Spiller and S. Amici, "PRISMA tutorial on wildfires classification by using one-dimensional Convolutional Neural Networks."
[284] I. Bangash. "NVIDIA Jetson Nano vs Google Coral vs Intel NCS. A Comparison." Medium. https://towardsdatascience.com/nvidia-jetson-nano-vs-google-coral-vs-intel-ncs-a-comparison-9f950ee88f0d (accessed.
[285] M. P. Del Rosso, A. Sebastianelli, D. Spiller, P. P. Mathieu, and S. L. Ullo, "On-Board Volcanic Eruption Detection through CNNs and Satellite Multispectral Imagery," *Remote Sensing,* vol. 13, no. 17, p. 3479, 2021.
[286] A. Kurniawan, "Administering NVIDIA Jetson Nano," pp. 21-47, 2021, doi: 10.1007/978-1-4842-6452-2_3.
[287] A. Kurniawan, "Introduction to NVIDIA Jetson Nano," pp. 1-6, 2021, doi: 10.1007/978-1-4842-6452-2_1.
[288] A. Kurniawan, "NVIDIA Jetson Nano Programming," pp. 49-62, 2021, doi: 10.1007/978-1-4842-6452-2_4.
[289] A. Süzen, B. Duman, and E. B. Şen, "Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep CNN," 2020, doi: 10.1109/HORA49412.2020.9152915.
[290] H. Cui and N. Dahnoun, "Real-Time Stereo Vision Implementation on Nvidia Jetson TX2," pp. 1-5, 2019, doi: 10.1109/MECO.2019.8760027.
[291] H. H. Nguyen, Đ. Trần, and J. Jeon, "Towards Real-Time Vehicle Detection on Edge Devices with Nvidia Jetson TX2," pp. 1-4, 2020, doi: 10.1109/ICCE-Asia49877.2020.9277463.
[292] S. Amici and A. Piscini, "Exploring PRISMA Scene for Fire Detection: Case Study of 2019 Bushfires in Ben Halls Gap National Park, NSW, Australia," *Remote Sensing,* vol. 13, p. 1410, 2021, doi: 10.3390/rs13081410.
[293] K. Ranasinghe *et al.*, "Advances in Integrated System Health Management for mission-essential and safety-critical aerospace applications," (in English), *Progress in Aerospace Sciences,* Review vol. 128, 2022, Art no. 100758, doi: 10.1016/j.paerosci.2021.100758.
[294] L. Martínez *et al.*, *Architectures and Synchronization Techniques for Distributed Satellite Systems: A Survey*. 2022.
[295] V. W. Chan, "Optical satellite networks," *Journal of Lightwave Technology,* vol. 21, no. 11, p. 2811, 2003.
[296] H. Hemmati and D. Caplan, "Optical satellite communications," *Optical Fiber Telecommunications,* pp. 121-162, 2013.

[297] A. Chaudhry and H. Yanikomeroglu, *Free Space Optics for Next-Generation Satellite Networks*. 2020.

[298] M. Toyoshima, W. Leeb, H. Kunimori, and T. Takano. "Optical communications work best over relatively short distances in space." https://www.spie.org/news/0088-optical-communications-work-best-over-relatively-short-distances-in-space?SSO=1 (accessed 02 December 2022.

[299] K. Thangavel *et al.*, "Autonomous Satellite Wildfire Detection Using Hyperspectral Imagery and Neural Networks: A Case Study on Australian Wildfire," *Remote Sensing*, vol. 15, no. 3, doi: 10.3390/rs15030720.

[300] L. Shikwambana, M. Kganyago, and S. Xulu, "Analysis of wildfires and associated emissions during the recent strong ENSO phases in Southern Africa using multi-source remotely-derived products," *Geocarto International,* pp. 1-17, 2022, doi: 10.1080/10106049.2022.2113449.

[301] M. Salis *et al.*, "Assessing exposure of human and ecological values to wildfire in Sardinia, Italy," *International Journal of Wildland Fire,* vol. 22, no. 4, pp. 549-565, 2013, doi: https://doi.org/10.1071/WF11060.

[302] A. A. Ager, H. K. Preisler, B. Arca, D. Spano, and M. Salis, "Wildfire risk estimation in the Mediterranean area," *Environmetrics,* https://doi.org/10.1002/env.2269 vol. 25, no. 6, pp. 384-396, 2014/09/01 2014, doi: https://doi.org/10.1002/env.2269.

[303] M. Chintoan-Uta and J. R. Silva, "Global maritime domain awareness: a sustainable development perspective," *WMU Journal of Maritime Affairs,* vol. 16, pp. 37-52, 2017.

[304] B. J. Tetreault, "Use of the Automatic Identification System (AIS) for maritime domain awareness (MDA)," in *Proceedings of Oceans 2005 Mts/Ieee*, 2005: Ieee, pp. 1590-1594.

[305] N. Z. C. RAHMAN, "Maritime domain awareness in Australia and New Zealand," in *Maritime Security*: Routledge, 2009, pp. 228-249.

[306] A. Lauro and C. R. Corrêa, "Futures for the Maritime Domain: Signs and Trends That Shape Scenarios," in *Power and the Maritime Domain*: Routledge, pp. 286-301.

[307] S. Andrews, "Four Oceans: An integrated approach to Australia's maritime domain," UNSW Sydney, 2021.

[308] "Australian Government Civil Maritime Security Strategy," Australian Government, 2022. Accessed: October 2022. [Online]. Available: www.homeaffairs.gov.au/about-us/our-portfolios/national-security/civil-maritime-security

[309] G. L. Denton and J. R. Harris, "Maritime piracy, military capacity, and institutions in the Gulf of Guinea," *Terrorism and political violence,* vol. 34, no. 1, pp. 1-27, 2022.

[310] S. R. Greenway and C. J. Sipes, "Maritime domain awareness in the South China Sea: an operational picture design," Naval Postgraduate School Monterey United States, 2018.

[311] C. Bouchard, "Small Island Developing States and Maritime Security," in *Routledge Handbook of Maritime Security*: Routledge, 2022, pp. 250-264.

[312] G. Bigatti and J. Signorelli, "Marine invertebrate biodiversity from the Argentine Sea, South Western Atlantic," *ZooKeys,* vol. 791, 2018, doi: 10.3897/zookeys.791.22587.

[313] A. Jugović and D. Schiozzi, "Comparative analysis of concessions on maritime domain in ports of regional significance in Croatia and Italy," *Pomorstvo,* vol. 27, no. 2, pp. 299-312, 2013.

[314] L. Cordner, "Rethinking maritime security in the Indian Ocean Region," *Journal of the Indian Ocean Region,* vol. 6, no. 1, pp. 67-85, 2010.

[315] "Fishy networks: Uncovering the companies and individuals behind illegal fishing globally," Financial Transparency Coalition 2022. Accessed: October 2022. [Online]. Available: https://financialtransparency.org/reports/fishy-networks-uncovering-companies-individuals-behind-illegal-fishing-globally/
[316] J. Behrens and B. Lal, "Exploring Trends in the Global Small Satellite Ecosystem," *New Space,* vol. 7, pp. 126-136, 2019, doi: 10.1089/space.2018.0017.
[317] D. Wang, B. Wu, and E. K. Poh, *Satellite Formation Flying Relative Dynamics, Formation Design, Fuel Optimal Maneuvers and Formation Maintenance*. (in eng), 2017.
[318] W. M. Brown, "Synthetic Aperture Radar," *IEEE Transactions on Aerospace and Electronic Systems,* vol. AES-3, no. 2, pp. 217-229, 1967, doi: 10.1109/TAES.1967.5408745.
[319] J. C. Curlander and R. N. Mcdonough, "Synthetic Aperture Radar: Systems and Signal Processing," 1991.
[320] J. Louet and S. Bruzzi, "ENVISAT mission and system," in *IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS'99 (Cat. No. 99CH36293)*, 1999, vol. 3: IEEE, pp. 1680-1682.
[321] G. Krieger *et al.*, "TanDEM-X: A satellite formation for high-resolution SAR interferometry," *Geoscience and Remote Sensing, IEEE Transactions on,* vol. 45, pp. 3317-3341, 12/01 2007, doi: 10.1109/TGRS.2007.900693.
[322] A. Budillon, V. Pascazio, and G. Schirinzi, "Multichannel along-track interferometric SAR systems: Moving targets detection and velocity estimation," *International Journal of Navigation and Observation,* vol. 2008, 2008.
[323] K. Gosh, "Multi-baseline polinsar inversion and simulation of interferometric wavenumber for forest height retrieval using spaceborne SAR data," University of Twente, 2018.
[324] P. A. Servidia and M. España, "On Autonomous Reconfiguration of SAR Satellite Formation Flight With Continuous Control," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 57, no. 6, pp. 3861-3873, 2021.
[325] S. D'Amico, "Autonomous formation flying in low earth orbit," TU Delft, 2010.
[326] O. Montenbruck, T. Ebinuma, E. G. Lightsey, and S. Leung, "A real-time kinematic GPS sensor for spacecraft relative navigation," *Aerospace Science and Technology,* vol. 6, no. 6, pp. 435-449, 2002.
[327] A. Hauschild, J. Tegedor, O. Montenbruck, H. Visser, and M. Markgraf, "Precise onboard orbit determination for LEO satellites with real-time orbit and clock corrections," in *Proceedings of the 29th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2016)*, 2016, pp. 3715-3723.
[328] T. Burroni and P. Servidia, "Control Orbital Autónomo Restringido de Bajos Empujes y Filtrado de Elementos Orbitales," in *2022 IEEE Biennial Congress of Argentina (ARGENCON)*, 2022: IEEE, pp. 1-8.
[329] M. Davidson, "SAOCOM-CS Mission and ESA Airborne Campaign Data," 2015.
[330] N. Gebert, B. C. Dominguez, M. W. Davidson, M. D. Martin, and P. Silvestrin, "SAOCOM-CS-A passive companion to SAOCOM for single-pass L-band SAR interferometry," in *EUSAR 2014; 10th European Conference on Synthetic Aperture Radar*, 2014: VDE, pp. 1-4.
[331] MorpheusSpace. https://morpheus-space.com/products/nanofeep/.morpheus-apace.com/products/nanofeep (accessed November, 2022).

[332] ENPULSION. www.enpulsion.com (accessed November, 2022).

[333] L. M. Marrero *et al.*, "Architectures and Synchronization Techniques for Distributed Satellite Systems: A Survey," *IEEE access,* vol. 10, pp. 45375-45409, 2022.

[334] S. H. Cerruti, M. España, and P. Servidia, "Software Architecture Design of Distributed Satellite Systems Test Bed," in *2022 IEEE Biennial Congress of Argentina (ARGENCON)*, 2022: IEEE, pp. 1-8.