Doctoral Dissertations and Master's Theses

Fall 12-15-2022

Health Management and Adaptive Control of Distributed Spacecraft Systems

Tatiana Alejandra Gutierrez Martinez
Embry-Riddle Aeronautical University, gutiert6@my.erau.edu

Follow this and additional works at: https://commons.erau.edu/edt

Part of the Navigation, Guidance, Control and Dynamics Commons, and the Space Vehicles Commons

Scholarly Commons Citation

Gutierrez Martinez, Tatiana Alejandra, "Health Management and Adaptive Control of Distributed Spacecraft Systems" (2022). *Doctoral Dissertations and Master's Theses*. 707.
https://commons.erau.edu/edt/707

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Doctoral Dissertations and Master's Theses by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

HEALTH MANAGEMENT AND ADAPTIVE CONTROL OF DISTRIBUTED SPACECRAFT SYSTEMS

By

Tatiana Alejandra Gutierrez Martinez

A Thesis Submitted to the Faculty of Embry-Riddle Aeronautical University
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Aerospace Engineering

December 2022
Embry-Riddle Aeronautical University
Daytona Beach, Florida

HEALTH MANAGEMENT AND ADAPTIVE CONTROL OF DISTRIBUTED SPACECRAFT SYSTEMS

By

Tatiana Alejandra Gutierrez Martinez

This Thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Hever Moncayo, Department of Aerospace Engineering, and has been approved by the members of the Thesis Committee. It was submitted to the Office of the Senior Vice President for Academic Affairs and Provost, and was accepted in the partial fulfillment of the requirements for the Degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE

Chair, Dr. Hever Moncayo

Member, Dr. Richard Prazenica

Member, Dr. Kadriye Merve Dogan

Graduate Program Coordinator,
Dr. Hever Moncayo

Date

Dean of the College of Engineering,
Dr. James W. Gregory

Date

Associate Provost of Academic Support,
Dr. Christopher Grant

Date

ACKNOWLEDGMENTS

This section's goal is to acknowledge and thank the people that have helped me in significant ways to accomplish this endeavor:

I would like to express my special thanks to my advisor, Dr. Hever Moncayo, for his invaluable guidance and support throughout my studies and for helping me shape and navigate the challenges of this research. I would also like to express my sincere gratitude to Dr. Dogan and Dr. Prazenica for their valuable feedback to improve this research and for helping me achieve a higher level of excellence. Likewise, I would like to gratefully acknowledge the Maritime and Multi-Agent Autonomy Group at NASA Jet Propulsion Laboratory for their technical input and constructive feedback. As well, special thanks to my colleagues at the Advanced Dynamics and Control Lab for always being willing to answer my questions and for their support.

To my parents and my sister, I am forever grateful for your love, unconditional support, and effort in educating me. Thank you for providing all the experiences that have given me everything I know and stand for today. To David, thank you for believing in me and for your unconditional love and support every step of the way. To my friends all around the World, thank you for supporting me and being part of my life.

Most of all, I would like to acknowledge the journey; in which achieving results is incredible but the true fulfilling reward is everything I have learned during the entire process. Special thanks to God for the many opportunities and for giving me the strength and perseverance to excel in this journey.

ABSTRACT

As the development of challenging missions like on-orbit construction and collaborative inspection that involve multi-spacecraft systems increases, the requirements needed to improve post-failure safety to maintain the mission performance also increases, especially when operating under uncertain conditions. In particular, space missions that involve Distributed Spacecraft Systems (e.g, inspection, repairing, assembling, or deployment of space assets) are susceptible to failures and threats that are detrimental to the overall mission performance. This research applies a distributed Health Management System that uses a bio-inspired mechanism based on the Artificial Immune System coupled with a Support Vector Machine to obtain an optimized health monitoring system capable of detecting nominal and off-nominal system conditions. A simulation environment is developed for a fleet of spacecraft performing a low-Earth orbit inspection within close proximity of a target space asset, where the spacecraft observers follow stable relative orbits with respect to the target asset, allowing dynamics to be expressed using the Clohessy-Wiltshire-Hill equations. Additionally, based on desired points of inspection, the observers have specific attitude requirements that are achieved using Reaction Wheels as the control moment device. An adaptive control based on Deep Reinforcement Learning using an Actor-Critic-Adverse architecture is implemented to achieve high levels of mission protection, especially under disturbances that might lead to performance degradation. Numerical simulations to evaluate the capabilities of the health management architecture when the spacecraft network is subjected to failures are performed. A comparison of different attitude controllers such as Nonlinear Dynamic Inversion and Pole Placement against Deep Reinforcement Learning based controller is presented. The Dynamic Inversion controller showed better tracking performance but large control effort, while the Deep Reinforcement controller showed satisfactory tracking performance with minimal control effort. Numerical simulations successfully demonstrated the potential of both the bio-inspired Health Monitoring System architecture and the controller, to detect and identify failures and overcome bounded disturbances, respectively.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ACS	Attitude Control Subsystem
$ADCL$	Advanced Dynamics and Control Laboratory
AI	Artificial Intelligence
AIS	Artificial Immune System
$AISO-SVM$	Artificial Immune System Optimized Support Vector Machine
ARE	Algebraic Ricatti Equation
CWH	Clohessy-Wiltshire-Hill
DL	Deep Learning
DRL	Deep Reinforcement Learning
DSS	Distributed Spacecraft Systems
ECI	Earth-centered Inertial
FDI	Fault Detection and Identification
$GANN$	Generative Adversarial Neural Networks
H_∞	H-infinity
HJB	Hamilton-Jacobi-Bellman
HJI	Hamilton-Jacobi-Isaacs
HM	Health Monitoring
HMS	Health Monitoring System
LEO	low-Earth-orbit

LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
$LVLH$	Local Vertical Local Horizontal
$NLDI$	Nonlinear Dynamics Inversion
NN	Neural Network
POI	Point of Interest
PRO	Passive Relative Orbit
RW	Reaction Wheel
SC	Spacecraft
SVM	Support Vector Machine

1 Introduction

1.1 Motivation

Space exploration has traditionally been performed by single monolithic spacecraft carrying multiple science instruments on board. However, only a portion of instruments can be operated simultaneously due to system constraints such as power requirements, viewing angle, storage capacity, and mechanical limitations. This condition and the increase in more challenging space missions (e.g, on-orbit construction and collaborative inspection), have accelerated the interest in implementing Distributed Spacecraft Systems (DSS). In fact, recent scientific advances in hardware miniaturization have provided new ideas for deploying large fleets of small satellites [2]. Such systems represent constellations of small satellites, which envision improvements in mission efficiency and performance, as the mission objectives do not rely on a single spacecraft. As a result, space missions involving DSS hold promise for better data quality results, shorter mission duration, and improved robustness to failures [3]. For instance, as stated by Foust et al. [6], recent theoretical developments have revealed that a team of collaborating spacecraft can augment the robustness and versatility of the mission at a reduced cost and time, which make these architectures attractive for new emerging applications.

Space missions that involve networked multi-spacecraft systems pose significant challenges to flight safety. This is especially the case due to the complexity of space missions and the harsh operating environments of space, which leaves these systems susceptible to threats and failures that affect flight safety and mission performance. Furthermore, there exist infinite possibilities of failure, during the different phases of a mission, and they can be caused due to external or internal factors. Moreover, according to the report presented in [7], small satellite failures are categorized as mission failures and partial mission failures, where the first ones happen when the spacecraft fails to operate or complete the mission objectives and the second one happens when the spacecraft experiences component failures.

The Attitude Control System (ACS) is considered a critical subsystem in spacecraft prone to failures caused by malfunctions in its components, actuators, and sensors, which could result in high energy consumption, loss of control, and catastrophic loss of spacecraft [8]. For example, the Hubble Space Telescope has experienced several failures in its reaction wheels; this represents a critical malfunction to the mission where the main goal is to maintain pointing accuracy.

In an effort to improve flight safety, an active research area is the development of Fault Detection and Identification (FDI) methodologies with the goal of providing a mechanism of status update, where systems can sense abnormal behavior, detect, and identify potential threats. The implementation of these mechanisms could lead to increased autonomy trust, reliability, and robustness of mechanical systems. Moreover, the development of robust and intelligent FDI tools capable of detecting and identifying sensors and system faults in real-time can assist in decision-making processes, mitigate performance degradation and increase the chances of mission success. Consequently, in the case of DSS, access to Health Monitoring Systems (HMS) information would provide knowledge of the mission status and allow agents to make decisions towards the benefit of completing the mission, by executing adjustments in the controller, trajectory or formation of the mission, among others.

These efforts are of significant importance in the aerospace industry as intelligent systems serve as a means to augment human expertise and assist in endeavors that save cost and time, which in space are considered critical assets. The rapidly growing application of data-driven algorithms using machine learning to describe and predict complex systems has provided significant contributions to the development of FDI systems, as these require the processing of enormous amounts of data. A promising data-driven approach in FDI is the Artificial Immune System (AIS) paradigm for fault detection in aerospace systems [9–11] and more recently in spacecraft systems [12]. The AIS paradigm employs bio-inspired algorithms and the principle of *self/non-self* discrimination to distinguish between different classes of data. The development of a HMS based on the AIS paradigm and machine learning classifier methods such as Support Vector Machine (SVM) to improve detection capabilities by optimizing data classification has been made at the Advanced Dynamics and Control Laboratory (ADCL) [13]. In a similar manner, this approach has been proposed to assist with hyperparameter optimization for a SVM classifer in [14].

Machine learning algorithms have also been explored for system control, by characterizing a control law that effectively regulates a complex system. Based on optimization theory, challenging problems for minimizing or maximizing a given quantity subject to constraints are solved. Control algorithms based on adaptive neural networks, genetic algorithms, evolutionary algorithms, and deep reinforcement learning are some of the different machine learning techniques explored for controlling complex, nonlinear systems [15].

Deep Reinforcement Learning (DRL) based algorithms such as Actor–Critic-Adverse structures have been studied and developed in machine learning to learn the optimal policy online [16]. Furthermore, in the presence of disturbances and uncertainties, Actor-Critic-Adverse structures that include Generative Adversarial Neural Networks (GANN) hold promising results based on concepts of game theory and H_∞ control, where the controller is a minimizing player and the disturbance a maximizing one. Thus, the algorithm provides the optimal control policy considering the cost of performance and the cost of actuation while the worst disturbance that maximizes the value function is chosen. DRL control structures are of special interest in this research considering the approximation of the value function (critic) involved in optimal control which finds the minimum control effort when minimizing for different variables in the system. Output information from the HMS could be included in the optimization process and this type of controller can be useful for resource allocation purposes.

The application of an Artificial Intelligence (AI) based HMS tool is applied in this research and the detection capabilities in terms of false alarms and detection rates are assessed for a fleet of spacecraft in low -Earth orbit that is performing on-orbit inspection of a target space asset, under failure conditions. Numerical simulations to evaluate the capabilities of the HMS and the performance of the DRL controller to overcome bounded disturbances are performed.

1.2 Problem Statement

FDI strategies for aerospace systems are needed to protect mission objectives and performance. It is desired to provide FDI capabilities to DSS missions especially due to the high susceptibility to failures in spacecraft subsystems and the harsh operating environment of space. This research effort proposes a distributed health management system for fault detection and identification on DSS in low Earth orbit inspection missions. Access to this information would provide knowledge of the mission status and allow agents to make decisions towards the benefit of completing the mission, by executing adjustments in the controller, trajectory (orbits) or formation of the mission, among others. The description of the design reference mission and the dynamical system is described in the following sections.

1.3 Thesis Outline

Chapter 1 presents a generalized background of DSS, trajectory optimization, data-driven FDI approaches and DRL based control. This chapter includes concepts from optimal control, robust control, and game theory. It also presents the neural network-based adaptive laws proposed in [17] for an Actor-Critic-Adverse control structure.

Chapter 2 presents the HM Framework, including the concepts of *self* and *non-self* representation, a description of the algorithms used to generate detectors: Variable Detector and Artificial Immune System Optimized Support Vector Machine (AISO-SVM), and a definition of the performance metrics used to assess the classification and detection capabilities of the HMS. The distributed HM architecture applied to a network of spacecrafts is also presented.

Chapter 3 provides an overview of the spacecraft dynamics and equations of motion for attitude determination and control. These equations are used in the design of an NLDI controller, while the spacecraft linear model is used in the derivation of a DRL based controller and Pole Placement controller. The equations of motion using Reaction Wheels (RW) as control moment devices are also provided.

Chapter 4 presents the simulation environment main components and describes the reference mission of multi-spacecraft following stable relative orbits governed by the Clohessy-Wiltshire-Hill (CWH) equations. A review of the spacecraft dynamics using the CWH equations, which provides a linear approximation of the relative dynamics between the target space asset and each spacecraft in the fleet is presented. Furthermore, a detailed description of the Reaction Wheel model used is developed and the description of three types of failures cases in the attitude control system that uses reaction wheels is presented.

Chapter 5 summarizes the results and numerical simulations for the tested failure scenarios. A description of the features used for training the HMS is shown. Additionally, an analysis of Local and Global HMS detection capabilities in terms of detection rates and false alarms is provided. A comparison between the DRL controller performance is made with respect to a NLDI controller and Pole Placement controller in the presence of failures and disturbances.

Chapter 6 delves into the main conclusions of this thesis and discusses methodologies to continue the research for future work.

1.4 Literature Review

In this section, a literature review of DSS using small satellites is presented along with the challenge of trajectory optimization. An overview of optimal control and deep learning concepts is also presented.

1.4.1 Distributed Spacecraft Systems

The research in DSS has gradually broadened in recent years and its benefits are already being explored in space by different multi-spacecraft missions.

In Earth orbit, several multi-spacecraft missions are in operation (e.g, the Afternoon Constellation with satellites GCOM-W1, OCO-2M Aqua, Aura, CALIPSO and CloudSat [1, 18] and GRACE-FO [19]) or planned for launch (e.g. CloudCT [20]) to provide simultaneous observations for Earth science applications. As an illustration, the Afternoon Constellation as of 2022, is presented in Figure 1.1, where although each satellite is operated separately, their operation is carefully coordinated to increase observation efficiency.

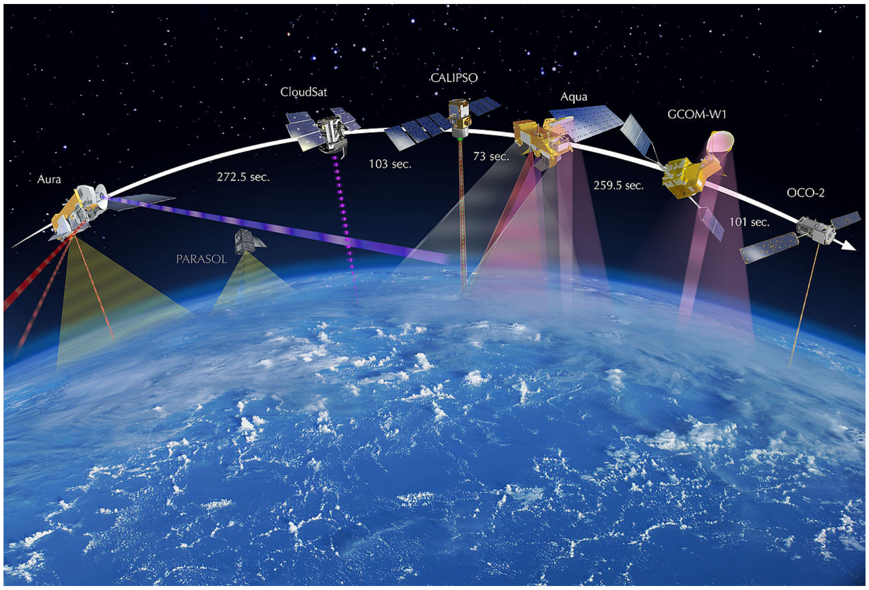


Figure 1.1 A-Train Constellation., from NASA [1]

In deep space, a reference mission is the the MarCO mission [21] which has successfully demonstrated the technical feasibility of DSS by being the first deep-space mission using multi-spacecraft vehicles. Future deep-space missions involving DSS include the SunRISE mission [22] which will consist of six small spacecraft in a geosynchronous Earth orbit to study solar activity, the Mars Swarm Array Transmitter [23] which consists of a high data rate swarm array to support future communications demands in space, and the Cooperative Autonomous Distributed Robotic Explorers (CADRE) [24]), which consists of multiple ground robots programmed to work cooperatively as an autonomous team to collect data from different areas of the Moon.

The use of smaller spacecraft has increased recently as endless applications have been identified, and more benefits from technological advances are developed. Benefits such as greater data processing, transmission capacity, optimized optical communications, improved autonomy, and inter-spacecraft navigation are some of the characteristics of smaller spacecraft [2].

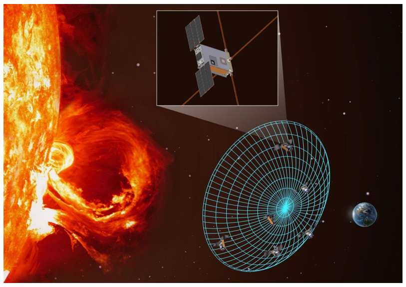


Figure 1.2 SunRISE Mission - 2024

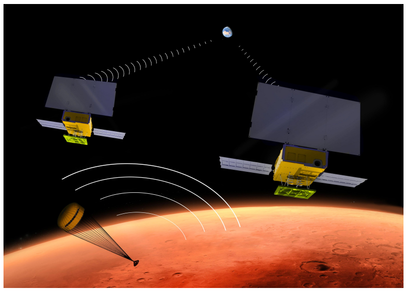


Figure 1.3 MarCo Mission - 2018

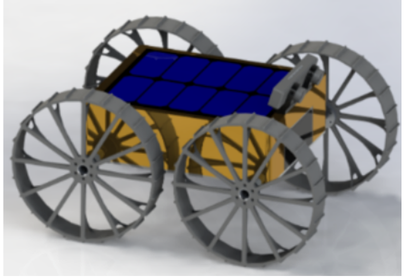


Figure 1.4 CADRE robot concept

The agreed size classification scheme for small spacecraft buses according to NASA [1] is shown in Table 1.1.

Table 1.1 Small Spacecraft Buses Categories from NASA [1]

Minisatellite	100 – 180 kg
Microsatellite	10 – 100 kg
Nanosatellite	1 – 10 kg
Picosatellite	100 g - 1 kg
Femtosatellite	Less than 100 g

From this classification, the most common small satellite types are the nanosatellite and microsatellites, commonly called CubeSats. A CubeSat is based on a 10 cm square cube or 1U form factor and weights less than 30 kg [2]. Regardless of their small size, nanosatellites usually have transmitters, receivers, antennae, and solar cells, and carry dozens of microprocessors. Furthermore, picosatellites usually have propulsion systems and attitude control systems in the form of miniature gyroscopes or magnetorques whose magnetic fields produce torque. On the other hand, femtosatellites do not have any means of attitude control or propulsion, but do include transmitters to rely data to the ground or to a larger spacecraft [7]. Figure 1.5 shows a representation of the common form factor of CubeSats based on unit 1U.

The initial objective of developing CubeSats was for educational purposes in universities as a means to provide hands-on experience for students. This idea has evolved, and as of today, the CubeSat spacecraft is a diverse platform that offers valuable processing power,

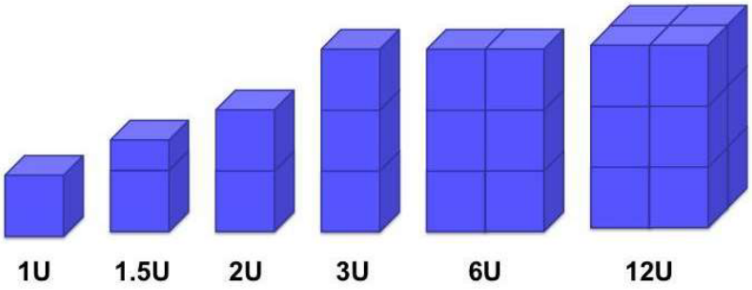


Figure 1.5 CubeSats standard size and form factor, from Yost et al. [2]

on-orbit capability, and science/data collection, which make Cubesats interesting systems for multiple space applications. As a reference of the evolution of CubeSats missions and applications, Figure 1.6 shows insights about small spacecraft mission trends from 2010 to 2020.

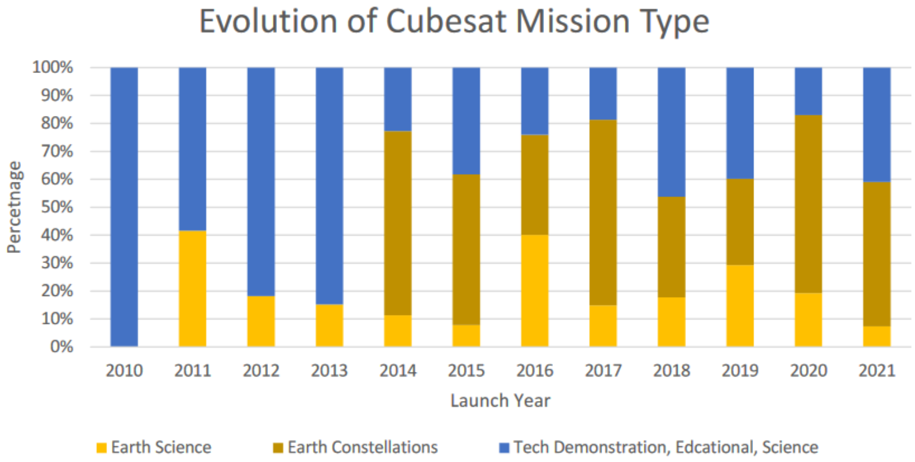


Figure 1.6 Evolution of CubeSat Missions, from Yost et al. [2]

Small satellite technology holds promise to improve mission efficiency and the broad spectrum of applications using these systems has motivated interest in designing multiple spacecraft architectures for space exploration. For instance, recent research by Rossi et al. [3], suggests that a swarm of small spacecraft could be more efficient for small body inspection compared to traditional monolithic architectures, resulting in better data quality, shorter mission duration, and additional robustness to failures of individual vehicles [3].

Nonetheless, implementing small multi-spacecraft missions involves considering multiple challenges, considering small satellites have multiple possible points of failure and the harsh environments of space. These vehicles typically have multiple microprocessors and related electronics used to operate the different subsystems onboard, such as control power management, solar cell operation, attitude control, telemetry, propulsion systems, star trackers, sun sensors, and earth sensors, among others. A malfunction in any one of these subsystems can cause a small satellite to fail or partially fail, thus affecting the mission objectives and performance [7]. From this motivation, it is desired to provide FDI capabilities to multiple spacecraft missions due to the high susceptibility to failures in spacecraft subsystems and the need to increase chances of mission success.

1.4.2 Trajectory Optimization

The challenge of orbit design for multiple spacecraft has been widely addressed. For example, flower constellations designs have been proposed for low-Earth orbit missions, which can meet particular mission requirements and are of interest for telecommunications coverage applications [25]. However, the challenge of orbit design remains briefly addressed in the literature for deep space missions.

Trajectory design for deep space applications is challenging, as the spacecraft network not only have to coordinate their motion, but also must consider inter-spacecraft communication constraints and onboard memory limitations. Thus, previous research have emphasized the design of optimized orbit algorithms that can account for instrument requirements and communication constraints. Nonetheless, it is necessary to also account for failures and disturbances while designing trajectory optimization algorithms. For instance, information from the modified flight envelop of the mission provided from the HMS could support trajectory optimization processes, considering failures or any abnormal conditions in the network.

A recent study by Rossi et al. [3] made significant contributions towards the development of orbit optimization considering communication constraints and irregular gravity fields from small bodies such as asteroids. Based on the mission concept of asteroid inspection by multiple spacecraft shown in Figure 1.7, the authors propose an optimization framework that simultaneously optimizes the observations captured by the spacecraft and the data flows for a given set of orbits, maximizing the amount of scientific data relayed to the carrier spacecraft.

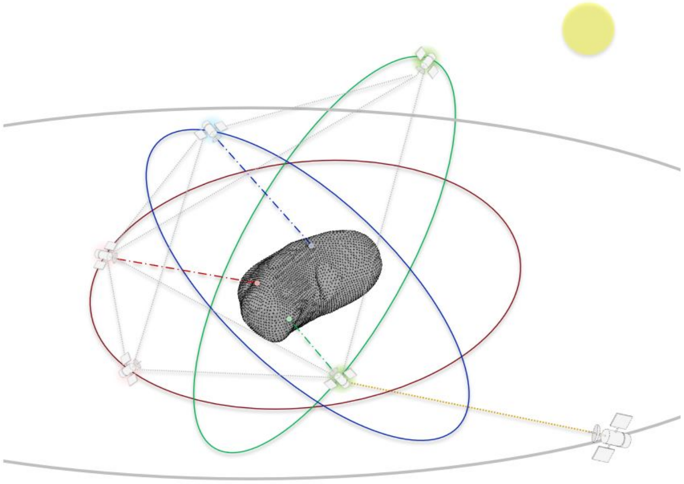


Figure 1.7 Representation of a mission configuration with 6 spacecraft orbiting around the asteroid 433 Eros. From Rossi et al. [3]

The importance of gathering data from a Health Management system and using it for decision-making processes is highlighted in this research, which could improve the robustness and accuracy of trajectory optimization algorithms.

1.4.3 Fault Detection and Identification Data-driven Approaches

Fault tolerant control technology plays a fundamental role in flight safety and control systems. The prompt detection of faults can avoid system damage and fatal accidents. Most early studies as well as current work on FDI methodologies focus on data-driven algorithms, as they have provided a great contribution to this area of research. Gao et al. [26] provide a comprehensive survey of data-driven fault detection strategies applied to autonomous systems and processes [26, 27]. In the literature, neural network classifiers and Kalman filter sensor fusion techniques have been applied to fault detection, diagnosis, and accommodation architectures within aerospace systems [28, 29]. These methods implement artificial intelligence techniques to available historic flight data of the system to categorize the information as nominal and abnormal behaviors.

A promising data-driven method is the Artificial Immune System (AIS) paradigm which consists of a bio-inspired algorithm similar to the immune system of living beings, to distinguish between different classes of data. The AIS has inspired algorithms in a wide range of applications such as anomaly detection, data mining, computer security, adaptive control, and pattern recognition [10].

Additionally, AIS augmentation has been proposed to assist with classifiers augmentation and hyperparameter optimization for a SVM classifer [29]. Research efforts at the Advanced Dynamics and Control Laboratory have extended this concept by implementing the Clonal Selection Algorithm, a bio-inspired optimization process to improve SVM training within a spacecraft fault detection system [30]. The SVM can also be combined with the Negative Selection and Variable Detection algorithms to improve data clustering performance within fault detection methodologies.

1.4.4 Review of Optimal Control and Deep Reinforcement Learning

Given a controllable nonlinear time-varying dynamical system as:

$$\dot{\bar{x}}(t) = f(\bar{x}, \bar{u}, t)$$

There exist many choices for stabilizing or performing tracking control by implementing a full-state feedback control law as shown in Equation 1.1:

$$\bar{u}(t) = -\mathbf{K}\bar{x}(t) \tag{1.1}$$

where K is a control gain matrix.

It is possible to place the eigenvalues of the closed-loop system as far as desired in the left half of the complex plane and make the system arbitrarily stable if the system is controllable. However, it is not guaranteed that overly stable eigenvalues will provide the minimum control effort. When this happens, expensive control expenditure resulting in actuation signals that exceed maximum allowable values is expected. Additionally, overly stable eigenvalues may also cause the control system to overreact to noise and disturbances affecting performance and robustness of system [15].

Several methods reported in the literature address the balance between the stability and aggressiveness of control. Optimal control is a well-defined framework for studying optimal decision-making processes and is usually implemented as a Linear Quadratic Regulator (LQR) or Linear Quadratic Gaussian (LQG), where the main concept within Optimal Control is to minimize a generalized cost function. Also, throughout time, other controllers have been proposed based on learning algorithms such as Deep-Reinforcement Learning control with promising applications to complex dynamical systems. These methodologies will be analyzed in more detail in the following subsections.

1.4.5 Optimal Control

The theory of optimal control has a long history as one of the fundamental approaches of modern control systems design. Specifically, the optimal control problem is to find the optimal control input $\bar{u}(t)$ on the time interval $[t_0, T]$ that drives a linearized dynamic system along a trajectory $\bar{x}(t)$ such that a generalized cost function such as Equation 1.2 is minimized:

$$J(t_0) = \phi(\bar{x}(T), T) + \int_{t_0}^{T} L(\bar{x}(t), \bar{u}(t), t)dt \tag{1.2}$$

where the final weighting function $\phi(\bar{x}(T), T)$ depends on the final state and final time, and the weighting function $L(\bar{x}(t), \bar{u}(t), t)$ depends on the state and input at intermediate times in $[t_0, T]$. The cost function is selected to make the system exhibit a desired type of performance and there exist various possibilities to select $J(t_0)$ [31].

In order to solve the optimal control (i.e., to obtain the solution to the cost function with minimum control effort), Lagrange multipliers are used to relate the constraints of the systems and the cost function. For this, the Hamiltonian is used:

$$\mathcal{H}(\bar{x}, \bar{u}, t) = L(\bar{x}(t), \bar{u}(t), t) + \lambda^T f(\bar{x}, \bar{u}, t) \tag{1.3}$$

where λ is a Lagrange multiplier. The Hamilton-Jacobi-Bellman (HJB) is shown in Equation 1.4:

$$0 = \frac{\partial J^*}{\partial t} + \mathcal{H}^* \tag{1.4}$$

where $\mathcal{H}^*$ is the minimized Hamiltonian with respect to the control input:

$$\mathcal{H}^* = \min_{\bar{u}(t)} \mathcal{H} \tag{1.5}$$

Numerous results are available for obtaining the solution of the discrete-time HJB equation; however, in the continuous-time (CT) case, the solution is more challenging due to the nature of the nonlinear differential equations. For this reason, the HJB equation is not easy to solve analytically [32]. This has led to significant research efforts for developing algorithms to approximate the solution of the HJB equation as will be discussed in Subsection 1.4.7.

1.4.6 Linear Quadratic Regulator from HJB

It is desired to find the optimal control to drive the state from a given $\bar{x}(t_0)$ to a desired state $\bar{x}(t_f)$ so that $J(t_0)$ is minimized.

Consider the linear dynamical system in Equation 1.6:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u} \tag{1.6}$$

with quadratic cost index or cost function:

$$J(t_0) = \frac{1}{2}\bar{x}^T(T)S(T)\bar{x}(T) + \frac{1}{2}\int_{t_0}^{T} (\bar{x}^T Q\bar{x} + \bar{u}^T R\bar{u})dt \tag{1.7}$$

In this cost function, the first term is a function of the final state which is desired to make small, for example energy $[\bar{x}^T(T)S(T)\bar{x}(T)]/2$, where **S** is a given weighting matrix [31].

Equation 1.7 considers the cost of performance and the cost of actuation, where matrix **Q** is positive semi-definite that weights the cost of the states and matrix **R** is positive definite that weights the cost of actuation. These matrices are often diagonal, and the diagonal elements may be tuned to consider the importance of the control objectives.

Applying Euler-Lagrange Equations, the following Hamiltonian form can be defined:

$$\begin{aligned} \mathcal{H}(\bar{x}, \bar{u}, \lambda) &= L(\bar{x}, \bar{u}) + \lambda^T f(\bar{x}, \bar{u}, t) \\ \mathcal{H}(\bar{x}, \bar{u}, \lambda) &= \frac{1}{2}(\bar{x}^T Q \bar{x} + \bar{u}^T R \bar{u}) + \lambda^T f(\bar{x}, \bar{u}, t) \end{aligned} \tag{1.8}$$

where λ is a Lagrange multiplier.

Now, minimizing the Hamiltonian $\mathcal{H}^*$ with $f(\bar{x}, \bar{u}, t) = A\bar{x} + B\bar{u}$:

$$\begin{aligned} \mathcal{H}^* = \frac{\partial H}{\partial \bar{u}} = 0 \\ R\bar{u} + \lambda B = 0, \end{aligned} \tag{1.9}$$

Then, the optimized control is obtained as:

$$\bar{u}^* = -R^{-1} B^T \lambda \tag{1.10}$$

Proposing $\lambda = Sx$, where S is a positive definite matrix, provides the solution to the Riccati Equation:

$$-\dot{S} = SA + A^T S - SBR^{-1} B^T S + Q \tag{1.11}$$

Therefore, the optimal control has a time-varying gain:

$$K_r(t) = R^{-1} B^T S(t)$$

which regulates the state of the system to:

$$\lim_{t \to \infty} x(t) = 0$$

where $S(t)$ is obtained by solving the Riccati Equation.

When $\dot{S} = 0$, Equation 1.11 is known as the Algebraic Ricatti Equation (ARE). Then, the optimal control law converges to:

$$U_*(t) = -K_\infty x(t)$$

$$K_\infty = R^{-1} B^T S_\infty$$

where S_∞ is the solution of the ARE.

1.4.7 Deep-Reinforcement Learning Control

Significant research efforts have been made for developing algorithms to approximate the solution of the HJB equation. These solutions suggest the use of Policy Iteration (PI) which is based on a two step iteration: policy evaluation and policy improvement. The algorithm starts by evaluating the cost of an initial control policy and this new cost is used to obtain a new improved control policy, with lower associated cost. This two step algorithm is repeated until the policy improvement no longer changes and convergence to the optimal controller is obtained.

For the optimal control problem of linear continuous-time systems when quadratic indices are used, the HJB equation becomes the Riccati Equation and research efforts have indicated that Neural Networks (NN) can be trained to become approximate solutions. In fact, the authors of [16] defined an algorithm based on actor-critic structures to solve the optimal control problem using PI algorithms. Here, the Hamiltonian-Jacobi-Isaac (HJI) equation can be interpreted as the value function (e.g., rewards function) to be approximated by a set of critic NN and a set of actor NN for approximation of the control policy [32]. Bhasin et al. [33] elaborated on these concepts to obtain adaptation of both actor and critic neural networks to approximately solve the continuous-time infinite horizon optimal control problem for uncertain nonlinear systems; however, solution methods are generally applied offline.

Moreover, in an effort to reduce supervision and training of machine learning algorithms, researchers study unsupervised learning often using generative modeling using generative adversarial NN. This kind of NN is based on a game, in the sense of game theory between two players or machine learning models using NNs [34]. Generative Adversarial Neural Networks (GANN) have been previously explored for obtaining control policies in the presence of disturbances or uncertainties in the dynamical system. More specifically, game theory and H_∞ control based on an optimization problem and a zero-sum game, where the controller is a minimizing player and the disturbance a maximizing one, provides an approach to solving the HJI equations. If the amplitude of the disturbances are mapped within the value function and are also approximated by a GANN (adversary), then an adaptive controller (actor) can be synthesized following a min-max solution to a H_∞ problem, also known as differential game [35].

Vamvoudakis and Lewis [17, 35] provide methods for online gaming by learning the optimal policy based on a 2-player zero-sum infinite horizon game online known as actor-critic-adverse structure. Within a H_∞ theory context, optimal control concepts can be borrowed to design a DRL system robust to uncertainties and disturbances [36].

Given the nonlinear system dynamics:

$$\dot{\bar{x}} = f(\bar{x}) + g(\bar{x})\bar{u}(\bar{x}) + k_d(\bar{x})w(\bar{x}) \tag{1.12}$$

where $\bar{x} \in X \subset \mathbf{R}^n$ is the state vector, $\bar{u} \in U \subset \mathbf{R}^m$ is the control input, $w \in \mathbf{W} \subset R^l$ is the disturbance input. In the H_∞ control problem, one can find a control input that satisfies the following constrained problem:

$$V(\bar{x},\bar{u},w)=\int_0^\infty q(\bar{x},\bar{u},w,t)dt=\int_0^\infty (r(\bar{x}(t),\bar{u}(t))+\omega(w(t)))dt \tag{1.13}$$

where $q(t)$ represents a reward function that is a function of $r(\bar{x},\bar{u})$, that includes the rewards for the state and the cost for the action; and the cost for the disturbance $\omega(w)$. One can use quadratic cost for the disturbance input as:

$$\omega(w)=-\gamma^2 w^T w \tag{1.14}$$

where γ is a robustness parameter. The reward function $r(\bar{x},\bar{u})$ could also be chosen to be quadratic as:

$$r(\bar{x},\bar{u})=\bar{x}^T Q\bar{x}+\bar{u}^T R\bar{u} \tag{1.15}$$

where $\mathbf{Q}>0$ and $\mathbf{R}>0$ are weighting matrices of the state $\bar{x}(t)$ and input $\bar{u}(t)$, respectively. For a tracking type of problem, the error dynamics $\bar{e}(t)$ can also be used in Equation 1.15 instead of the state $\bar{x}(t)$ without loss of generality.

The problem formulated in Equation 1.16 can be considered as a differential game where the optimal control output $\bar{u}$ that minimizes the function V is calculated while the worst disturbance w that maximizes V is chosen. The differential game problem can then be formulated as:

$$V^*(\bar{x},\bar{u},w)=\min_{\bar{u}}\max_{w}\int_0^\infty (r(\bar{x}(t),\bar{u}(t))-\gamma^2 w^T w)dt \tag{1.16}$$

The condition for the optimal value function is given by the well known Hamilton-Jacobi-Isaacs equation:

$$0 = \min_{\bar{u}} \max_{w} \left[r(\bar{x}, \bar{u}) - \gamma^2 w^T w + \frac{\partial V^*}{\partial x} (f(\bar{x}) + g(\bar{x})\bar{u} + k_d(x)w) \right] \tag{1.17}$$

From the HJI equation, the optimal control $\bar{u}$ and the worst disturbance w can be derived by solving:

$$\frac{\partial r(\bar{x}, \bar{u})}{\partial \bar{u}} + \frac{\partial V^*}{\partial \bar{x}} \frac{\partial [f(\bar{x}) + g(\bar{x})\bar{u} - k_d(\bar{x})w]}{\partial \bar{u}} = 0 \tag{1.18}$$

$$\frac{\partial \omega}{\partial w} + \frac{\partial V^*}{\partial \bar{x}} \frac{\partial [f(\bar{x}) + g(\bar{x})\bar{u} - k_d(\bar{x})w]}{\partial w} = 0 \tag{1.19}$$

Notice that in Equations 1.17, 1.18 and 1.19, $\frac{\partial V^*}{\partial x}$ represent the steepest ascent direction of the value function, which is then transformed by the partial derivatives of f into a direction in the control output and disturbance input space, respectively [36].

Given a solution, from Equations 1.18 and 1.19, the associated best control law and the worst disturbance are as follows:

$$\bar{u} = -\frac{1}{2} R^{-1} g^T(\bar{x}) \nabla \Lambda^* \tag{1.20}$$

$$w = \frac{1}{2\gamma^2} {k_d}^T(\bar{x}) \nabla \Lambda^* \tag{1.21}$$

where $\nabla \Lambda^* = \frac{\partial V^*}{\partial x}$ is the transposed gradient with $V^*(0) = 0$ and in convergence $H^* = 0$.

It is important to note that one of the main problems in optimal control is the calculation of the exact value of V^* as the optimal solution of the Hamiltonian equation. As such, a uniform approximation of the value function V^* and its gradient can be performed using

NNs as follows [17, 37, 38]:

$$V^*(x) = W_c^{*T}\Phi_c(x) + \epsilon_c(x) \tag{1.22}$$

$$\nabla\Lambda^*(x) = \nabla\Phi_c^T W_c^* + \nabla\epsilon_c(x) \tag{1.23}$$

where $W_c \in \mathbf{R}^N$ is the NN weights, with N the number of activation functions, and Φ_c a completely independent basis set or activation functions within the NN architecture. ϵ_c is the critic approximation error. Subscript c refers to the *critic* NN within the DRL architecture.

Furthermore, the feedback policy in the form of an *action* NN computes the optimal control as follows:

$$u = -\frac{1}{2}R^{-1}g^T(x)\nabla\Phi^T W_a^* \tag{1.24}$$

$$w = \frac{1}{2\gamma^2}k_d^T(x)\nabla\Phi^T W_w^* \tag{1.25}$$

Subscript a refers to the *actor* and subscript ω refers to the *adverse* NN within the DRL architecture. Following a gradient descent method, updating laws for the critic NN and for the actor NN can be formulated as proposed by Vamvoudakis and Lewis [17]. System stability and convergence of the NN has properly been analyzed by the authors. Developing the adaptation laws for the NN as follows:

$$\dot{\hat{W}}_c = -\alpha_c \frac{\sigma}{(\sigma^T\sigma + 1)^2}(\sigma^T W_c + x^T Q x + u^T R u - \gamma^2 w^T w) \tag{1.26}$$

where $\sigma = \nabla\Phi(f(x) + g(u)u - k_d(x)w)$ and $\alpha_c > 0$ is the updating rate to current status contribution. Persistence of excitation has also been formulated by adding a memory factor to the updating law which compares current with previously undergone costs, states, input, and disturbance inputs. The reader is referred to [17] and [35] for more details on its implementation.

Similarly, the actor NN weights can use the following adaptation law:

$$\dot{\hat{W}}_a = -\alpha_a \left\{ \frac{1}{2} D_a \Big(W_a - W_c \Big) - \frac{1}{4} D_a W_a \left(\frac{\sigma}{(\sigma^T \sigma + 1)^2} \right)^T W_c \right\} \tag{1.27}$$

where $D_a = \nabla\Phi(x) g(x) R^{-1} g(x)^T \nabla\Phi(x)^T$. The presence of $\frac{\sigma}{(\sigma^T\sigma+1)^2}$ guarantees a boundary limit to the adaptation and α_a is the actor NN weight update rate or learning rate [17].

For tuning the adversarial NN:

$$\dot{\hat{W}}_w = -\alpha_w \left\{ \frac{1}{2} E_w \Big(W_w - W_c \Big) + \frac{1}{4\gamma^2} E_w W_w \left(\frac{\sigma}{(\sigma^T \sigma + 1)^2} \right)^T W_c \right\} \tag{1.28}$$

where $E_w = \nabla\Phi(x) k_d k_d^T \nabla\Phi(x)^T$ and α_w is the actor NN weight update rate or learning rate.

Figure 1.8 illustrates an schematic of machine learning control, where the control objective is to minimize a well-defined cost function J within the space of possible control laws. An online learning loop provides experiential data to train the controller. Furthermore, the machine learning control can be a DRL architecture as an Adverse-Actor-Critic structure as described in Subsection 1.4.7.

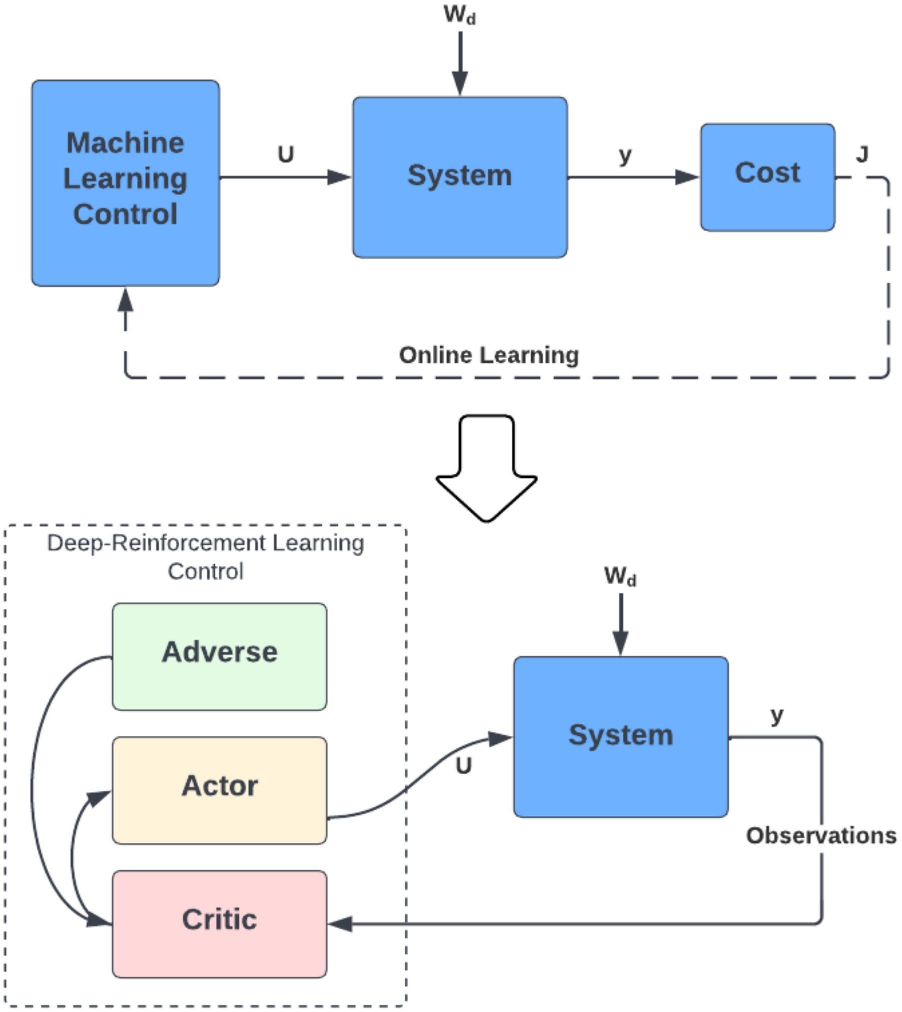


Figure 1.8 Actor-Critic-Adverse Structure

2 Health Management Framework

In this chapter, an extensive discussion of the proposed threat detection methodology for the HMS is provided. As it will be described later in this chapter, the threat detection strategy based on the Artificial Immune System coupled with SVM, creates an optimized HMS able to detect off-nominal conditions.

2.1 Health Management Architecture

A number of authors have recognized the need for a solution to the FDI problem for aerospace vehicles. In short, the literature strongly suggests that machine learning and data-driven techniques hold promising results in addressing this problem [12]. Gao et al. [26] provides a comprehensive survey of data-driven fault detection strategies applied to autonomous systems and processes [26, 28]. In spacecraft applications intelligent algorithms based on Fuzzy logic and Q-learning have been proposed for fault detection in the attitude control systems of satellites [39]. Furthermore, multiple Kalman filters for detection and diagnosis of Reaction Wheel failures [8] and dynamic NN structures for detecting faults in Reaction Wheels [40] include some of the previous research that have implemented data-driven approaches for FDI.

However, in the quest of finding a solution to this widely acknowledged problem, different perspectives have emerged. A promising alternative concept based on data-driven Artificial Immune System (AIS) paradigm has been addressed by [9–11].

The AIS paradigm operates in a similar manner as the immune system of living beings, as it distinguishes between entities that belong to the organism and entities that do not, using the principle of *self/non-self* discrimination to distinguish between different classes of data. However, some limitations when implementing AIS strategies include the large amount of data needed for training to provide information about nominal and off-nominal system behaviours.

When applied to dynamical systems the *self* refers to space of nominal data and the *non-self* refers to a space of off-nominal data, where failure data or off-nominal data is part of the *non-self* and considered abnormal conditions.

In this research, a HMS known as the AISO-SVM (Artificial Inmmue System Optimized Support Vector Machine) is applied. The AISO-SVM which is based on Negative selection-based algorithms and Variable Detector (V-Detector) algorithms coupled with SVM can provide a good candidate to separate the *self/non-self* spaces and thus, improve the HMS detection capabilities.

The following subsections present a description of the HM framework and augmentation algorithms.

2.1.1 Self and Non-self representation

The selection of features that represent the dynamics of the system and are sensitive to nominal and off-nominal conditions is fundamental in the development of the AISO-SVM scheme. These features will directly affect the success and performance of the HMS [10]. The features are variables, usually, functions of time that completely define the dynamics of the system and are expected to have an impact on the abnormal conditions considered, in terms of occurrence, presence, type, severity, and consequences.

The *self* S, is the subset of the system feature space Σ that corresponds to normal flight conditions, and the *non-self* $\bar{S}$ corresponds to the abnormal conditions.

$$\bar{S} \cap S = 0 \qquad \bar{S} \cup S = \Sigma$$

The features are typically normalized to $[0, 1]$ based on known reference values under abnormal conditions.

$$\varphi_i \in \{0, \ldots, 1\} \ . \tag{2.1}$$

Different shapes depending on the dimension of the feature space can be considered for the *self/non-self*, such as hyper-cubes, hyper-spheres and hyper-ellipsoid [41].

Table 2.1 illustrates the key terms and definitions used in the AIS paradigm.

The dimension of the feature space is based on the number of features to relate. For this

Table 2.1 Main Biological Terms

Biological Term	**AIS Paradigm Counterpart**
Self	Space under normal operating conditions
Non-self	Regions of the feature space that are outside of normal operating conditions
Antigen	Set of current features values (*feature point*) at abnormal conditions
Antibody (*detector*)	Data cluster in the non-self feature space

Certain terms adapted from [41]

research, the dimension of the feature space is equal to two and the nominal information provided by each feature is normalized between 0 and 1.

2.1.2 Antibodies generation using Variable Detector Algorithm

The approach to generating antibodies is based on gathering nominal flight data in supervised and controlled conditions that most accurately represent ideal nominal conditions and passing this information through a Variable Detector (V-detector) algorithm that uses a negative selection process to generate the antibodies or detectors for a specific system feature space.

The V-detector algorithm uses an optimization process to obtain a variable radius of each antibody cluster in order to maximize the *non-self* region coverage without overlapping the *self* [42]. The V-detector strategy normalizes the feature data in the range [0, 1] and candidate detectors are selected following an optimization process that considers distance thresholds to indicate the amount of acceptable overlap between detectors, proximity to self and radius. The reader is referred to [43] for more information about the implementation of the V-detector algorithm and the maximum allowable overlap between detectors.

Figure 2.1 shows an example of the V-Detector algorithm that generated antibodies with varying radius size surrounding the self. In this example, a projection relating quaternion q_0

(Feature-20) and Torque in x (Feature-3) is illustrated.

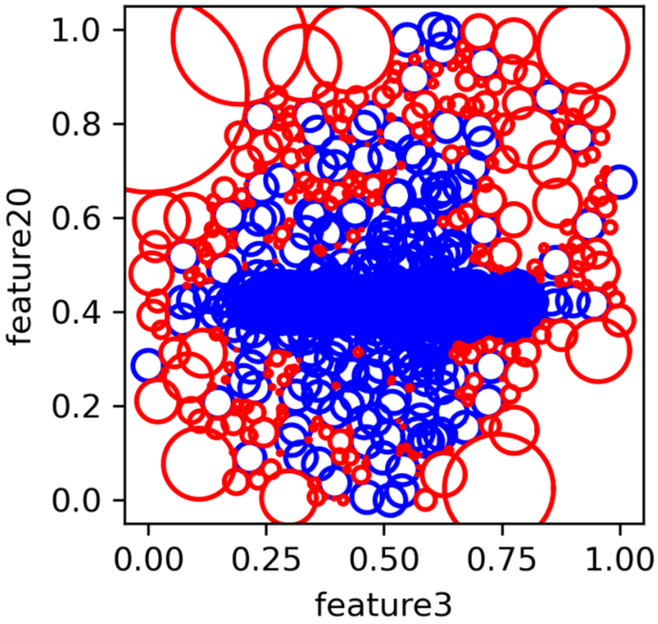


Figure 2.1 Generated detectors (red) surrounding self data (blue) using the V-Detector method.

The antibodies generation process can be stopped after the chosen number of iterations, either when the maximum number of acceptable detectors is reached, or when the desired coverage of the non-self is obtained.

2.1.3 Support Vector Machine Algorithm

Support Vector Machines are supervised learning methods based on machine learning and the principle of structural risk minimization, for developing classification models using two-class classifier methods [14]. These kind of algorithms have been used in many application areas such as pattern recognition, image recognition, fault diagnosis, among others. For the AISO-SVM implementation, the SVM is used to minimize the generalization error and maximize the geometric margin between two classes, in this case, nominal and off-nominal data.

The use of a Kernel function is used to map data into a high-dimensional feature space and find a linear separating hyper-plane to optimize the classification process. The hyper-plane is defined by input training data (support vectors) which generate the decision boundary separating the two classes, where the input vector belongs to one of two classes, class 1 or class 2. Depending on the class, data is labeled as +1 or -1 and the linearly separated decision boundary can be defined by:

$$y = f(x, w) = w^T x + b$$

where $w \in R^n$ is an n-dimensional weight vector and $b \in R$ is a bias value. This equation determines the maximum margin to separate class 1 from class 2. The optimal hyper plane maximizes the margin between two classes in the feature space.

Similarly to Figure 2.1, Figure 2.2 shows an example of the AISO-SVM optimization process to determine the *self/non-self* projection for Feature-20 and Feature-3.

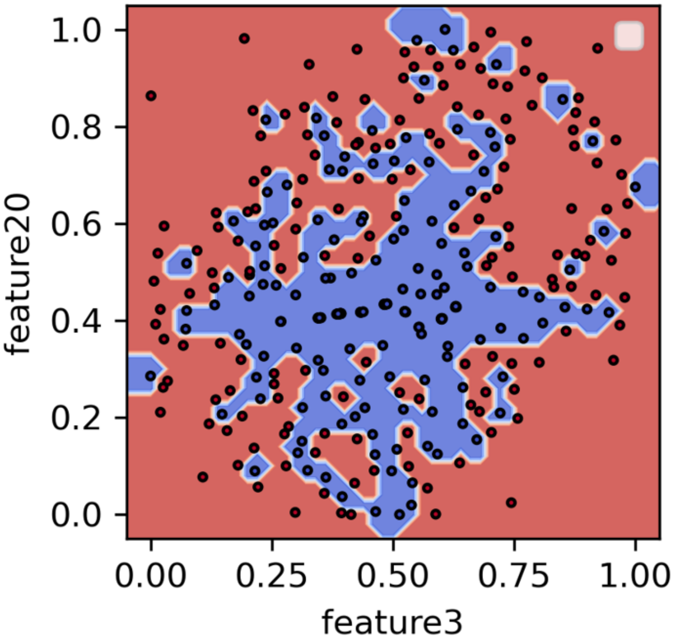


Figure 2.2 AISO-SVM corresponding support vectors for a feature set

2.1.4 Detection Rates and False Alarms

The performance of the HMS is assessed in terms of false alarms (FA) and detection rate (DR). FA are defined as the percentage ratio between the number of time samples when an abnormal condition was declared and the total number of samples for a test under normal conditions, while DR is defined as the percentage ratio between the number of time samples when an abnormal condition was declared and the total number of samples for a test under abnormal conditions.

Detectors can be activated even though no off-nominal conditions are present and this condition generates FA. A knowledge of the estimated percentage of false alarms in the HMS is of importance to assess the performance and accuracy of the models.

When analyzing statistical classification, such as the SVM algorithm, a confusion matrix provides parameters about of the performance of an algorithm. Table 2.2 defines the variables obtained from a confusion matrix.

Table 2.2 Terminology and Derivations from a Confusion Matrix

Terminology	**Definition**
Condition positive (P)	The number of real positive cases in the data
Condition negative (N)	The number of real negative cases in the data
True positive (TP)	A test result that correctly indicates the presence of a condition or characteristic
True negative (TN)	A test result that correctly indicates the absence of a condition or characteristic
False positive (FP)	A test result which wrongly indicates that a particular condition or attribute is present
False negative (FN)	A test result which wrongly indicates that a particular condition or attribute is absent

Based on terminology from Table 2.2, Equation 2.3 is used to calculate DR and Equation 2.2 is used to calculate FA.

$$FA = \frac{FP}{FP + TN} \times 100 \qquad (2.2) \qquad\qquad DR = \frac{TP}{TP + FN} \times 100 \qquad (2.3)$$

2.1.5 Distributed Health Management Architecture

The goal of implementing a distributed HMS within a distributed network is to provide a sense of global immunity and status update, where agents can sense abnormal behavior such as failures and disturbances The AISO-SVM scheme can be applied within a more general multi-spacecraft architecture to increase resilience and autonomy of a cooperative mission. As illustrated in Figure 2.3, the HMS can be separated between a local self and global self, where the conditions of a single system have an impact on the conditions on the network of systems.

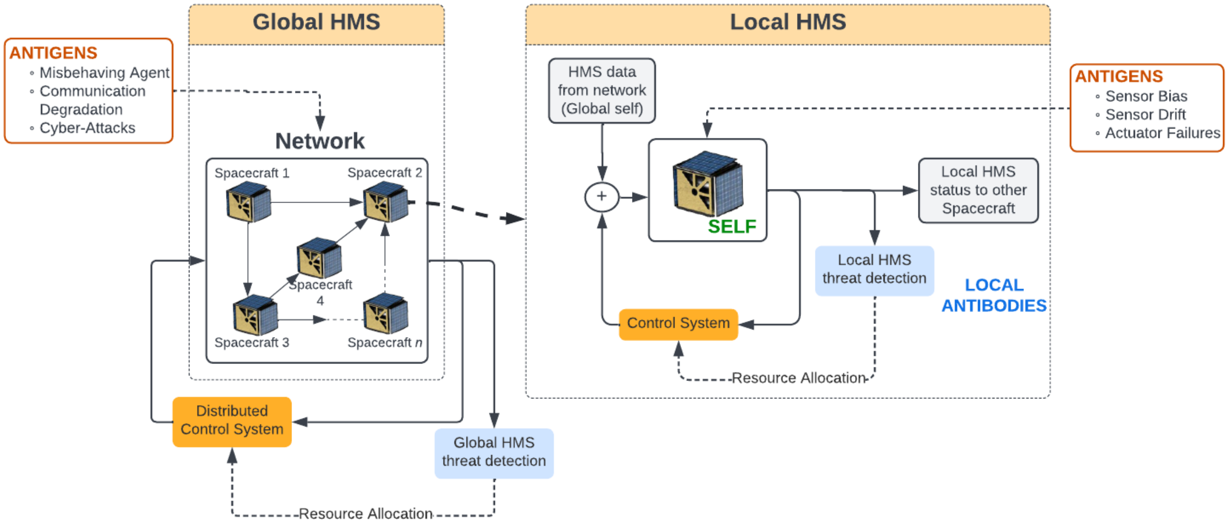


Figure 2.3 Interaction of a single spacecraft within its multi-spacecraft network

During a typical mission operation, spacecraft are vulnerable to failures and are subject to threats. For the local self, a threat might be a hardware failure (e.g., actuators, sensors, onboard computer, power source) or operational abnormal conditions. For the global self, threads and disturbances might include radiation pressure, atmospheric drag, GPS signal obstruction, and cyber-attacks. Even though these threats are unpredictable, their effects limit the operational envelop of the overall mission, but can be abated if assessed and processed in time.

One goal of the distributed HMS scheme is that once a spacecraft detects a failure, a warning can be communicated to the network, so that other cooperative spacecraft notice the novelty and trigger collaborative actions to preserve mission objectives and performance. Hence, each spacecraft within the mission has information of both local and global self conditions.

Similarly to the proposed local HMS for individual vehicles, the global HMS requires nominal flight data from the network of spacecraft to train the model and generate the feature spaces that will best represent the dynamic footprint for the dynamical system. For the global case, the features can be combinations from systems in the network, this approach increases the robustness of the HMS by detecting threats on the network based on combination of states from different spacecraft.

Figure 2.4 illustrates the main architecture applied to the problem of failure detection in multi-spacecraft systems, along with its main elements and their interactions.

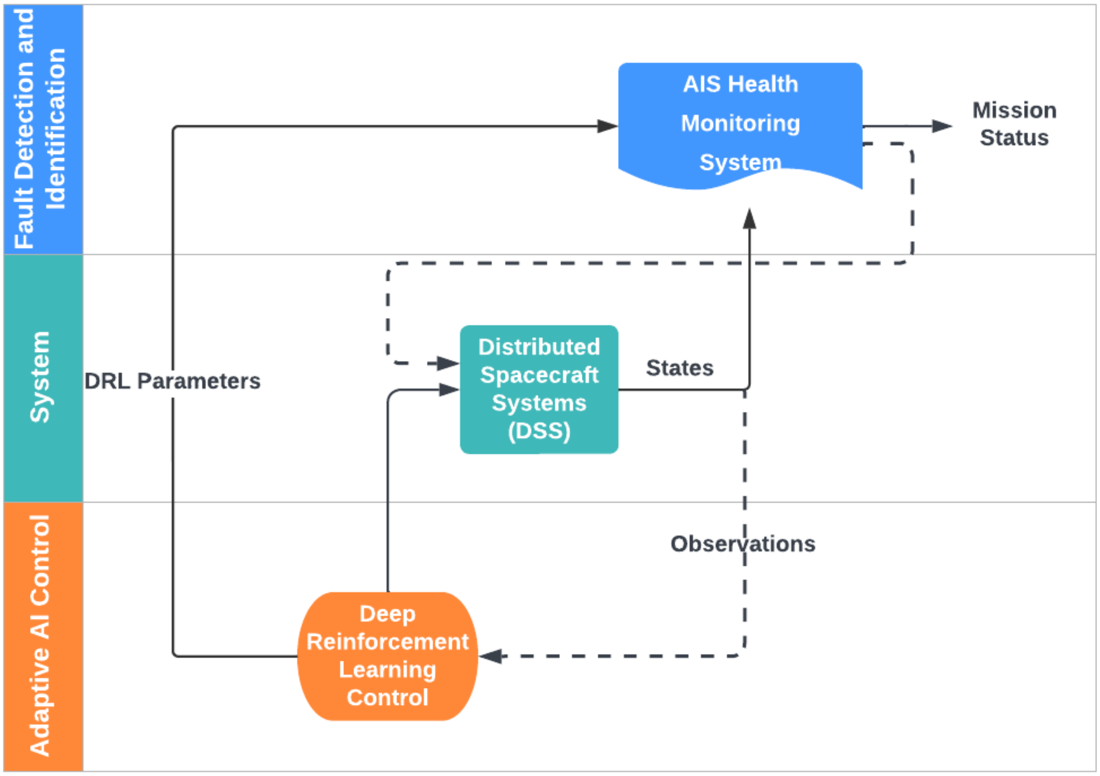


Figure 2.4 DSS HMS framework

3 Attitude Determination and Control

This chapter begins with an overview of the spacecraft attitude determination and is followed by an introduction of the spacecraft dynamics and equations of motion used in the control design. Since the attitude determination and control system uses error quaternions as states and Reaction Wheels as actuators, the spacecraft kinematics using these parameters are reviewed.

3.1 Attitude Determination

The attitude control system is responsible for obtaining desired orientations of the vehicle and pointing instruments such as antennas, sensors, among others. This can be accomplished by implementing systems that generate an external torque to orient the spacecraft, or systems that generate an internal torque. The former includes thruster systems and magnetic systems, which use the Earth's magnetic field. The latter consist of angular momentum exchange systems, such as Reaction Wheels, which are the focus of the work below.

Attitude coordinates are defined as a sets of coordinates that completely describe the orientation of a body relative to some reference. For space applications, attitude is defined as the relative orientation of the Earth-Centered Inertial (ECI) frame to the body fixed frame of the system or vehicle.

Research on attitude coordinates has a long tradition studied by Euler, Jacobi, Hamilton, Cayley, Klein, Rodrigues, and Gibbs which has led to the definition of different sets of attitude coordinates. The proper selection of attitude coordinates can improve the control design by avoiding mathematical and singularities; while, the poor selection of attitude coordinates can limit the performance of a controlled system.

Minimum three coordinates are required to completely describe an orientation; however, any set of three coordinates will have an orientation where coordinates are singular as stated by Junkins and Schaub [44]. In space, singularities are a problem considering the occurrence of large rotations and tumbling effects. Hence, for space applications redundant sets of four or more coordinates are desired to avoid mathematical and geometrical singularities. Euler

Parameters or Quaternions are the most employed attitude coordinates for space.

3.1.1 Quaternion Kinematics

Quaternions or Euler Parameters are the preferred attitude coordinates for large rotations as they provide a redundant and non-singular attitude description. Additionally, their differential equations are linear, improving computational processing and accuracy in the integration.

Quaternions are defined as: $\mathbf{q} = [q_0\ \vec{q_1}\ \vec{q_2}\ \vec{q_3}]^T$, where the first element q_0 is the scalar component and the remaining elements are vector components.

The kinematic differential equation for quaternions has the form:

$$\dot{\vec{q}} = \frac{1}{2}\Omega(\omega)\vec{q} \tag{3.1}$$

where $\Omega(\omega)$ represents the skew symmetric matrix of the angular velocity vector which results in a cross product [45].

$$\Omega(\omega) = \begin{bmatrix} -[\omega\times] & \omega \\ \omega^T & 0 \end{bmatrix} \tag{3.2}$$

Expanding Equation 3.1 in matrix form:

$$\begin{pmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{pmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix}\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}$$

Or by transmutation as presented by Junkins and Schaub [44], the kinematic differential equation has the elegant form:

$$\begin{pmatrix} \dot{q}_o \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{pmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}, \tag{3.3}$$

where $\omega = [\omega_1\ \omega_2\ \omega_3]^T$ is the instantaneous rotation vector of the rigid body. If $\omega(t)$ is known the differential equations for the Quaternions are completely linear.

3.1.2 Spacecraft Kinematics and Kinetics (Rigid Body)

The kinematic equation of motion for rigid bodies is summarized in that the angular momentum rate in the Inertial reference frame is equal to the moment summation:

$$\sum \vec{M} = \frac{d\vec{H}}{dt}^{I} \tag{3.4}$$

The angular momentum $\vec{H}$ taken with respect to the Body reference frame is:

$$\vec{H} = J\vec{\omega}_{B/I} \tag{3.5}$$

where the entries in the J matrix are the moments and products of inertia of the rigid body about its center of mass.

To obtain the angular momentum rate in the Inertial reference frame from Equation 3.4, is necessary to use the transport theorem as follows:

$$\sum \vec{M} = \frac{d\vec{H}}{dt}^{I} = \frac{d\vec{H}}{dt}^{B} + \vec{\omega}_{B/I} \times \vec{H} \tag{3.6}$$

Using the fact that J is constant as seen by the Body reference frame for a rigid body, and using Equation 3.5, the derivative of the angular momentum vector $\vec{H}$ as seen by the Body reference frame is:

$$\frac{d\vec{H}}{dt}^B = \frac{dJ}{dt}^B \vec{\omega}_{B/I} + J\frac{d\vec{\omega}}{dt}^B = J\dot{\vec{\omega}}_{B/I} \tag{3.7}$$

Omitting the subscript notation $\omega_{B/I}$, the angular momentum rate is dictated by Euler's rotational equations of motion as:

$$\vec{M} = \frac{d\vec{H}}{dt}^I = J\dot{\vec{\omega}} + \vec{\omega} \times J\dot{\vec{\omega}} \tag{3.8}$$

From Equation 3.8, we can obtain the following differential equations that describes the evolution of $\vec{\omega}$ in time [44]:

$$[J]\dot{\vec{\omega}} = -[\tilde{\omega}][J]\vec{\omega} + \vec{M}$$

$$\dot{\vec{\omega}} = J^{-1}\{-\vec{\omega} \times (J\vec{\omega}) + \vec{M}\} \tag{3.9}$$

where J is the 3×3 moment of inertia tensor (moments and products of inertia of the rigid body about its center of mass in the Body reference frame), $\vec{M}$ is the net external torque acting on the body, and $\tilde{\omega}$ is a 3x3 skew-symmetric matrix defined as:

$$\tilde{\omega} = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}$$

Therefore, from Euler's Rotational Equation 3.7 and Quaternion's kinematic differential Equation 3.1, the system of equations that describe the dynamics and kinetics of a spacecraft modeled as a rigid body are:

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) = \begin{cases} \dot{\vec{\omega}} = J^{-1}\{-\vec{\omega} \times (J\vec{\omega}) + \vec{M}(t)\} \\ \dot{\vec{q}} = \frac{1}{2}\Omega(\omega)\vec{q} \end{cases} \tag{3.10}$$

where the four quaternions and the three angular rates are the states of the system. The instantaneous net torque on the body $\vec{M}$ is determined by summing the contributions of:

$$\sum \vec{M} = \vec{M}_R + \vec{M}_T + \vec{M}_{RP} + \vec{M}_A + \vec{M}_M$$

where,

$\vec{M}_R$: Moments from Reaction wheels

$\vec{M}_T$: Moments from Thrusters

$\vec{M}_{RP}$: Moments from Solar Radiation Pressure

$\vec{M}_A$: Moments from Aerodynamics

$\vec{M}_M$: Moments from Magnetorques

3.1.3 Equations of Motion for Spacecraft with Reaction Wheels

This subsection develops the equations of motion of spacecraft containing Reaction Wheel control devices.

Reaction Wheels utilize a spinning momentum wheel powered by an internal brushless DC motor and exchanges momentum with the spacecraft body by accelerating or decelerating the wheel, thus changing the wheel momentum magnitude. They act as a momentum transfer and storage device to provide reaction torque and store angular momentum [40]. These devices are commonly used as the actuators in the ACS of spacecrafts.

As stated by Montalvo [45], each Reaction Wheel has its own angular velocity ω_{Ri} and angular acceleration α_{Ri}. The Reaction Wheel is modeled as a disk with finite radius r_{RW} and height h_{RW} and the inertia of each Reaction Wheel is written about the center of mass of the Reaction Wheel as given by Equation 3.11. The subscript R is used to denote that this inertia matrix is about the center of mass of the Reaction Wheel, while the superscript R is used to denote the reference frame of the Reaction Wheel.

$$I_{Ri}^{R} = \begin{bmatrix} m_R r^2/2 & 0 & 0 \\ 0 & (m_R/12)(3r_{RW}^2 + h_{RW}^2) & 0 \\ 0 & 0 & (m_R/12)(3r_{RW}^2 + h_{RW}^2) \end{bmatrix} \tag{3.11}$$

To obtain the total angular momentum of the vehicle, is necessary to account for the inertia of the Reaction Wheel system. Thus, obtaining the Reaction Wheel inertia in the Body reference frame using the parallel axis theorem, the total angular momentum of the vehicle becomes:

$$\vec{H}_S = I_B \vec{\omega}_{B/I} + \sum_{I=1}^{4} I_{Ri}^{B} \omega_{Ri} \hat{n}_{Ri} \tag{3.12}$$

where, $\omega_{B/I}$ is the angular velocity of the Reaction Wheel and the vector $\hat{n}_{Ri}$ is used to define the axis about which the Reaction Wheel rotates.

In a similar way, the total torque placed on the vehicle is equal to the following:

$$\vec{M}_R = \sum_{I=1}^{4} I_{Ri}^{B} \alpha_{Ri} \hat{n}_{Ri} \tag{3.13}$$

where α_{Ri} is the vector of angular acceleration of the Reaction Wheels.

Furthermore, in line with Newton's third law, the reaction torque applied to the spacecraft $\bar{M}_{SC}$ is equal and opposite to the net torque M_R which accelerates or decelerates the wheel:

$$M_R = -M_{SC}$$

Using Reaction Wheels in the control scheme, is possible to solve for the vector of angular accelerations α_{Ri}; however, since $\mathbf{J}$ is a $3 \times N_{RW}$ matrix is not possible to simply invert the matrix and solve for the vector of angular accelerations α_{Ri}. Thus, a minimization routine is required where the solution found also happens to be the lowest amount of angular acceleration. For this, the Lagrange's method is used to find the vector of angular accelerations, obtaining the following expression [45]:

$$\vec{M}_{desired} = \vec{M}_R = \sum_{I=1}^{4} I^B_{Ri}\alpha_{Ri}\hat{n}_{Ri} = \mathbf{J}\vec{\alpha} \tag{3.14}$$

$$\vec{\alpha} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}\vec{M}_{desired} \tag{3.15}$$

3.2 Spacecraft Linear Model

To obtain the linear model of the spacecraft, recall Equation 3.10, which describes the nonlinear dynamics of spacecraft.

$$\dot{\vec{x}} = f(\bar{x}, \bar{u})$$

where the main states that need to be controlled are the angular rates and quaternions, and the control inputs are the moments of the system in the three axes, as illustrated in Equation 3.16:

$$\vec{x} = \begin{bmatrix} p \\ q \\ r \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \qquad \vec{u} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \tag{3.16}$$

as can be noted, q_0 is neglected due to the linear dependency of quaternions.

Performing a Taylor Series expansion of function $f(\bar{x}, \bar{u})$ about equilibrium points (x_e, u_e):

$$\dot{\vec{x}} = f(\bar{x}, \bar{u}) = \cancel{f(\bar{x}_e, \bar{u}_e)} + \frac{\partial f}{\partial \bar{x}}_{x_e,u_e}(\bar{x} - \bar{x}_e) + \frac{\partial f}{\partial \bar{u}}_{\bar{x}_e,\bar{u}_e}(\bar{u} - \bar{u}_e) + \cancel{H.O.T} \tag{3.17}$$

$$\cancel{\dot{\vec{x}}_e} + \Delta\dot{\vec{x}} = A\Delta\bar{x}(t) + B\Delta\bar{u}(t) \tag{3.18}$$

From the equilibrium points of the spacecraft: $p = 0, q = 0, r = 0, q_0 = 1, q_1 = 0, q_2 = 0, q_3 = 0, M_x = 0, M_y = 0, M_z = 0$; matrices A and B are found calculating the Jacobian as follows:

$$A = \frac{\partial f}{\partial x}_{x_e,u_e} \quad B = \frac{\partial f}{\partial u}_{x_e,u_e}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.19}$$

(3.20)

Using the inertias of a reference Cubesat with dimensions:

$$h = 0.30\ m \qquad w = 0.20\ m \qquad d = 0.10\ m$$

and inertias shown in Table 3.1:

Table 3.1 Products of Inertia from Reference Cubesat

$I_{xx} = 0.0874 \frac{kg}{m^2}$	$I_{yy} = 0.1136 \frac{kg}{m^2}$	$I_{zz} = 0.0437 \frac{kg}{m^2}$
$I_{xy} = 0$	$I_{yz} = 0$	$I_{xz} = 0$
$I_{yx} = I_{xy}$	$I_{zy} = I_{yz}$	$I_{zx} = I_{xz}$

Furthermore, studying the controllability of the system using the Controllability matrix:

$$P = \left[\begin{array}{c|c|c|c|c} B & AB & A^2B & \cdots & A^{n-1}B \end{array} \right],$$

It is found that $rank(P) = N$, where N = 6 is the dimension of matrix A. Hence, the system is controllable.

Attitude determination and control for spacecraft have been extensively researched. One approach for attitude control is a LQR controller augmented with DLR applied to a linearized version of a spacecraft state space dynamics, which will be implemented in this thesis. The design parameters of such controller are presented in Subsection 5.4.1.

3.3 Nonlinear Dynamic Inversion (NLDI) Controller

Recalling the system of equations of motion in nonlinear state space form from Equation 3.10:

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) = \begin{cases} \dot{\vec{\omega}} = J^{-1}\{-\bar{\omega} \times (J\bar{\omega}) + \vec{M}(t)\} \\ \dot{\vec{q}} = \frac{1}{2}\Omega(\bar{\omega})\bar{q} \end{cases}$$

where the state vector corresponds to $\vec{x} = [\bar{\omega}, \delta\bar{q}]^T$, the control law for the NLDI controller is proposed by Markley and Crassidis [46] as:

$$\vec{M}(t) = \bar{\omega} \times (J\bar{\omega}) + J\bar{u}_v(t) \tag{3.21}$$

where $u_v(t)$ is a virtual controller or pseudo-controller chosen to stabilize $\bar{\omega}$ to zero and $\delta\bar{q}$ to the identity Quaternion.

$$u_v(\bar{q}, \bar{\omega}) = -K_\Omega\bar{\omega} - K_q\delta\bar{q}_{1:3} \tag{3.22}$$

where K_q and K_Ω are pseudo gains defined as:

$$K_q = \begin{bmatrix} K_{q_1} & 0 & 0 \\ 0 & K_{q_2} & 0 \\ 0 & 0 & K_{q_3} \end{bmatrix} \qquad K_\Omega = \begin{bmatrix} K_{\omega_x} & 0 & 0 \\ 0 & K_{\omega_y} & 0 \\ 0 & 0 & K_{\omega_z} \end{bmatrix} \tag{3.23}$$

Substituting Equation 3.22 in Equation 3.21, gives the following control law:

$$\vec{M}(t) = \bar{\omega} \times (J\bar{\omega}) + J[-K_q\delta\ \bar{q}_{1:3} + K_d\ \bar{\omega}] \tag{3.24}$$

Finally, the closed-loop system after implementing the control law in Equation 3.24 has the form:

$$\begin{cases} \dot{\bar{\omega}} = \bar{u}_v(t) \\ \delta\dot{\bar{q}} = \frac{1}{2}\Omega(\omega)\delta\bar{q} \end{cases} \tag{3.25}$$

where $\bar{u}_v$ is a virtual controller used to obtain the desired dynamics

$$\bar{u}_v = K_v(\bar{\omega}_d - \bar{\omega}) \tag{3.26}$$

Here ω_d is represented in terms of the quaternion error, hence, the virtual controller takes the form:

$$\bar{u}_v(\delta\bar{q}, \bar{\omega}) = -K_q\delta\bar{q}_{1:3} - K_\Omega\bar{\omega} \tag{3.27}$$

A slight modification to the virtual controllers is introduced to guarantee the shortest path in final orientation, as stated by Markley and Crassidis [46]:

$$u_v(q, \omega) = -K_\Omega\bar{\omega} - K_q sign(\delta q_4)\delta\bar{q}_{1:3} \tag{3.28}$$

$$\vec{M}(t) = \bar{\omega} \times (J\bar{\omega}) + J[-K_\Omega\bar{\omega} - K_q sign(\delta q_4)\delta\bar{q}_{1:3}] \tag{3.29}$$

4 Simulation Environment

This chapter describes the simulation environment built to simulate multi-spacecraft missions. First, a description of the spacecraft dynamics governed by the Clohessy-Wiltshire-Hill (CWH) equations. A representaton of the simulation framework is shown and a detailed description of the Reaction Wheel model developed is presented. Finally, three different failure scenarios affecting the ACS are described and compared with respect to nominal conditions.

4.1 Reference Mission: Inspection of a Target Spacecraft

The design reference mission consists of a fleet of spacecraft in low Earth orbit performing on-orbit inspection of a target space asset. The fleet consists of four CubeSats that are deployed from standard dispensers (e.g. Nanoracks or P-POD) and injected to relative orbits around the target spacecraft. Each CubeSat will have mounted sensors for inspection, whose pointing can be precisely controlled by the Attitude Determination and Control System (up to 0.003 deg for each axis). The CubeSats will also be equipped with a cold/warm propulsion unit that allows reconfiguration to different relative orbits. However, to minimize the expenditure of limited fuel budget, CubeSats will utilize stable Passive Relative Orbits (PROs) throughout the inspection and only transfer when the information gain justifies the fuel cost of the transfer.

The spacecraft fleet will collectively perform inspection of the target space asset (Chief), which requires close coordination. The CubeSats within the fleet will constantly communicate information gathered on the points of interest (POI) on the target, which is then utilized to make global decision on which POIs each CubeSat should focus in the next planning horizon. Note that delays in communication due to degradation or intentional misinformation on the POIs due to external attacks could negatively affect the overall performance of the coordinated inspection.

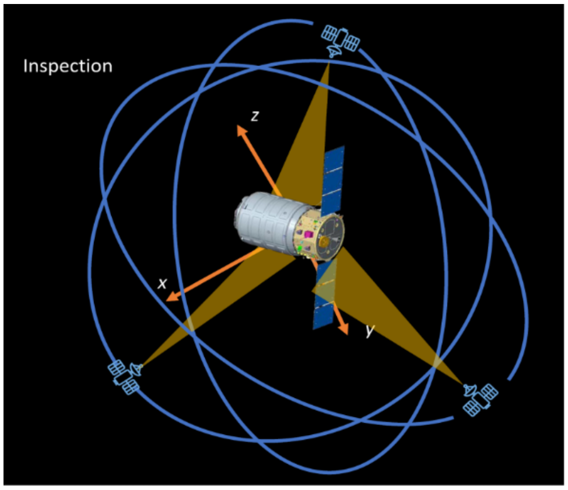


Figure 4.1 Pointing Control for Inspection of the Target Spacecraft, from Nakka et al. [4]

The observers (deputy) spacecraft correspond to a 6U CubeSat (20 cm × 10 cm × 30 cm) with mass of $m_i = 10$ kg and are equipped with a Reaction Wheel configuration described in detail in Subsection 4.3.1.

4.2 Spacecraft Dynamics

The spacecraft fleet will perform inspection within close proximity of the target space asset, allowing dynamics to be expressed using the Clohessy-Wiltshire-Hill (CWH) equations, which provides a linear approximation of the relative dynamics between the target space asset and each deputy spacecraft in the fleet. The CWH equations assume the target space asset to be in a circular orbit and that the Earth is perfectly spherical (ignoring the perturbation effect of the J_2 term). The 3-DOF CWH equations, originally developed in [47], describe the motion of a given spacecraft in the Local-Vertical, Local-Horizontal (LVLH) coordinate system. The target is located at the origin of this coordinate system, where the x direction points radially away from Earth toward the target, the y direction points in the direction of the orbital velocity of the target space asset and the z direction points in the direction of the angular momentum of the target.

The orbit parameters the target spacecraft are defined using orbital elements, such as orbit altitude r, eccentricity e, inclination i, right ascension of the ascending node Ω, argument of perigee ω and true anomaly f_0. Once the target orbit has been located, the LVLH rotating frame $\mathscr{L}$ or Euler-Hill frame can be defined for the target spacecraft to locate the deputy spacecraft. A representation of the LVLH and ECI reference frames is shown in Figure 4.2

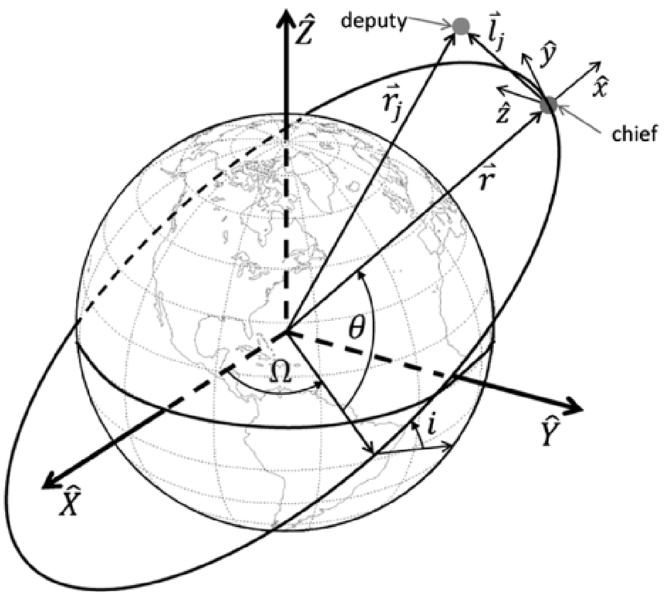


Figure 4.2 ECI frame $\hat{X}, \hat{Y}, \hat{Z}$ and LVLH frame $\hat{x}, \hat{y}, \hat{z}$, centered at the chief spacecraft. The deputy position vectors in the rotating and inertial reference frames are denoted by l_j and r_j, respectively. From: Morgan and Chung [5]

The CWH equations in the LVLH coordinate system are given in Equation 4.1 as:

$$\begin{aligned} \ddot{x} &= 3n^2x + 2n\dot{y} \\ \ddot{y} &= -2n\dot{x} \\ \ddot{z} &= -n^2z \end{aligned} \tag{4.1}$$

where the mean motion of the target object, n, is defined in Equation 4.2 as

$$n = \sqrt{\frac{\mu}{r_0^3}} \tag{4.2}$$

where μ is the standard gravitational parameter and r_0 is the radius of the orbit of the target spacecraft.

It is convenient to obtain the state-space representation $\dot{\bar{x}}(t) = A\bar{x}(t)$ of the differential Equations in 4.1, and define the CWH equations in State Transition Matrix (STM) form as follows:

$$\bar{x}(t) = e^{A(t-t_0)}\bar{x}(t_0). \tag{4.3}$$

where the state vector is $\bar{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$ and the A matrix is given by:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix}$$

If the initial states are known, the STM can be used to find the states at any given time. Thus, using Equation 4.3, the closed form solution of the CWH equations is given in Equation 4.4.

$$\begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} 4 - 3\cos nt & 0 & 0 & \frac{1}{n}\sin nt & \frac{2}{n}(1-\cos nt) & 0 \\ 6(\sin nt - nt) & 1 & 0 & -\frac{2}{n}(1-\cos nt) & \frac{1}{n}(4\sin nt - 3nt) & 0 \\ 0 & 0 & \cos nt & 0 & 0 & \frac{1}{n}\sin nt \\ 3n\sin nt & 0 & 0 & \cos nt & 2\sin nt & 0 \\ -6n(1-\cos nt) & 0 & 0 & -2\sin nt & 4\cos nt - 3 & 0 \\ 0 & 0 & -n\sin nt & 0 & 0 & \cos nt \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} \tag{4.4}$$

The CWH equations are good linear approximations of the relative dynamics for spacecraft, with the benefit of obtaining a linear time-invariant closed-form solution. Nonetheless, neglecting J_2 perturbations in LEO can lead to large errors in the motion predicted. The work developed by Morgan and Chung [5], aims to extend the nonlinear relative dynamics taking into account the effects of J_2 perturbations on the swarm motion.

For a spherical Earth, the gravity gradient vector has a constant direction and the magnitude depends only on r. The Keplerian gravity gradient vector is:

$$\nabla U = \frac{\mu}{r^2}\hat{x}$$

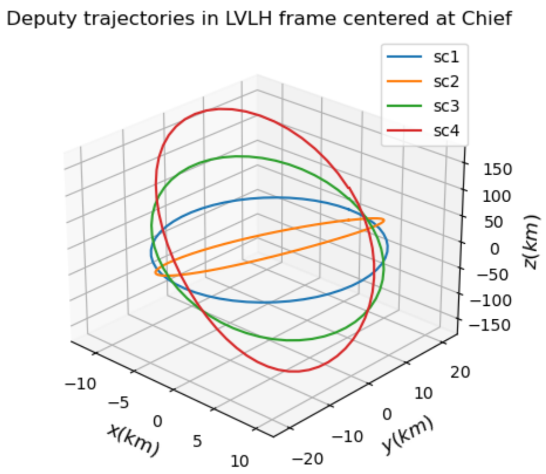
However, a change in the gravity gradient vector arises with the addition of J_2 disturbance. Thus, under the influence of J_2 the gravity gradient vector is:

$$\nabla U_{J_2} = \frac{\mu}{r^2}\hat{x} + \frac{K_{J2}}{r^4}(1 - 3\ sin^2 i\ sin^2\theta)\hat{x} + \frac{K_{J2}\ sin^2 i\ sin2\theta}{r^4}\hat{y} + \frac{K_{J2}\ sin2i\ sin\theta}{r^4}\hat{z}$$

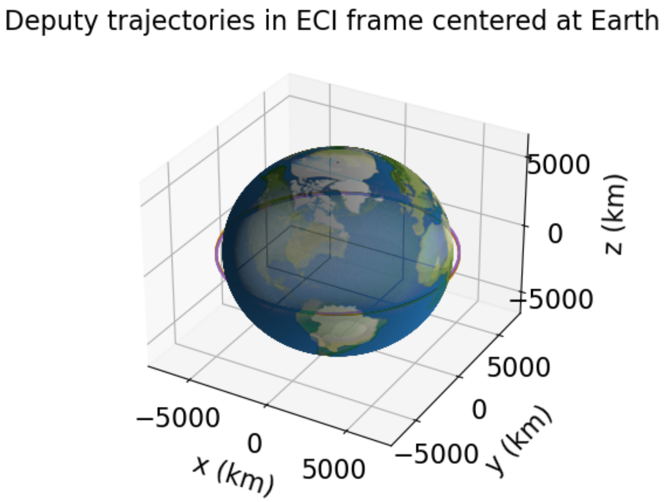
Considering that ∇U_{J_2} is not aligned with the LVLH frame, these equations must be transformed back to the LVLH performing the respective rotations. As a result, Equation 4.5 is used to obtain the desired initial conditions for PROs accounting for J_2 disturbance. The reader is referred to [5] for further theoretical analysis.

$$\begin{bmatrix} \dot{x}_{0,L,J2} \\ \dot{y}_{0,L,J2} \\ \dot{z}_{0,L,J2} \end{bmatrix} = \begin{bmatrix} \frac{3}{2}c_\alpha s_\alpha c_\beta - c_\alpha^2 s_\beta^2 t_{\theta_0} & \frac{1}{2}c_\alpha^2 c_\beta + 2s_\alpha^2 c_\beta - c_\alpha s_\alpha s_\beta^2 t_{\theta_0} & 2s_\alpha s_\beta + c_\alpha c_\beta s_\beta t_{\theta_0} \\ -2c_\alpha^2 c_\beta - \frac{1}{2}s_\alpha^2 c_\beta - c_\alpha s_\alpha s_\beta^2 t_{\theta_0} & -\frac{3}{2}c_\alpha s_\alpha c_\beta - s_\alpha^2 s_\beta^2 t_{\theta_0} & -2c_\alpha s_\beta + s_\alpha c_\beta s_\beta t_{\theta_0} \\ -\frac{1}{2}s_\alpha s_\beta + c_\alpha c_\beta s_\beta t_{\theta_0} & -\frac{1}{2}c_\alpha s_\beta + s_\alpha c_\beta s_\beta t_{\theta_0} & -c_\beta^2 t_{\theta_0} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{4.5}$$

Applying the CWH equations under J_2 perturbation with initial conditions given by Equation 4.5 and orbit parameters from the target spacecraft, the following stable relative orbits shown in Figure 4.3a are obtained for four observers in the LVLH reference frame, while Figure 4.3b shows the orbit trajectories in the ECI reference frame.



(a) Spacecraft PROs in LVLH Coordinate System



(b) Spacecraft orbits in ECI Coordinate System

Figure 4.3 Relative orbits in the LVLH and the ECI coordinate frames.

4.3 Simulation Framework

The architecture of the simulation environment presented in this thesis is modular and flexible enough to simulate multi-spacecraft missions, process flight data at normal and abnormal conditions, test different failure and disturbance scenarios, and perform risk assessments associated to such missions. The main elements of the simulation environment and their interconnections are presented in Figure 4.4. A simulation time step of $dt = 0.1$ s was used. Furthermore, a detailed description of the Reaction Wheel model is presented in Subsection 4.3.1.

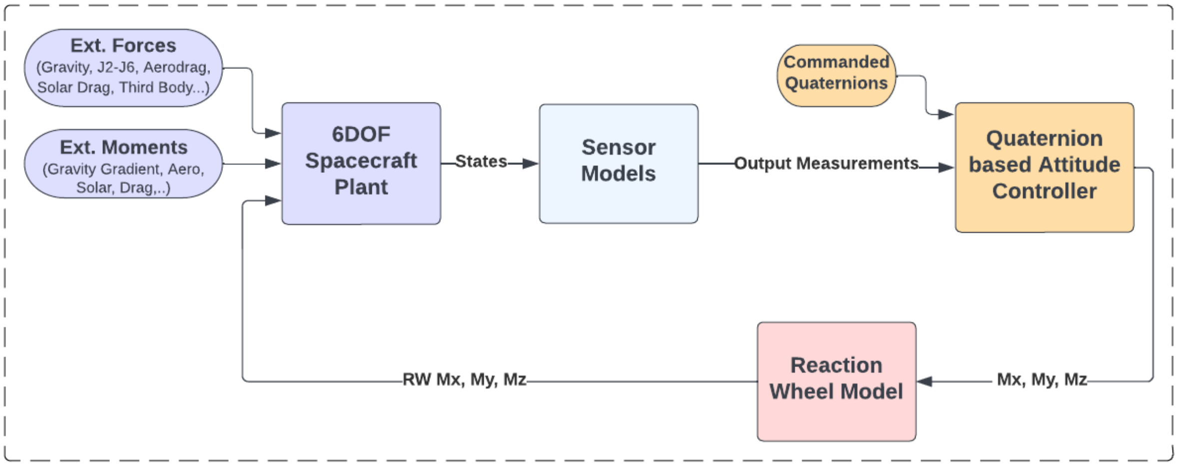


Figure 4.4 Model Overview of Simulation Environment

4.3.1 Reaction Wheel Model

Momentum exchange systems such as Reaction Wheels are often preferred for Attitude Control System (ACS) because of their fuel-free operation and precise combination of torque and momentum storage necessary for pointing precision. The ACS goal is to stabilize the system and orient it in the desired attitude despite disturbance torques or failures acting on it. Thus, providing a high fidelity mathematical model of Reaction Wheel configuration allows to obtain realistic responses to evaluate different failures in the ACS subsystem.

The Reaction Wheel model follows a redundant set of four Reaction Wheels arranged in a tetrahedron configuration as shown in Figure 4.5 with parameters based on a commercially developed RWP015 by Blue Canyon Technologies, with the following specifications:

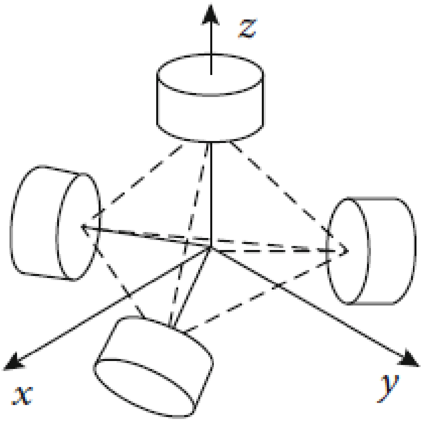


Figure 4.5 Tetrahedral RW configuration

RWP015 Specs	
Momentum	0.015 Nms
Max. Torque	0.004 Nm
Mass	0.130 kg
Volume	42 x 42 x 19 mm
Voltage	12 VDC
Static Unbalance	0.25 g-mm
Dynamic Unbalance	2.5 g-mm^2

Table 4.1 RWP015 Specifications

A detailed block diagram of the Reaction Wheel model is shown in Figure 4.6, where the bottom part of the Figure shows the four Reaction Wheels control scheme; where the input is a commanded voltage that controls motor current, and as a result, angular velocity of the wheel calculated from the desired moment. The upper part of the Figure shows the components of the Reaction Wheel Model which includes the DC Motor, Friction, Rotational Imbalances and Thermals, where Friction generates torque loss in the system and is subtracted from the overall torque.

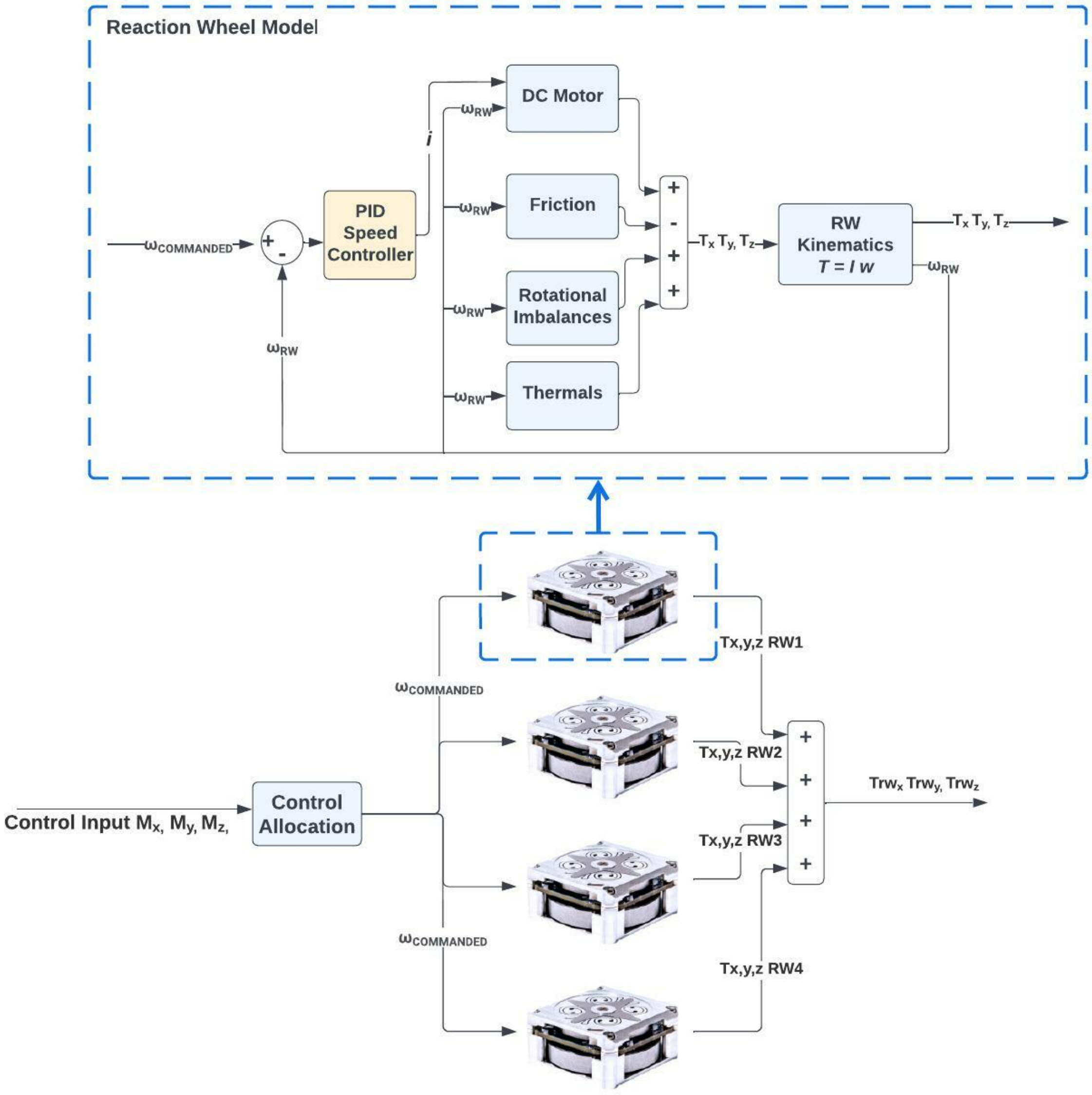


Figure 4.6 Reaction Wheel Model

4.3.2 Failure Scenarios

Failures in small satellites can be classified into two categories according to report [7]. The first category being mission failure and the second category being partial mission failure. The former category happens when small spacecraft fails to operate altogether or only achieves very little of the mission objectives, while the latter category happens when small spacecraft experiences component failures. However, categorizing a failure as a mission failure or a partial mission failure depends on the objectives the spacecraft was supposed to achieve before the malfunction. Furthermore, in report [7] is stated that between the years 2000 to 2016, 41.3% of all small satellites launched experienced total or partial mission failure. From these, 6.1% were launch vehicle failures, 11% were partial mission failures, and 24.2% were total mission failures. Nonetheless, as mission objectives become progressively more challenging, also mission failure tends to increase as as there exist more chances for mission failure.

An illustration of the spacecraft simulated in this study is shown in Figure 4.7. The failure scenarios considered in this research are failures applied in the ACS of *Spacecraft* 2 (SC#2) in the network.

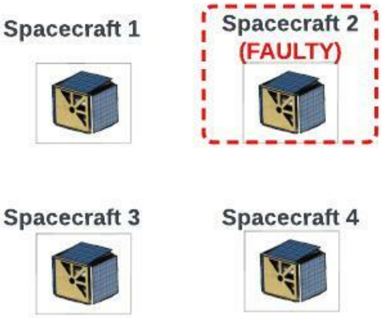


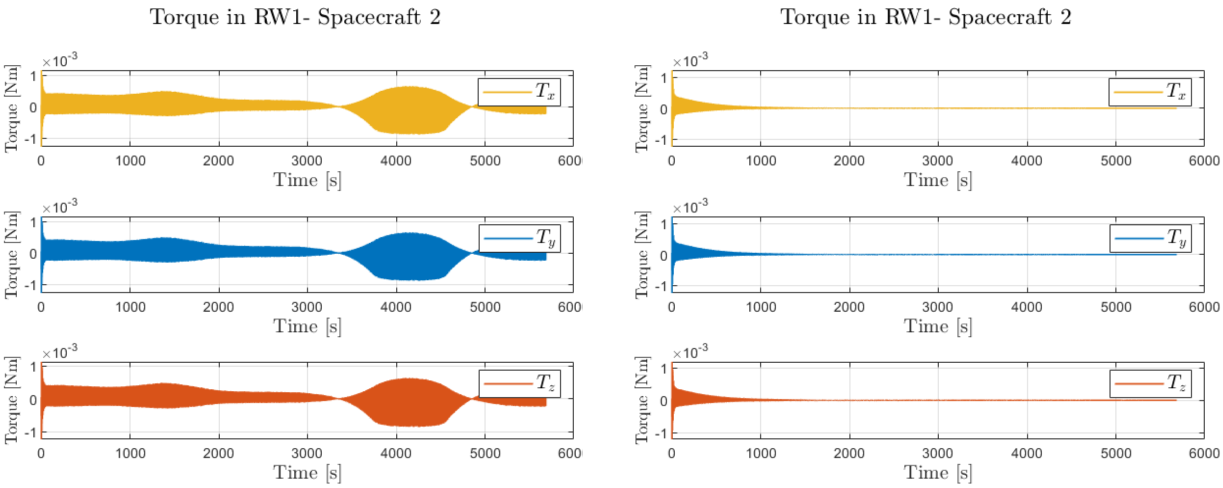
Figure 4.7 Spacecraft 2 Faulty in the network

Failures are injected in the RWs as these actuators are highly prone to malfunctions. Considering the redundant RW configuration described in Subsection 4.3.1, malfunctions in only one RW will not cause significant impact on the attitude tracking performance as the system is able to maintain desired orientation with a minimum of three RW. However, when two or more RW are affected, the attitude controller is not able to fulfill the attitude tracking requirements.

Despite failing only one RW will not affect the attitude control of the system, it is important to detect and identify any type of malfunction, as they can affect system performance in other ways, such as requiring more effort from other actuators or systems in order to comply with mission objectives. For instance, failing one RW will require more effort from other RWs to continue performing maneuvers, which could lead to mechanical wear and saturation problems. Hence, obtaining real-time information about the health status of the system is relevant for decision making processes due to the limitations impaired for not operating in nominal conditions.

In this study, three different failure scenarios affecting the ACS were tested and compared with respect to nominal conditions to test the HMS detection capabilities. These failures affect only the ACS of SC#2 within a network of four spacecraft. These failure scenarios are described as follows:

- **Failure Case I:** This failure is started a $t = 0\ s$ and consists of a progressive drop in current i that provides power to the brushless motor of RW#1 in SC#2. As a result, a progressive drop in the torque $T_{x_{RW1}}, T_{y_{RW1}}, T_{z_{RW1}}$ is experimented. This type of failure generates the change in torque shown in Figure 4.8 compared to nominal conditions operation.



(a) Nominal Torque in RW1 - Spacecraft 2 *(b)* Failure Case I Torque in RW1 - Spacecraft 2

Figure 4.8 Effect of Failure Case I

The quaternion attitude tracking performance when this failure occurs is shown in Figure 4.9. As mentioned before, failing only one RW does not have significant impact on the attitude tracking performance.

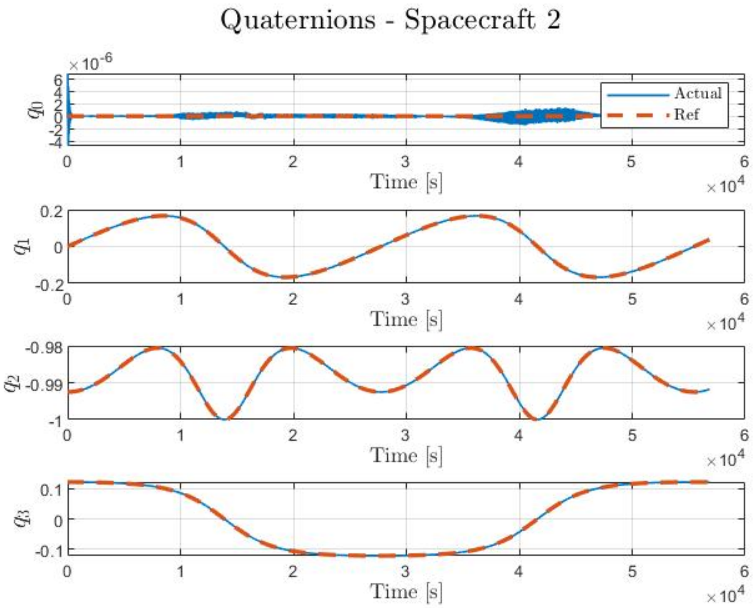
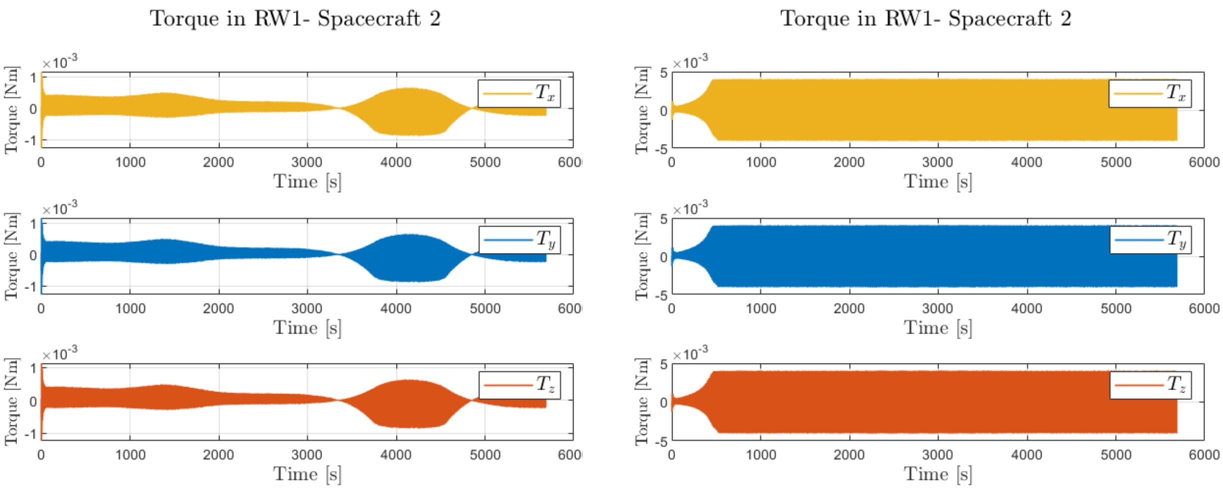


Figure 4.9 Attitude tracking performance in Spacecraft 2 when Failure Case I is injected

- **Failure Case II:** This failure is started a $t = 0\ s$ and consists of saturating a RW to the maximum allowed torque 0.004 N-m. This occurs when the actuators stores enough momentum to exceed the maximum speed of the wheel, resulting in a maximum allowed torque during operation. This failure is simulated by a progressive increase in the angular rate $\bar{\omega}$ of RW#1 in SC#2, which causes the maximum allowed torque in $T_{x_{RW1}}$, $T_{y_{RW1}}$, $T_{z_{RW1}}$.

  This type of failure generates the change in torque shown in Figure 4.10, with respect to nominal conditions operation.



(a) Nominal Torque in RW1 - Spacecraft 2

(b) Failure Case II Torque in RW1 - Spacecraft 2

Figure 4.10 Effect of Failure Case II

The attitude tracking performance when Failure Case II occurs is shown in Figure 4.11. As stated before, failing only one RW does not have significant impact on the attitude tracking.

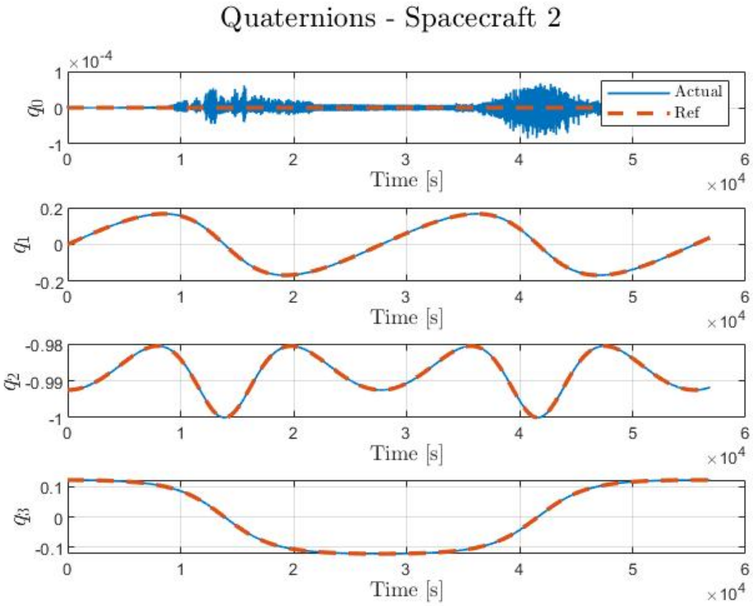
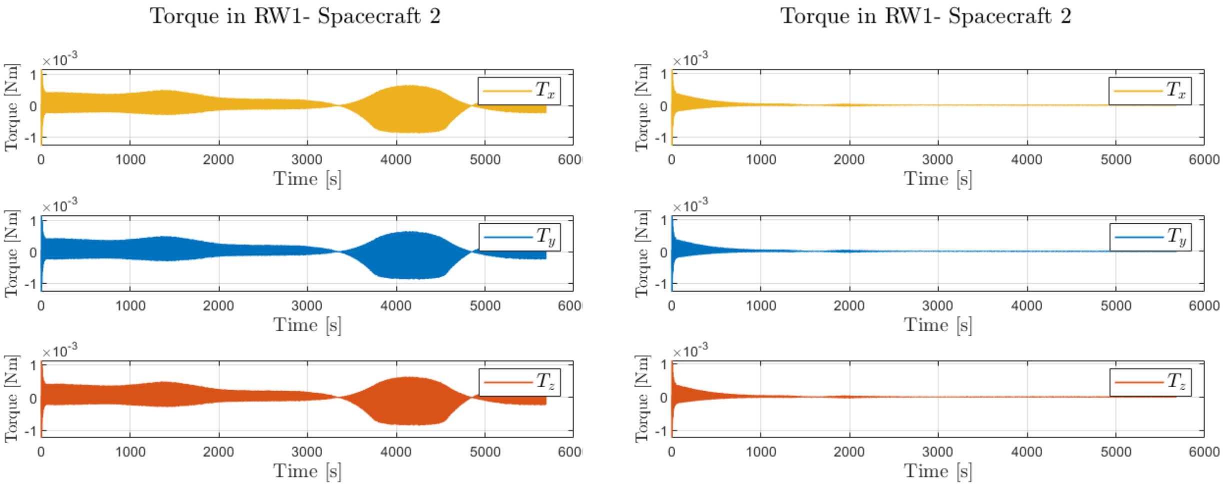


Figure 4.11 Attitude tracking performance in Spacecraft 2 when Failure Case II is injected

- **Failure Case III:**

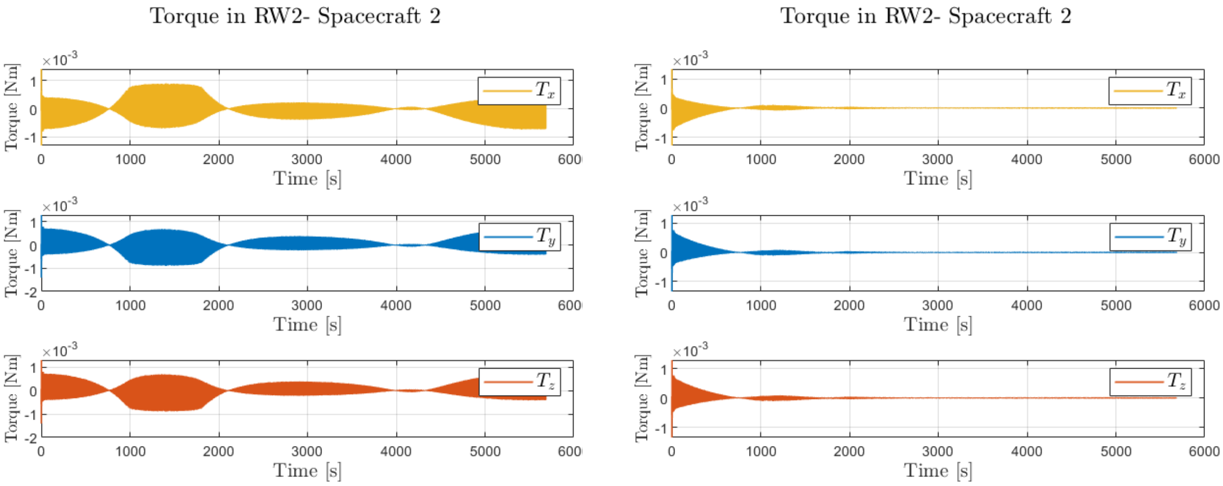
This failure is started a $t = 0$ s and consists of a progressive drop in current i applied to RW#1 and RW#2 of SC#2, which in turn will cause a progressive drop in their produced torques $T_{x_{RW1}}$, $T_{y_{RW1}}$, $T_{z_{RW1}}$ and $T_{x_{RW2}}$, $T_{y_{RW2}}$, $T_{z_{RW2}}$. This failure is similar to Failure Case I, but this time applied to two RWs.

Figure 4.12 and Figure 4.13 shows torque changes with respect to the nominal conditions operation of RW#1 and RW#2.



(a) Nominal Torque in RW1 - Spacecraft 2 *(b)* Failure Case III Torque in RW1 - Spacecraft 2

Figure 4.12 Effect of Failure Case III



(a) Nominal Torque in RW2 - Spacecraft 2 *(b)* Failure Case III Torque in RW2 - Spacecraft 2

Figure 4.13 Effect of Failure Case III

The attitude tracking performance when Failure case III occurs is shown in Figure 4.14. This time, affecting two RWs causes the attitude controller to not able to fulfill the attitude tracking requirements as depicted in Figure 4.14.

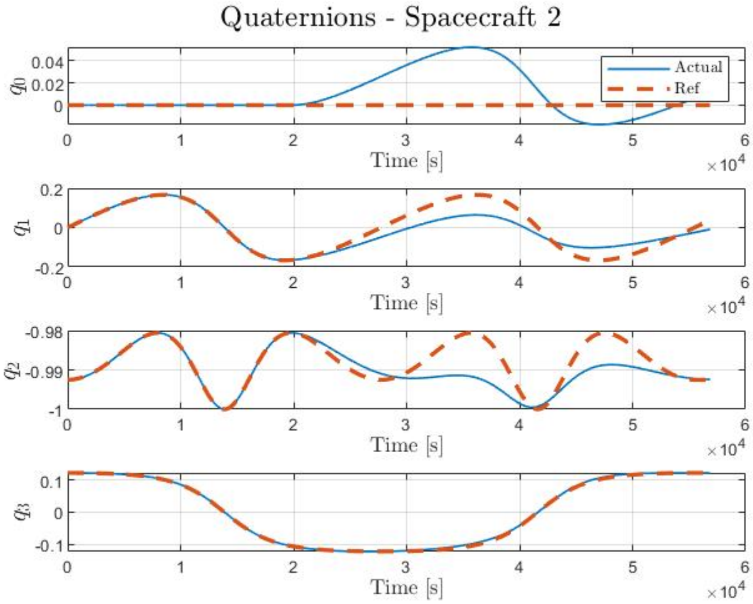


Figure 4.14 Attitude tracking performance in Spacecraft 2 when Failure Case III is injected

5 Numerical Simulations and Results Analysis

This section summarizes the results of the HMS performance on detecting the simulated three failures described in Subsection 4.3.2, in one of the deputy spacecraft within a network performing the inspection mission. Furthermore, the comparison of different attitude controllers under disturbances and failure scenarios are analyzed in terms of tracking performance and control effort.

5.1 HMS training

To train the AISO-SVM algorithm, enough flight data was gathered from the spacecraft system dynamics following optimized orbits and different attitude combinations in nominal conditions. The selection of features that represent the dynamics of the system and are sensitive to nominal and off-nominal conditions is the main step in the HMS training process [10].

The features shown in Table 5.1 were selected as primary features for their effectiveness in the HMS performance analysis and include 34 states. Nonetheless, the HMS strategy relies on the assumption that some features may be relevant in capturing some types of failures and may lack relevance with respect to others.

Table 5.1 Features/States Selected from Spacecraft

Feature ID	**Feature Variable**	**Feature Name**
0	$T_{x_{RW1}}$	Torque in x axis in $_{RW1}$
1	$T_{y_{RW1}}$	Torque in y axis in $_{RW1}$
2	$T_{z_{RW1}}$	Torque in z axis in $_{RW1}$
3	$T_{x_{RW2}}$	Torque in x axis in $_{RW2}$
4	$T_{y_{RW2}}$	Torque in y axis in $_{RW2}$
5	$T_{z_{RW2}}$	Torque in z axis in $_{RW2}$
6	$T_{x_{RW3}}$	Torque in x axis in $_{RW3}$
7	$T_{y_{RW3}}$	Torque in y axis in $_{RW3}$
8	$T_{z_{RW3}}$	Torque in z axis in $_{RW3}$
9	$T_{x_{RW4}}$	Torque in x axis in $_{RW4}$
10	$T_{y_{RW4}}$	Torque in y axis in $_{RW4}$
11	$T_{z_{RW4}}$	Torque in z axis in $_{RW4}$
12	$\omega_{desired_{RW1}}$	Desired angular velocity in $_{RW1}$
13	$\omega_{desired_{RW2}}$	Desired angular velocity in $_{RW2}$
14	$\omega_{desired_{RW3}}$	Desired angular velocity in $_{RW3}$
15	$\omega_{desired_{RW4}}$	Desired angular velocity in $_{RW4}$
16	$i_{error_{RW1}}$	Current error in $_{RW1}$
17	$i_{error_{RW2}}$	Current error in $_{RW2}$
18	$i_{error_{RW3}}$	Current error in $_{RW3}$
19	$i_{error_{RW4}}$	Current error in $_{RW3}$
20	$q_{0actual}$	Quaternion actual $q0$
21	$q_{1actual}$	Quaternion actual $q1$
22	$q_{2actual}$	Quaternion actual $q2$
23	$q_{3actual}$	Quaternion actual $q3$
24	q_{0error}	Quaternion error $q0$
25	q_{1error}	Quaternion error $q1$
26	q_{2error}	Quaternion error $q2$
27	q_{3error}	Quaternion error $q3$
28	$\omega_{actual_{RW1}}$	Actual angular velocity in $_{RW1}$
29	$\omega_{actual_{RW2}}$	Actual angular velocity in $_{RW2}$
30	$\omega_{actual_{RW3}}$	Actual angular velocity in $_{RW3}$
31	$\omega_{actual_{RW4}}$	Actual angular velocity in $_{RW4}$
32	$p_{s/c}$	Roll rate
33	$q_{s/c}$	Pitch rate
34	$r_{s/c}$	Yaw rate

The HMS training process was performed using the V-Detector algorithm and the AISO-SVM two-class classifier algorithm to generate the antibodies or detectors. The total number feature spaces of N_{ss} projections that can be built for a complete set of $N = 34$ features and

$N_{max} = 2$ feature combinations is:

$$N_{ss,2D} = \frac{N!}{N_{max}!(N - N_{max})!} = \frac{34!}{2!32!} = 561 \tag{5.1}$$

However, from the possible 561 feature combinations, only 73 were analyzed. The selected 73 feature space combinations are shown in Table 5.2.

Remark 1: In the case of the multi-agent systems, it is not necessary to select the same number of features for all agents. The goal is to have features that will create a unique representation of the agents within the network in order to detect specific abnormal conditions.

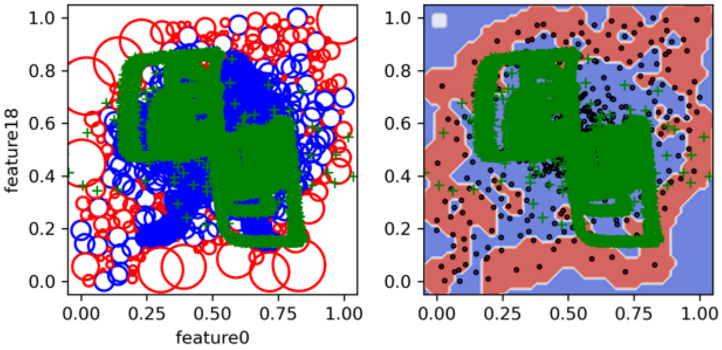
Remark 2: Using 2D projections from a set of features allows intuitive analytical and graphical representations of distances, thresholds, and boundaries [9].

Table 5.2 List of Considered $2 - D$ Self Projections from $Spacecraft$ 2

Self	Features		Self	Features		Self	Features	
1.	$T_{x_{RW1}}$	$T_{y_{RW1}}$	26.	$T_{x_{RW1}}$	q_{2error}	51.	$T_{y_{RW2}}$	$q_{1actual}$
2.	$T_{x_{RW1}}$	$T_{z_{RW1}}$	27.	$T_{x_{RW1}}$	q_{3error}	52.	$T_{y_{RW2}}$	$q_{2actual}$
3.	$T_{x_{RW1}}$	$T_{x_{RW2}}$	28.	$T_{x_{RW1}}$	$\omega_{actual_{RW1}}$	53.	$T_{y_{RW2}}$	$q_{3actual}$
4.	$T_{x_{RW1}}$	$T_{y_{RW2}}$	29.	$T_{x_{RW1}}$	$\omega_{actual_{RW2}}$	54.	$T_{y_{RW2}}$	q_{0error}
5.	$T_{x_{RW1}}$	$T_{z_{RW2}}$	30.	$T_{x_{RW1}}$	$\omega_{actual_{RW3}}$	55.	$T_{y_{RW2}}$	q_{1error}
6.	$T_{x_{RW1}}$	$T_{x_{RW3}}$	31.	$T_{x_{RW1}}$	$\omega_{actual_{RW4}}$	56.	$T_{y_{RW2}}$	q_{2error}
7.	$T_{x_{RW1}}$	$T_{y_{RW3}}$	32.	$T_{x_{RW1}}$	$p_{s/c}$	57.	$T_{y_{RW2}}$	q_{3error}
8.	$T_{x_{RW1}}$	$T_{z_{RW3}}$	33.	$T_{x_{RW1}}$	$q_{s/c}$	58.	$T_{y_{RW2}}$	$\omega_{actual_{RW1}}$
9.	$T_{x_{RW1}}$	$T_{x_{RW4}}$	34.	$T_{x_{RW1}}$	$r_{s/c}$	59.	$T_{y_{RW2}}$	$\omega_{actual_{RW2}}$
10.	$T_{x_{RW1}}$	$T_{y_{RW4}}$	35.	$T_{y_{RW2}}$	$T_{z_{RW2}}$	60.	$T_{y_{RW2}}$	$\omega_{actual_{RW3}}$
11.	$T_{x_{RW1}}$	$T_{z_{RW4}}$	36.	$T_{y_{RW2}}$	$T_{x_{RW3}}$	61.	$T_{y_{RW2}}$	$\omega_{actual_{RW4}}$
12.	$T_{x_{RW1}}$	$\omega_{desired_{RW1}}$	37.	$T_{y_{RW2}}$	$T_{y_{RW3}}$	62.	$T_{y_{RW2}}$	$p_{s/c}$
13.	$T_{x_{RW1}}$	$\omega_{desired_{RW2}}$	38.	$T_{y_{RW2}}$	$T_{z_{RW3}}$	63.	$T_{y_{RW2}}$	$q_{s/c}$
14.	$T_{x_{RW1}}$	$\omega_{desired_{RW3}}$	39.	$T_{y_{RW2}}$	$T_{x_{RW4}}$	64.	$T_{y_{RW2}}$	$r_{s/c}$
15.	$T_{x_{RW1}}$	$\omega_{desired_{RW4}}$	40.	$T_{y_{RW2}}$	$T_{y_{RW4}}$	65.	q_{1error}	q_{2error}
16.	$T_{x_{RW1}}$	$i_{error_{RW1}}$	41.	$T_{y_{RW2}}$	$T_{z_{RW4}}$	66.	q_{1error}	q_{3error}
17.	$T_{x_{RW1}}$	$i_{error_{RW2}}$	42.	$T_{y_{RW2}}$	$\omega_{desired_{RW1}}$	67.	q_{1error}	$\omega_{actual_{RW1}}$
18.	$T_{x_{RW1}}$	$i_{error_{RW3}}$	43.	$T_{y_{RW2}}$	$\omega_{desired_{RW2}}$	68.	q_{1error}	$\omega_{actual_{RW2}}$
19.	$T_{x_{RW1}}$	$i_{error_{RW4}}$	44.	$T_{y_{RW2}}$	$\omega_{desired_{RW3}}$	69.	q_{1error}	$\omega_{actual_{RW3}}$
20.	$T_{x_{RW1}}$	$q_{0actual}$	45.	$T_{y_{RW2}}$	$\omega_{desired_{RW4}}$	70.	q_{1error}	$\omega_{actual_{RW4}}$
21.	$T_{x_{RW1}}$	$q_{1actual}$	46.	$T_{y_{RW2}}$	$i_{error_{RW1}}$	71.	q_{1error}	$p_{s/c}$
22.	$T_{x_{RW1}}$	$q_{2actual}$	47.	$T_{y_{RW2}}$	$i_{error_{RW2}}$	72.	q_{1error}	$q_{s/c}$
23.	$T_{x_{RW1}}$	$q_{3actual}$	48.	$T_{y_{RW2}}$	$i_{error_{RW3}}$	73.	q_{1error}	$r_{s/c}$
24.	$T_{x_{RW1}}$	q_{0error}	49.	$T_{y_{RW2}}$	$i_{error_{RW4}}$			
25.	$T_{x_{RW1}}$	q_{1error}	50.	$T_{y_{RW2}}$	$q_{0actual}$			

Once the HMS Selves are trained, careful consideration should be given to false alarms (FA).

Flight data from SC#3 trajectory was used to validate that nominal data not used as part of the training process is correctly classified as nominal. False alarms were calculated using Equation (2.2). Figure 5.1 illustrates in green a sample of the validation process using Feature-10 and Feature-0 projection for which FA = 4.58% were achieved.



FP = 2609
TN = 54247
FA = 4.58%

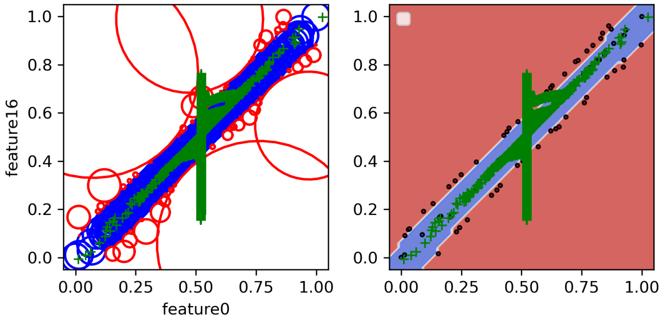
Figure 5.1 Validation: Feature 0 sc_2 vs Feature 18 sc_2

5.2 Local HMS Detection Capabilities

The projections listed in Table 5.2 were trained with the V-detector and AISO-SVM algorithms and tested againts the failure cases described in Subsection 4.3.2. When testing failure cases, Selves with FA lower than 10% were pre-selected as potential projections.

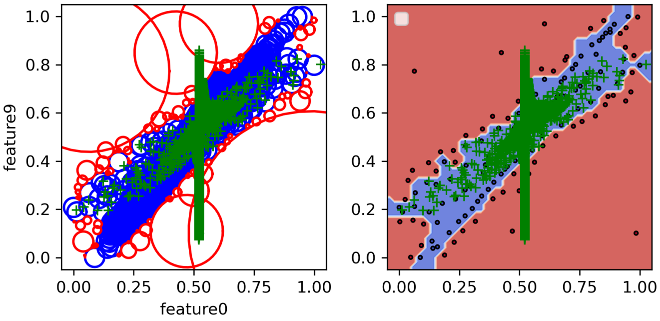
Failure Case I:

Figures 5.2 to 5.5 shows some samples of relevant selves with an acceptable DR and low FA when Failure Case I is injected. Even though the DR are not as high, the low FA allow to infer that off-nominal behaviour is detected.



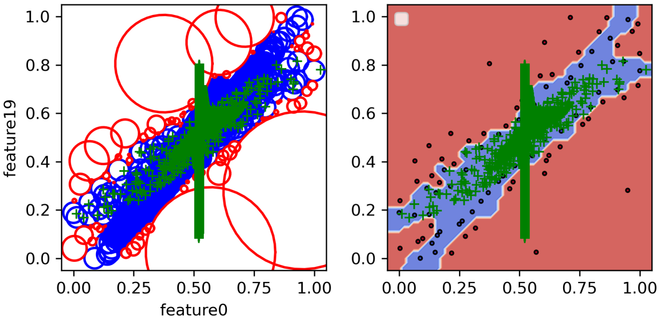
TP = 13786
FN = 43070
FP = 0
TN = 56856
FA = 0%
DR = 24.25%

Figure 5.2 Fail Case I: Feature 0 sc_2 vs Feature 16 sc_2



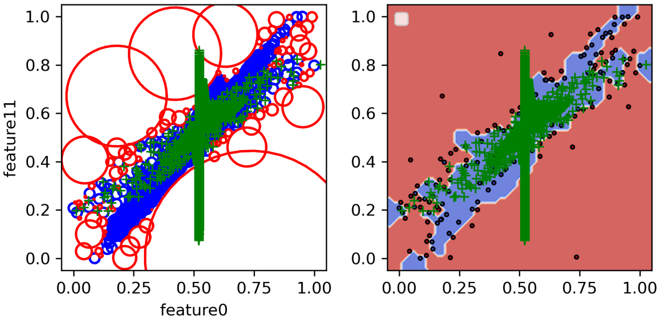
TP = 8650
FN = 48206
FP = 41
TN = 56815
FA = 0.07%
DR = 15.21%

Figure 5.3 Fail Case I: Feature 0 sc_2 vs Feature 9 sc_2



TP = 8812
FN = 48044
FP = 16
TN = 56840
FA = 0.03%
DR = 15.50%

Figure 5.4 Fail Case I: Feature 0 sc_2 vs Feature 19 sc_2

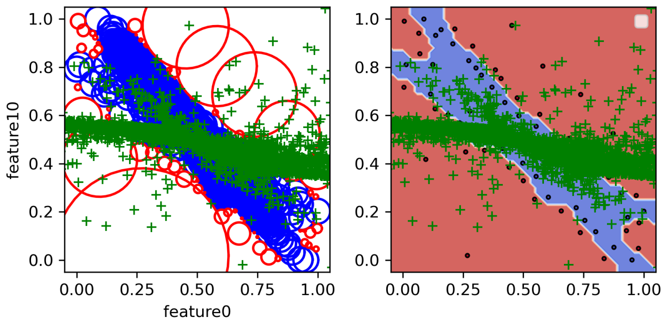


TP = 9482
FN = 47374
FP = 623
TN = 56846
FA = 1.09 %
DR = 16.68%

Figure 5.5 Fail Case I: Feature 0 sc_2 vs Feature 11 sc_2

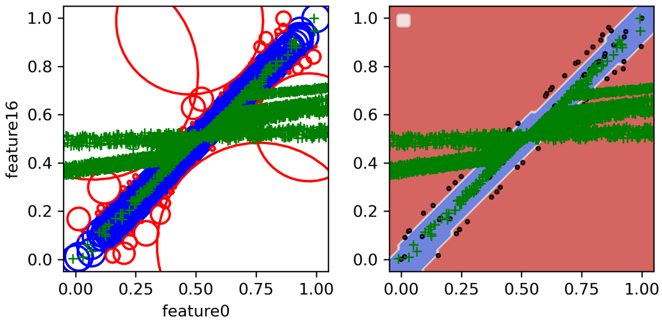
Failure Case II:

Figures 5.6 to 5.9 illustrate relevant results with high DR and low FA from selves able to detect abnormal conditions when Failure Case II occurs. As can be noted, the selected features are good candidates for detecting torque saturation in RWs. It is worth discussing the interesting fact revealed by self ([0-16]) in Figure 5.2 and Figure 5.7, where the same feature combination is able to detect Failure Case I and Failure Case II, but the pattern of the failure data (green) is different. This provides evidence for a failure identification logic, considering that different types of failure generate different patterns within a generated self more sensitive to specific failures.



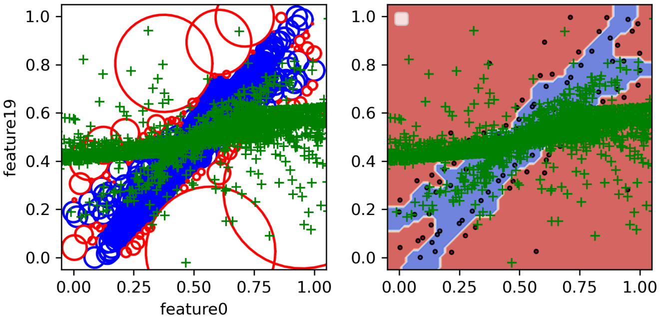
$TP = 53288$
$FN = 3568$
$FP = 10$
$TN = 56846$
$FA = 0.017\ \%$
$DR = 93.72\%$

Figure 5.6 Fail Case II: Feature 0 sc_2 vs Feature 10 sc_2



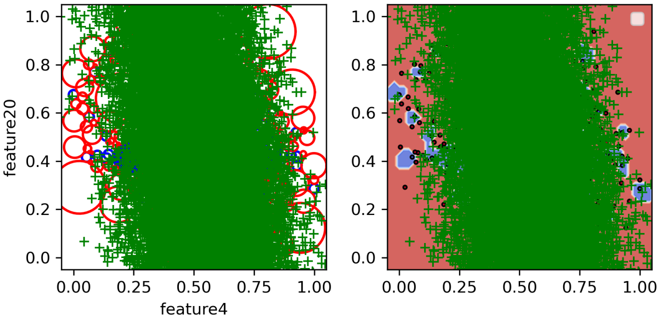
$TP = 54336$
$FN = 2520$
$FP = 0$
$TN = 56856$
$FA = 0\%$
$DR = 95.56\%$

Figure 5.7 Fail Case II: Feature 0 sc_2 vs Feature 16 sc_2



TP = 53321
FN = 3535
FP = 16
TN = 56840
FA = 0.03 %
DR = 93.78%

Figure 5.8 Fail Case II: Feature 0 sc_2 vs Feature 19 sc_2

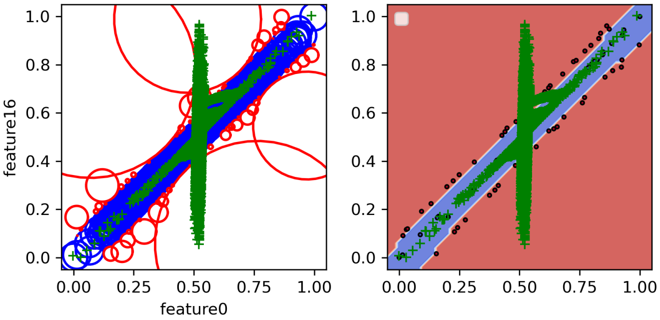


TP = 35533
FN = 21323
FP = 112
TN = 56744
FA = 0.19 %
DR = 62.50%

Figure 5.9 Fail Case II: Feature 4 sc_2 vs Feature 20 sc_3

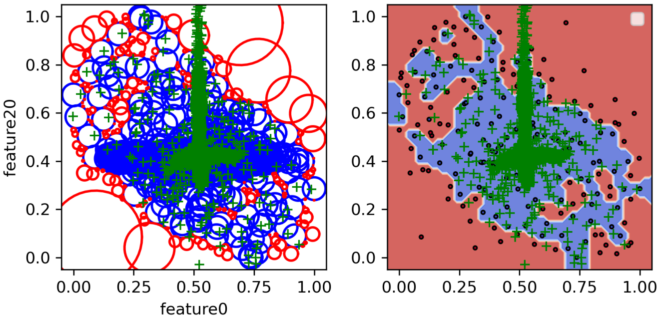
Failure Case III:

Figures 5.10 to 5.13 illustrate some samples of selves with high DR and low FA, and able to detect abnormal conditions when Failure Case III is present. As can be noted, the selected features are good candidates for detecting when two RWs operate in abnormal conditions.



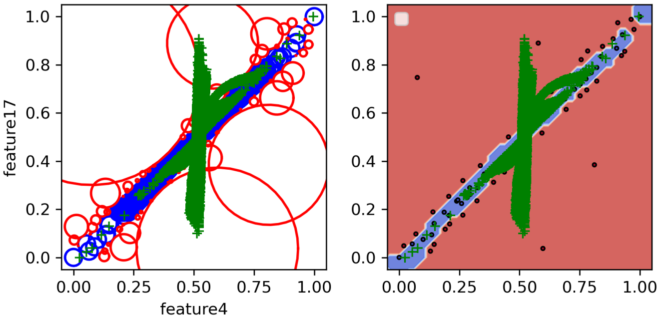
TP = 26104
FN = 30752
FP = 0
TN = 56856
FA = 0%
DR = 45.91%

Figure 5.10 Fail Case III: Feature 0 sc_2 vs Feature 16 sc_2



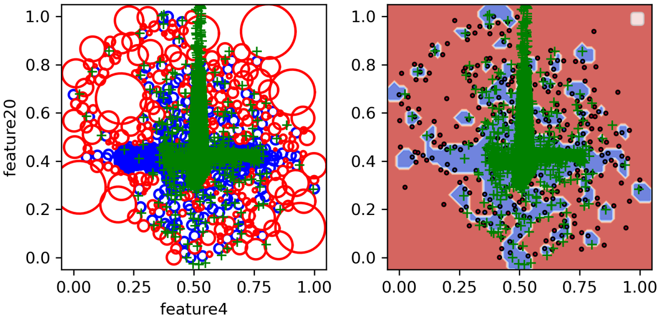
TP = 52319
FN = 4537
FP = 55
TN = 56801
FA = 0.09 %
DR = 92.02%

Figure 5.11 Fail Case III: Feature 0 sc_2 vs Feature 20 sc_2



TP = 35780
FN = 21076
FP = 0
TN = 56856
FA = 0 %
DR = 62.93%

Figure 5.12 Fail Case III: Feature 4 sc_2 vs Feature 17 sc_2



TP = 40459
FN = 16397
FP = 112
TN = 56744
FA = 0.2 %
DR = 71.16%

Figure 5.13 Fail Case III: Feature 4 sc_2 vs Feature 20 sc_2

5.3 Global HMS Detection Capabilities

Off-nominal conditions on a single system within a network can affect mission performance of the complete network. Implementing a global HMS is beneficial for neighboring systems to notice abnormal conditions and trigger collaborative actions to maintain mission objectives. Recalling that from an individual system is possible to built a local HMS based on its own feature states, similarly, from a network of systems is possible to built a global HMS based on a cross-combination of feature states by treating the network as a single dynamic system. This concept is illustrated in Figure 5.14 where the monitoring of the network states allows to detect misbehaving agents within the network.

In the following detection examples, the faulty agent is selected to be SC#2, and V-detector and AISO-SVM selves are generated from features taken from a cross-combination of futures from SC#2 and SC#3, and SC#3 and SC#4. It is therefore expected that selves that include features from the faulty agent (SC#2) would be more prone to detect such abnormal behaviours in SC#2. A set of crossed combination features ([4-17]), ([0-16]) and ([25-30]) was selected based on the performance shown in the local HMS to detect failures with high DR and low FA.

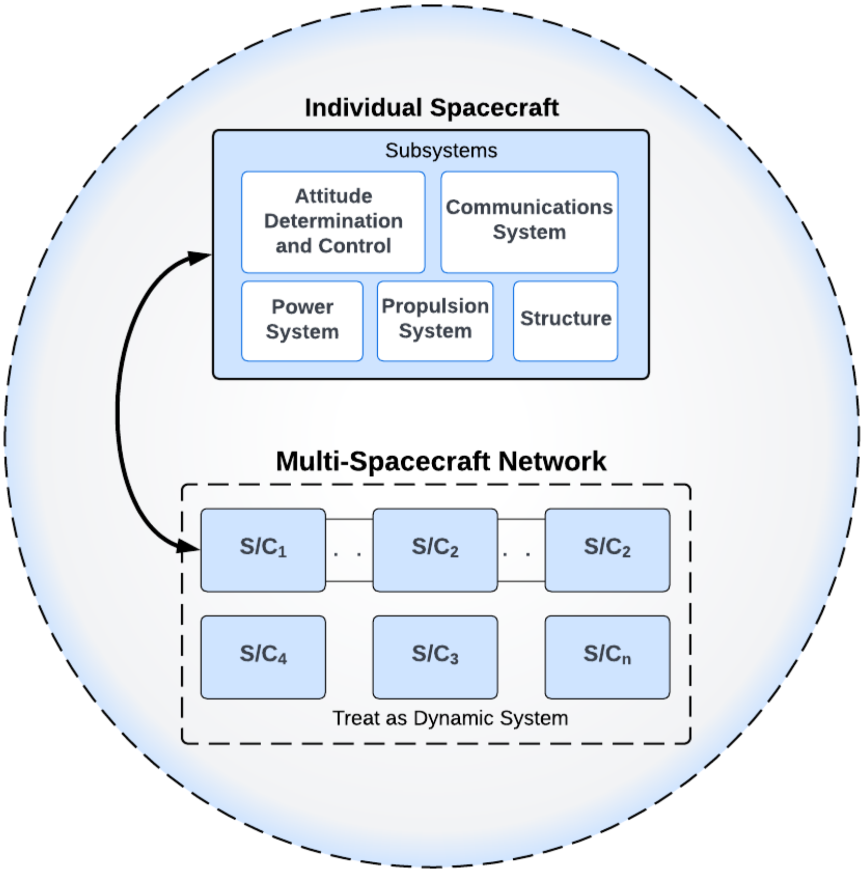
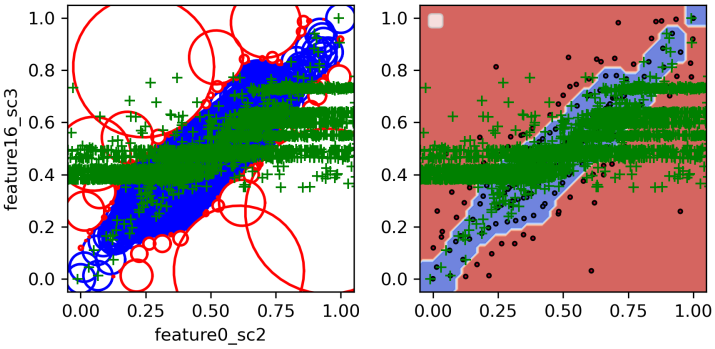


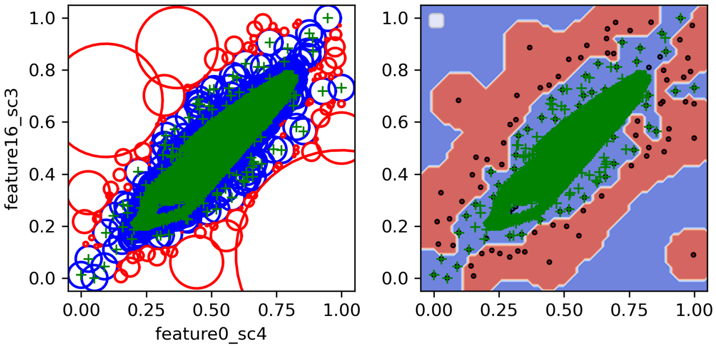
Figure 5.14 Representation of Global HMS for detecting abnormal conditions in a single spacecraft

Figures 5.15 to 5.20 show the results for selected cross-feature projections against Failures Cases II and III (in green). These detection results are representative for detecting abnormal conditions in a network and identifying the possible faulty agent. As can be analyzed from Figures 5.15, 5.17 and 5.19, when SC#2 is misbehaving because a failure, only selves that contain features of that particular faulty agent will be activated. This information can be used to isolate the agent and it can followed by a local self analysis to determine the cause.



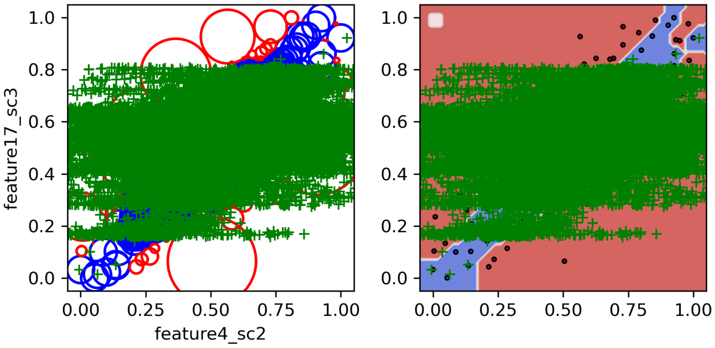
TP = 53059
FN = 3791
DR = 93.32%

Figure 5.15 Fail Case II: Feature 0 sc_2 vs Feature 16 sc_3



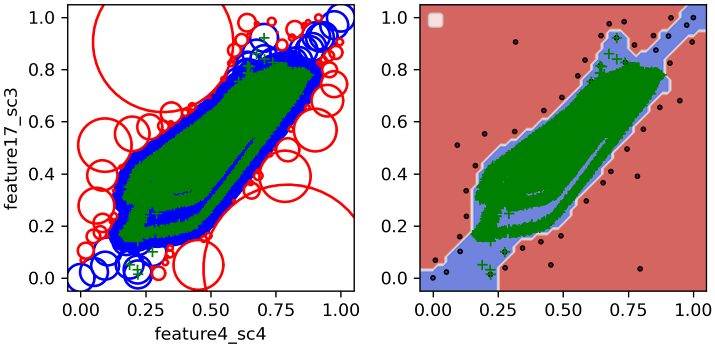
TP = 0
FN = 56856
DR = 0.0%

Figure 5.16 Global Self, Fail Case II: Feature 0 sc_4 vs Feature 16 sc_3



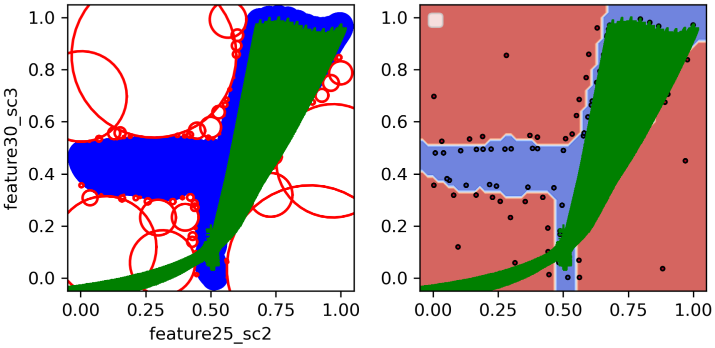
TP = 9061
FN = 47795
DR = 15.93%

Figure 5.17 Fail Case II: Feature 4 sc_2 vs Feature 17 sc_2



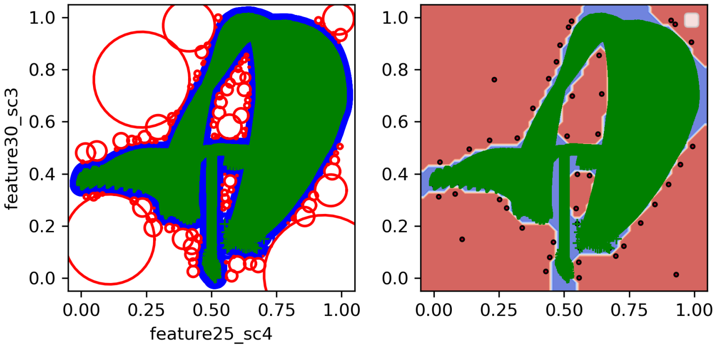
TP = 0
FN = 56856
DR = 0.0%

Figure 5.18 Global Self, Fail Case II: Feature 4 sc_4 vs Feature 17 sc_3



TP = 38654
FN = 18202
DR = 67.98%

Figure 5.19 Fail Case III: Feature 25 sc_2 vs Feature 30 sc_3



TP = 0
FN = 56856
DR = 0.0%

Figure 5.20 Global Self, Fail Case III: Feature 25 sc_4 vs Feature 30 sc_3

These detection results are representative for detecting abnormal conditions in a network and identifying the possible faulty agent. As can be analyzed from Figures 5.15, 5.17, and 5.19, only selves that contain features from the faulty agent (SC#2) are prone to detect abnormal conditions. From these plots it can be inferred that there is some malfunctioning, even though there exist uncertainty about which spacecraft is the faulty system. The global HMS can follow a local HMS logic to confirm the faulty agent by verifying outputs from the local selves. On the other hand, when flight data of non-faulty systems is plotted against selves that do not include features from the faulty agent, there is no detection, as these systems remain in nominal conditions.

5.4 Attitude Controllers

An adaptive attitude controller based on DRL and optimal control concepts that uses an actor-critic-adverse architecture is strategically implemented in this study, looking forward to the future goals of this research. The DRL controller has online adaptive capabilities to learn the optimal control solution for the worst failure scenario by evaluating a cost function that considers system performance, control actuation and disturbances. Furthermore, other constraints defined by the user can be included in the cost function with the aim of obtaining the optimal control law that accounts for the considered limitations of the system. This approach provides a connection between the HMS output data and the decision making process for modifying the controller or modifying the trajectory optimization algorithms based on the new flight conditions.

To test the capabilities of the DRL controller, a comparison against a NDLI and Pole Placement controller was made for two different failure cases. For these failure cases, the faulty spacecraft was SC#4, while the rest of vehicles in the network remain in nominal conditions. A description of the tested cases is presented below:

- **Case I**: Progressive decrease in RW2 of SC#4 and a sinusoidal disturbance of magnitude $d = sin(\pi t)$ is added in the control channels M_x, M_y, M_z, starting at t = 0 s.
- **Case II**: RW1 in SC#4 operating at 20% of its capacity, progressive decrease in RW2 of SC#4, and a sinusoidal disturbance $d = 100sin(\pi t)$ is added in the control channels M_x, M_y, M_z, starting at t = 0 s.

5.4.1 Design Parameters

Using the spacecraft linear model presented in Section 3.2, the DRL controller based in optimal control concepts was designed. Also, the Pole Placement controller was designed based on the referenced linear model. On the other hand, the NLDI controller was designed based on the nonlinear equations presented in Equation 3.10. A description of the parameters for each controller are:

- **DRL controller**

  The Q and R matrices along with the optimal gain K^* and S matrix from the solution to the ARE, are shown in Equations 5.2, 5.3, 5.4, and 5.5.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

$$R = \begin{bmatrix} 0.7 & 0 & 0 \\ 0 & 0.7 & 0 \\ 0 & 0 & 0.7 \end{bmatrix} \tag{5.3}$$

$$K^* = \begin{bmatrix} 1.2382 & 0 & 0 & 1.1952 & 0 & 0 \\ 0 & 1.2507 & 0 & 0 & 1.1952 & 0 \\ 0 & 0 & 1.2169 & 0 & 0 & 1.1952 \end{bmatrix} \tag{5.4}$$

$$S = \begin{bmatrix} 0.0758 & 0 & 0 & 0.0731 & 0 & 0 \\ 0 & 0.0995 & 0 & 0 & 0.0950 & 0 \\ 0 & 0 & 0.0372 & 0 & 0 & 0.0366 \\ 0.0731 & 0 & 0 & 2.0718 & 0 & 0 \\ 0 & 0.0950 & 0 & 0 & 2.0929 & 0 \\ 0 & 0 & 0.0366 & 0 & 0 & 2.0362 \end{bmatrix} \tag{5.5}$$

Furthermore, the tuning parameters for the adaptive weight laws are:

$\gamma = 1$, $\alpha_o = 1$, $\alpha_c = 10$, $\alpha_a = 1$

The quadratic form of NN was chosen for both *actor* and *critic* based on x_1, x_2, x_3, x_4, x_5 and x_6, as follows:

$$\Phi_c = \begin{bmatrix} x_1^2 & x_1x_2 & x_2^2 & x_1x_3 & x_2x_3 & x_3^2 & x_1x_4 & x_2x_4 & x_3x_4 & x_4^2 & x_1x_5 \\ & x_2x_5 & x_3x_5 & x_4x_5 & x_5^2 & x_1x_6 & x_2x_6 & x_3x_6 & x_4x_6 & x_5x_6 & x_6^2 \end{bmatrix}$$

where their respective gradients are:

$$\nabla\Phi_c = \begin{bmatrix} 2x_1 & 0 & 0 & 0 & 0 & 0 \\ x_2 & x_1 & 0 & 0 & 0 & 0 \\ 0 & 2x_2 & 0 & 0 & 0 & 0 \\ x_3 & 0 & x_1 & 0 & 0 & 0 \\ 0 & x_3 & x_2 & 0 & 0 & 0 \\ 0 & 0 & 2x_3 & 0 & 0 & 0 \\ x_4 & 0 & 0 & x_1 & 0 & 0 \\ 0 & x_4 & 0 & x_2 & 0 & 0 \\ 0 & 0 & x_4 & x_3 & 0 & 0 \\ 0 & 0 & 0 & 2x_4 & 0 & 0 \\ x_5 & 0 & 0 & 0 & x_1 & 0 \\ 0 & x_5 & 0 & 0 & x_2 & 0 \\ 0 & 0 & x_5 & 0 & x_3 & 0 \\ 0 & 0 & 0 & x_5 & x_4 & 0 \\ 0 & 0 & 0 & 0 & 2x_5 & 0 \\ x_6 & 0 & 0 & 0 & 0 & x_1 \\ 0 & x_6 & 0 & 0 & 0 & x_2 \\ 0 & 0 & x_6 & 0 & 0 & x_3 \\ 0 & 0 & 0 & x_6 & 0 & x_4 \\ 0 & 0 & 0 & 0 & x_6 & x_5 \\ 0 & 0 & 0 & 0 & 0 & 2x_6 \end{bmatrix}$$

- **Pole Placement**

The desired poles were chosen arbitrarily to design the Pole Placement gain K^*_{PP} as follows:

$\lambda_1 = -2,\ \lambda_2 = -3,\ \lambda_3 = -1,\ \lambda_4 = -4,\ \lambda_5 = -5,\ \lambda_6 = -2$

where the gain K_{PP}^* is:

$$K_{PP}^* = \begin{bmatrix} 0.6118 & 0 & 0 & 1.7480 & 0 & 0 \\ 0 & 0.4544 & 0 & 0 & 0.6816 & 0 \\ 0 & 0 & 0.2622 & 0 & 0 & 0.6992 \end{bmatrix} \tag{5.6}$$

- **Nonlinear Dynamic Inversion**

The gains used in the pseudo controller from Equation 3.27 were chosen as:

$$K_q = \begin{bmatrix} 2.0934 & 0 & 0 \\ 0 & 2.0934 & 0 \\ 0 & 0 & 2.0934 \end{bmatrix} \qquad K_\Omega = \begin{bmatrix} 2.8571 & 0 & 0 \\ 0 & 2.8571 & 0 \\ 0 & 0 & 2.8571 \end{bmatrix} \tag{5.7}$$

5.4.2 Controllers Performance Comparison

Figure 5.21 presents the attitude tracking performance when the DRL, NLDI and Pole Placement controllers are implemented when Failure Case I occurs in SC#4.

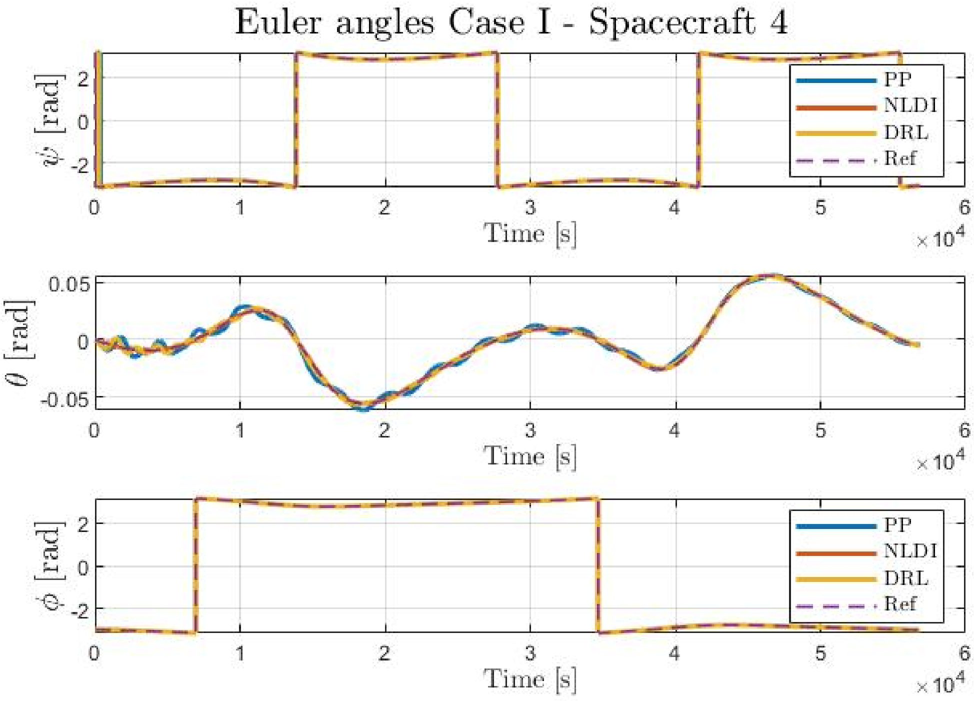


Figure 5.21 Attitude tracking with DRL based controller - Case I

As can be noted in Figure 5.21, all three controllers are able to cope with the tracking requirements with some oscillations, mainly in the pitch state. The controller that tracks better is the NLDI controller, followed by the DRL controller with some oscillations at the beginning and finally the Pole Placement controller with less accuracy.

Figure 5.22 shows the control effort of the various controllers compared to the control effort in nominal conditions, when Failure Case I occurs. From these results, it can be concluded that the NLDI controller requires the greatest control effort to provide the attitude tracking performance, followed by the DRL and the Pole Placement controller. Hence, the DRL controller demonstrates a good balance between control effort and attitude tracking performance.

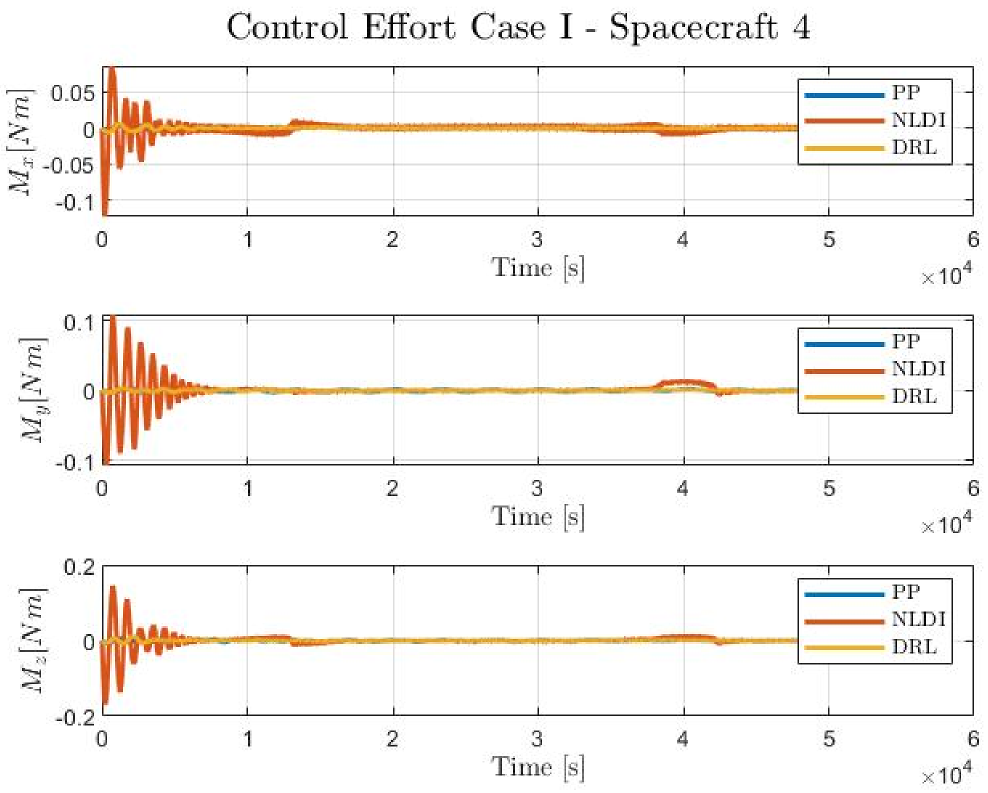


Figure 5.22 Attitude tracking with DRL based controller - Case I

Furthermore, Figures 5.23 and 5.24 show the adverse weight adaptation and adverse function approximation, respectively. The adverse weights are initialized at zero and their behaviour converges to a constant value, which provides evidence about the adaptation of the NNs to approximate the optimal solution of the control law while accounting for the disturbance w in the system.

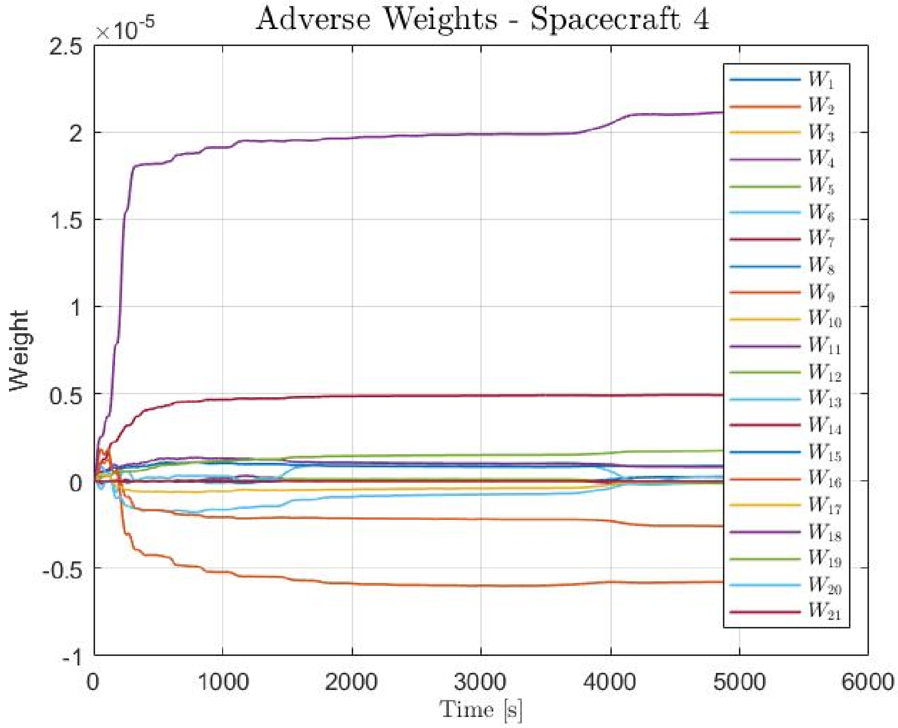


Figure 5.23 Adverse adaptive weights - Case I

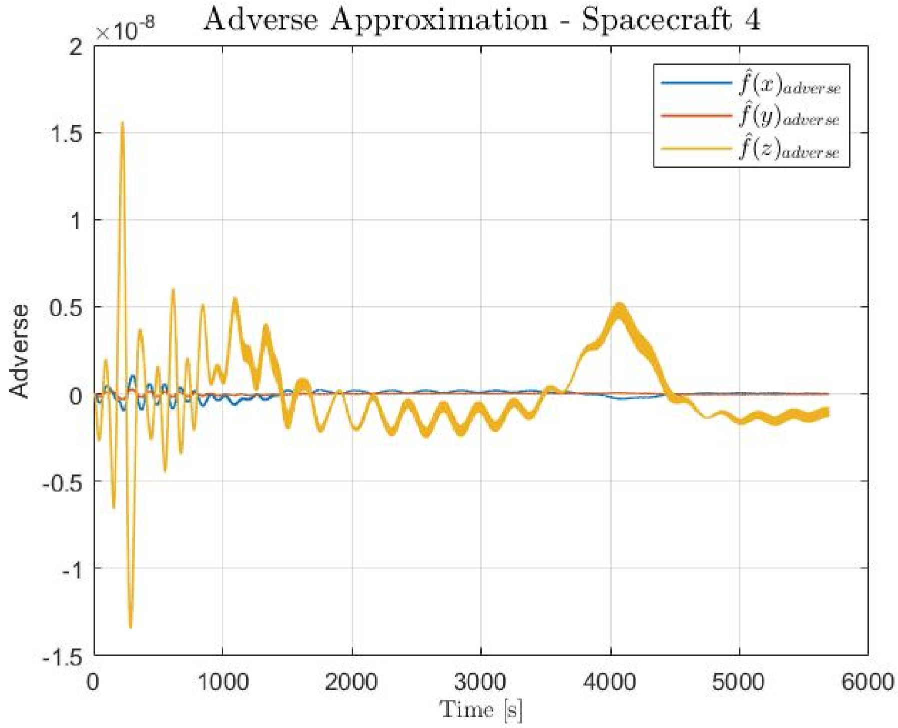


Figure 5.24 Adverse function approximation - Case I

Similarly to Failure Case I above, Figure 5.25 presents the attitude tracking performance when the DRL, NLDI and Pole Placement controllers are implemented when Failure Case II occurs in SC#4.

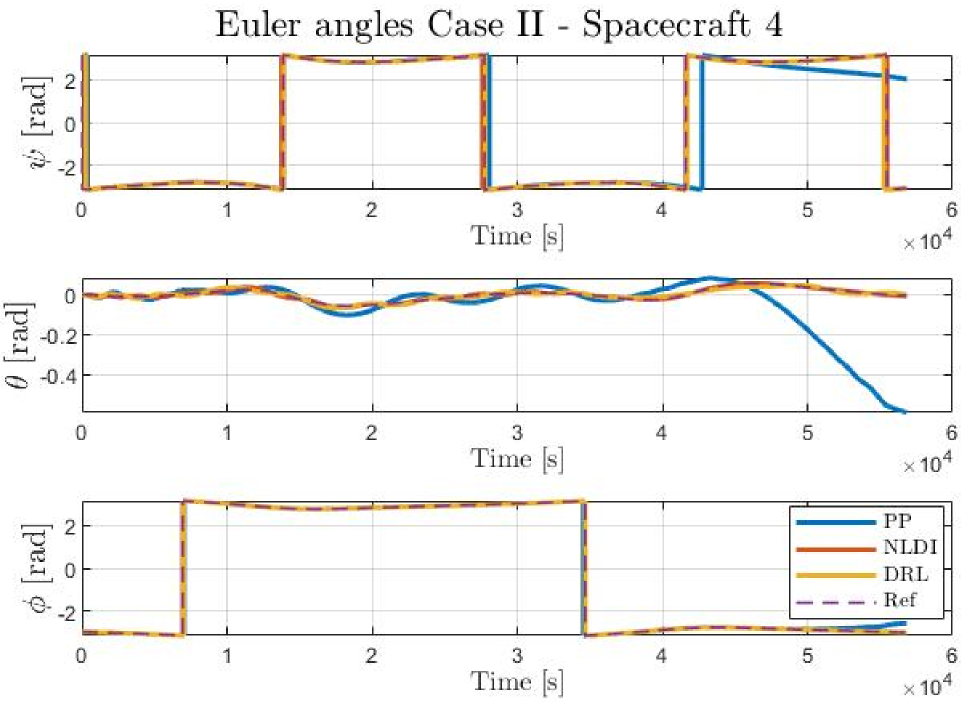


Figure 5.25 Attitude tracking with DRL based controller - Case II

As can be noted, Failure Case II is more aggressive, which makes it harder for the controllers to cope with the attitude tracking requirements. The DRL controller tries to follow the reference attitude with good approximation, while the NLDI controller performs better with less oscillations. On the other hand, the Pole Placement controller is not able to comply with the attitude requirements. However, the Pole Placement controller could be improved by setting design requirements.

Figure 5.26 shows the control effort of the various controllers when Failure Case II occurs compared to the control effort in nominal conditions. From these results, it can be concluded that the Pole Placement controller experiences the greatest control effort, followed by NLDI and DRL. These results provide evidence of the robustness of the DRL controller in the presence of disturbances and demonstrates a good balance between control effort and attitude tracking performance.

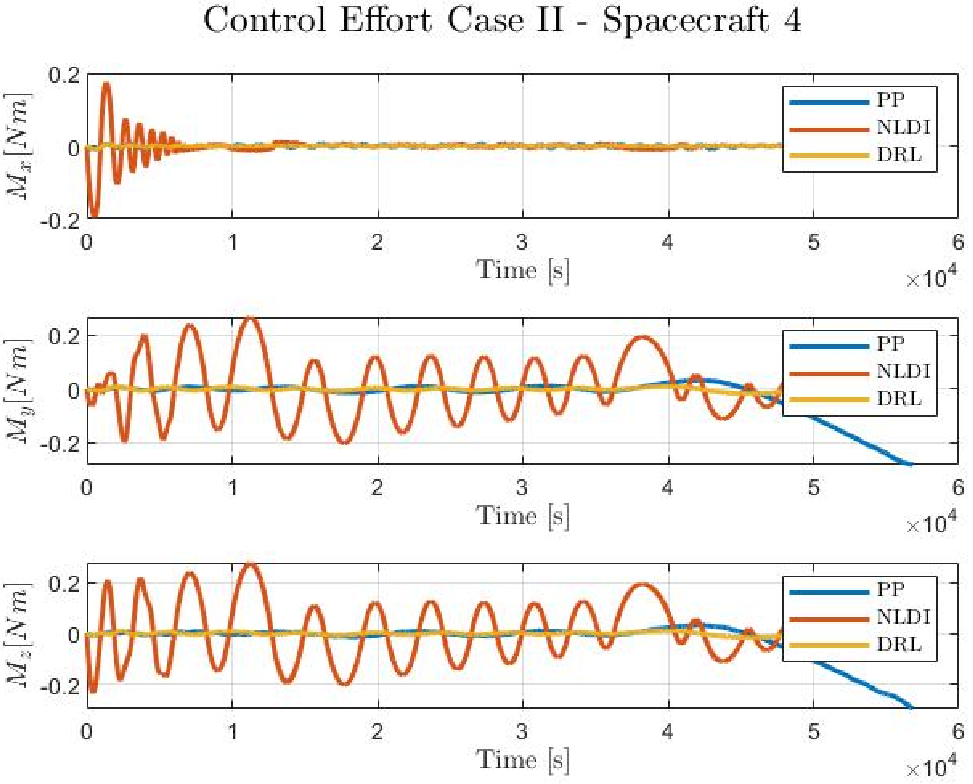


Figure 5.26 Attitude tracking with DRL based controller - Case II

Finally, Figures 5.27 and 5.28 show the adverse weight adaptation and adverse function approximation, respectively.

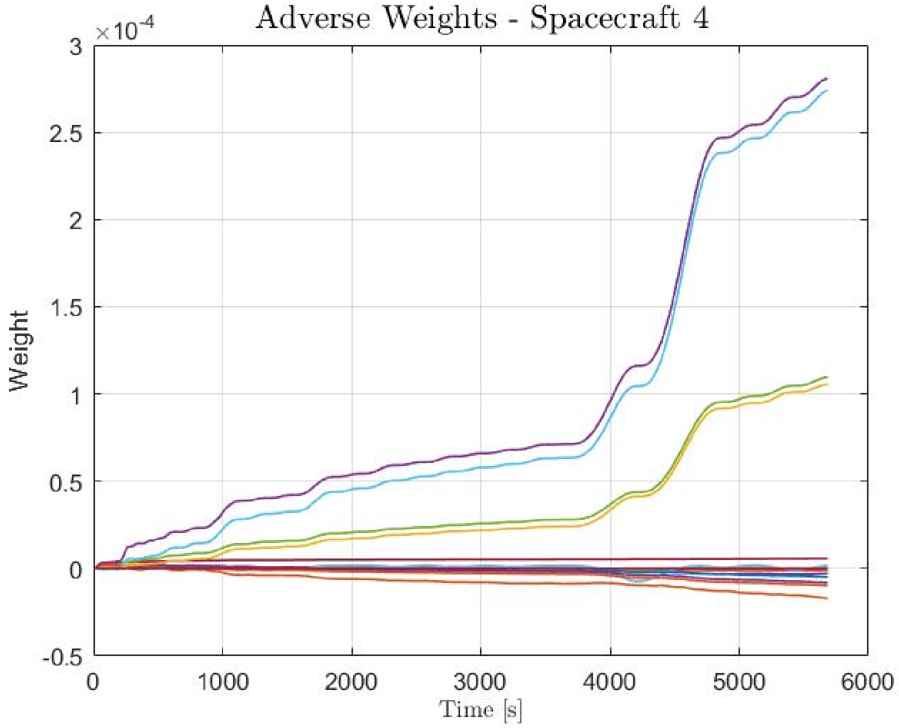


Figure 5.27 Adverse adaptive weights - Case II

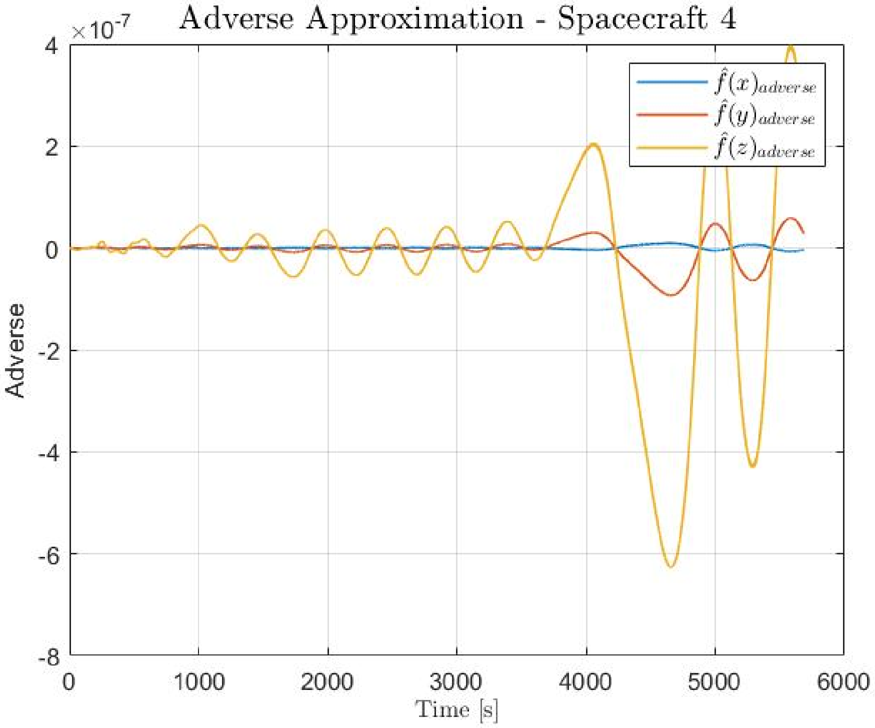


Figure 5.28 Adverse function approximation - Case II

6 Conclusions and Future Work

On-orbit and deep space missions involving Distributed Spacecraft Systems have gained significant interest for their extensive range of benefits and possibilities; as a consequence, it is important to guarantee flight safety and mission performance. The importance of gathering real-time data from a health management system and using it for decision-making processes is highlighted in this research, which could improve the robustness and resiliency of aerospace systems. A Health Monitoring System based on the Variable Detector and the AISO- SVM algorithm has been implemented to obtain system status. To assess the capabilities of the proposed health-management framework, a simulation environment for a fleet of spacecraft performing a low-Earth orbit inspection was developed. Three types of failures in the attitude control system that uses Reaction Wheels were also modeled. Results presented in this thesis show the potential capabilities of the proposed architecture for the detection of subtle changes or failures during multi-spacecraft operations. Furthermore, a Deep Reinforcement Learning-based controller was compared against an NDLI and Pole Placement controller for two different failure scenarios. The Deep Reinforcement Learning controller demonstrated a good balance between control effort and attitude-tracking performance in the presence of failures and disturbances, which makes it suitable for future research goals, such as executing adjustments in the controller accounting for information of the Health Monitoring System.

Future research should consider investigating a mathematical way to include the Health Monitoring System information in the controller and trajectory optimization algorithms to demonstrate the potential for resource allocation. Additionally, it is of interest to develop new adaptation laws for the Deep Reinforcement Learning based controller using Lyapunov analysis in conjunction with other mathematical tools to guarantee the boundedness of the states and adaptive gains. Further work implementing the Health Monitoring System on networks with communication topologies is a matter of interest, as faulty agents might affect neighboring agents' performance.

REFERENCES

[1] NASA, "The Afternoon Constellation," , 2022. URL `https://atrain.nasa.gov/`.

[2] Yost, B., Weston, S., Benavides, G., Krage, F., Hines, J., Mauro, S., Etchey, S., O'Neill, K., and Braun, B., "Small Spacecraft Technology State of the Art," Tech. rep., 2021.

[3] Rossi, F., Bandyopadhyay, S., Mote, M., de la Croix, J.-P., and Rahmani, A., "Communication-Aware Orbit Design for Small Spacecraft Swarms Around Small Bodies," *Journal of Guidance, Control, and Dynamics*, 2022, pp. 1–15. https://doi.org/10.2514/1.G006515, URL https://arc.aiaa.org/doi/10.2514/1.G006515.

[4] Nakka, Y. K., Hönig, W., Choi, C., Harvard, A., Rahmani, A., and Chung, S.-J., "Information-Based Guidance and Control Architecture for Multi-Spacecraft On-Orbit Inspection," *Journal of Guidance, Control, and Dynamics*, 2022, pp. 1–18. https://doi.org/10.2514/1.G006278, URL https://arc.aiaa.org/doi/10.2514/1.G006278.

[5] Morgan, D., and Chung, S.-J., "Swarm-Keeping Strategies for Spacecraft Under J2 and Atmospheric Drag Perturbations," 2012.

[6] Foust, R. C., Lupu, E. S., Nakka, Y. K., Chung, S.-J., and Hadaegh, F. Y., "Autonomous in-orbit satellite assembly from a modular heterogeneous swarm," *Acta Astronautica*, Vol. 169, 2020, pp. 191–205. https://doi.org/10.1016/j.actaastro.2020.01.006, URL https://linkinghub.elsevier.com/retrieve/pii/S0094576520300060.

[7] Jacklin, S. A., "Small-Satellite Mission Failure Rates," 2019, p. 46.

[8] Tudoroiu, N., and Khorasani, K., "Satellite Fault Diagnosis using a Bank of Interacting Kalman Filters," Vol. 43, No. 4, 2007, p. 17.

[9] Perhinschi, M. G., and Moncayo, H., "Artificial Immune System for Comprehensive and Integrated Aircraft Abnormal Conditions Management," *Advances in Computational Intelligence and Autonomy for Aerospace Systems*, Progress in Astronautics

and Aeronautics, Vol. Volume 254, American Institute of Aeronautics and Astronautics, Inc., 2018, pp. 147–218. https://doi.org/10.2514/5.9781624104794.0147.0218, URL https://arc.aiaa.org/doi/10.2514/5.9781624104794.0147.0218.

[10] Azzawi, D. A., Moncayo, H., Perhinschi, M. G., Perez, A., and Togayev, A., "Comparison of immunity-based schemes for aircraft failure detection and identification," *Engineering Applications of Artificial Intelligence*, Vol. 52, 2016, pp. 181–193. https://doi.org/10.1016/j.engappai.2016.02.017, URL https://www.sciencedirect.com/science/article/pii/S0952197616300379.

[11] Moncayo, H., Perhinschi, M. G., and Davis, J., "Artificial-Immune-System-Based Aircraft Failure Evaluation over Extended Flight Envelope," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, 2011, pp. 989–1001. https://doi.org/10.2514/1.52748, URL https://arc.aiaa.org/doi/10.2514/1.52748.

[12] Garcia, D. F., Perez, A. E., Moncayo, H., Rivera, K., Betancur, Y., DuPuis, M., and Mueller, R. P., "Spacecraft Heath Monitoring Using a Biomimetic Fault Diagnosis Scheme," *Journal of Aerospace Information Systems*, Vol. 15, No. 7, 2018, pp. 396–413. https://doi.org/10.2514/1.I010612, URL https://arc.aiaa.org/doi/10.2514/1.I010612.

[13] Coulter, N. L., and Moncayo, H., "Data-Driven Approaches for Spacecraft Fault Detection Systems," , 2022.

[14] Aydin, I., Karakose, M., and Akin, E., "A multi-objective artificial immune algorithm for parameter optimization in support vector machine," *Applied Soft Computing*, Vol. 11, No. 1, 2011, pp. 120–129. https://doi.org/10.1016/j.asoc.2009.11.003, URL https://linkinghub.elsevier.com/retrieve/pii/S1568494609002166.

[15] Brunton, S. L., and Kutz, J. N., *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, 1st ed., Cambridge University Press,

2019. https://doi.org/10.1017/9781108380690, URL https://www.cambridge.org/core/product/identifier/9781108380690/type/book.

[16] Miller, W., Sutton, R., and Werbos, P., "A Menu of Designs for Reinforcement Learning Over Time," *Neural Networks for Control*, 1995, pp. 67 – 95.

[17] Vamvoudakis, K. G., and Lewis, F. L., "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, Vol. 46, No. 5, 2010, pp. 878–888. https://doi.org/10.1016/j.automatica.2010.02.018, URL https://linkinghub.elsevier.com/retrieve/pii/S0005109810000907.

[18] Stephens, G., Winker, D., Pelon, J., Trepte, C., Vane, D., Yuhas, C., L'ecuyer, T., and Lebsock, M., "CloudSat and CALIPSO within the A-Train: Ten years of actively observing the Earth system," Tech. rep., Bulletin of the American Meteorological Society, 2018.

[19] Flechtner, F., Morton, P., Watkins, M., and Webb, F., "Status of the GRACE Follow-On Mission," *Gravity, Geoid and Height Systems*, Vol. 141, edited by U. Marti, Springer International Publishing, Cham, 2014, pp. 117–121. https://doi.org/10.1007/978-3-319-10837-7_15, URL http://link.springer.com/10.1007/978-3-319-10837-7_15, series Title: International Association of Geodesy Symposia.

[20] Schilling, K., Schechner, Y. Y., and Koren, I., "CloudCT – Computed Tomography of Clouds by a Small Satellite Formation," *Proceedings of the 12th IAA symposium on Small Satellites for Earth Observation*, 2019, p. 7.

[21] Schoolcraft, J., Klesh, A. T., and Werne, T., "MarCO: Interplanetary Mission Development On a CubeSat Scale," *SpaceOps 2016 Conference*, American Institute of Aeronautics and Astronautics, Daejeon, Korea, 2016. https://doi.org/10.2514/6.2016-2491, URL https://arc.aiaa.org/doi/10.2514/6.2016-2491.

[22] Kasper, J., Lazio, J., Romero-Wolf, A., Lux, J., and Neilsen, T., "The Sun Radio Interferometer Space Experiment (SunRISE) Mission Concept," *2019 IEEE Aerospace Conference*, IEEE, Big Sky, MT, USA, 2019, pp. 1–11. https://doi.org/10.1109/AERO.2019.8742146, URL https://ieeexplore.ieee.org/document/8742146/.

[23] Quadrelli, M. B., Hodges, R., Vilnrotter, V., Bandyopadhyay, S., Tassi, F., and Bevilacqua, S., "Distributed Swarm Antenna Arrays for Deep Space Applications," *2019 IEEE Aerospace Conference*, 2019, pp. 1–15. https://doi.org/10.1109/AERO.2019.8742019, iSSN: 1095-323X.

[24] Vitug, E., "Cooperative Autonomous Distributed Robotic Exploration (CADRE)," , Feb. 2021. URL http://www.nasa.gov/directorates/spacetech/game_changing_development/projects/CADRE.

[25] Mortari, D., Wilkins, M. P., and Bruccoleri, C., "The Flower Constellations," *The Journal of the Astronautical Sciences*, Vol. 52, No. 1-2, 2004, pp. 107–127. https://doi.org/10.1007/BF03546424, URL https://link.springer.com/10.1007/BF03546424.

[26] Gao, Z., Cecati, C., and Ding, S. X., "A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis With Model-Based and Signal-Based Approaches," *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 6, 2015, pp. 3757–3767. https://doi.org/10.1109/TIE.2015.2417501, URL http://ieeexplore.ieee.org/document/7069265/.

[27] Gao, Z., Cecati, C., and Ding, S. X., "A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part II: Fault Diagnosis With Knowledge-Based and Hybrid/Active Approaches," *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 6, 2015, pp. 3768–3774. https://doi.org/10.1109/TIE.2015.2417501, URL http://ieeexplore.ieee.org/document/7069265/.

[28] Chen, Y., and Lee, M., "Neural networks-based scheme for system failure detection and

diagnosis," *Mathematics and Computers in Simulation*, Vol. 58, No. 2, 2002, pp. 101–109. https://doi.org/10.1016/S0378-4754(01)00330-5, URL https://linkinghub.elsevier.com/retrieve/pii/S0378475401003305.

[29] Rahimi, A., Kumar, K. D., and Alighanbari, H., "Fault estimation of satellite reaction wheels using covariance based adaptive unscented Kalman filter," *Acta astronautica*, Vol. 134, 2017, pp. 159–169. Place: Elmsford Publisher: Elsevier Ltd.

[30] Coulter, N., and Moncayo, H., "Artificial Immune System Optimized Support Vector Machine for Satellite Fault Detection," *AIAA SCITECH 2022 Forum*, American Institute of Aeronautics and Astronautics, San Diego, CA & Virtual, 2022. https://doi.org/10.2514/6.2022-1713, URL https://arc.aiaa.org/doi/10.2514/6.2022-1713.

[31] Lewis, F., Vrabie, D., and Syrmos, V., *Optimal Control*, John Wiley & Sons, Inc, 2012.

[32] Vrabie, D., Vamvoudakis, K., and Lewis, F., "Adaptive optimal controllers based on Generalized Policy Iteration in a continuous-time framework," *2009 17th Mediterranean Conference on Control and Automation*, IEEE, Thessaloniki, Greece, 2009, pp. 1402–1409. https://doi.org/10.1109/MED.2009.5164743, URL http://ieeexplore.ieee.org/document/5164743/.

[33] Bhasin, S., Kamalapurkar, R., Johnson, M., Vamvoudakis, K., Lewis, F., and Dixon, W., "A novel actor–critic–identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, Vol. 49, No. 1, 2013, pp. 82–92. https://doi.org/10.1016/j.automatica.2012.09.019, URL https://linkinghub.elsevier.com/retrieve/pii/S0005109812004827.

[34] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial networks," *Communications of the ACM*, Vol. 63, No. 11, 2020, pp. 139–144. https://doi.org/10.1145/3422622, URL https://dl.acm.org/doi/10.1145/3422622.

[35] Vamvoudakis, K. G., and Lewis, F. L., "Online Gaming: Real Time Solution of Nonlinear Two-Player Zero-Sum Games Using Synchronous Policy Iteration," *Advances in Reinforcement Learning*, edited by A. Mellouk, InTech, 2011. https://doi.org/10.5772/13209.

[36] Morimoto, J., and Doya, K., "Robust Reinforcement Learning," *Neural Computation*, Vol. 17, No. 2, 2005, pp. 335–359. https://doi.org/10.1162/0899766053011528, URL https://direct.mit.edu/neco/article/17/2/335-359/6920.

[37] Fravolini, M. L., and Campa, G., "Design of a Neural Network Adaptive Controller via a Constrained Invariant Ellipsoids Technique," *IEEE Transactions on Neural Networks*, Vol. 22, No. 4, 2011, pp. 627–638. https://doi.org/10.1109/TNN.2011.2111385, URL http://ieeexplore.ieee.org/document/5732704/.

[38] Carrillo, L. R. G., and Vamvoudakis, K. G., "Deep-Learning Tracking for Autonomous Flying Systems Under Adversarial Inputs," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 2, 2020, pp. 1444–1459. https://doi.org/10.1109/TAES.2019.2930017, URL https://ieeexplore.ieee.org/document/8767968/.

[39] Choi, Y.-C., Son, J.-H., and Ahn, H.-S., "Fault detection and isolation for a small CMG-based satellite: A fuzzy Q-learning approach," Vol. 47, 2015, pp. 340–355.

[40] Al-Zyoud, I., and Khorasani, K., "Detection ofactuator faults using a dynamic neural network for the attitude control subsystem of a satellite," *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, Vol. 3, IEEE, Montreal, QC, Canada, 2005, pp. 1746–1751. https://doi.org/10.1109/IJCNN.2005.1556144, URL http://ieeexplore.ieee.org/document/1556144/.

[41] Valasek, J., *Advances in Computational Intelligence and Autonomy for Aerospace Systems*, American Institute of Aeronautics and Astronautics, Inc., Reston ,VA, 2018. https://doi.org/10.2514/4.104794, URL https://arc.aiaa.org/doi/abs/10.2514/4.104794, _eprint: https://arc.aiaa.org/doi/pdf/10.2514/4.104794.

[42] Perhinschi, M., Moncayo, H., and Davis, J., "Integrated Framework for Artificial Immunity-Based Aircraft Failure Detection, Identification, and Evaluation | Journal of Aircraft," 2010. URL https://arc.aiaa.org/doi/10.2514/1.45718.

[43] Dasgupta, D., Krishnakumar, K., Wong, D., and Berry, M., "Immunity-Based Aircraft Fault Detection System," *AIAA 1st Intelligent Systems Technical Conference*, American Institute of Aeronautics and Astronautics, Chicago, Illinois, 2004. https://doi.org/10.2514/6.2004-6277, URL https://arc.aiaa.org/doi/10.2514/6.2004-6277.

[44] Junkins, J. L., and Schaub, H., *Analytical Mechanics of Space Systems, Second Edition: Second Edition*, American Institute of Aeronautics and Astronautics, Reston ,VA, 2009. https://doi.org/10.2514/4.867231, URL https://arc.aiaa.org/doi/book/10.2514/4.867231.

[45] Montalvo, C., *Aerospace Mechanics and Controls*, University of South Alabama, 2022.

[46] Markley, F. L., and Crassidis, J. L., *Fundamentals of Spacecraft Attitude Determination and Control*, Springer New York, New York, NY, 2014. https://doi.org/10.1007/978-1-4939-0802-8, URL http://link.springer.com/10.1007/978-1-4939-0802-8.

[47] Clohessy, W., and Wiltshire, R., "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, Vol. 27, No. 9, 1960, pp. 653–658.

PUBLICATIONS

Journal Papers

- D.F, Hever Moncayo, Christoph Aoun, and **Tatiana Gutierrez**. "Adaptive Controller for Disturbance Rejection in Multi-Agent Systems". *Journal of Aerospace Information Systems.* 2022. (Accepted pending publication).

Conference Proceedings

- **Tatiana Gutierrez**, Nolan Coulter, Hever Moncayo, Yashwanth Kumar, Changrak Choi, Amir Rahmani and Akshita Gupta. "Distributed Health Management for Resilient Multi-agent Collaborative Spacecraft Inspection". *AIAA SciTech.* 2023.

- Andrei Cuenca, **Tatiana Gutierrez**, Eduardo Morillo, Brock Steinfeldt and Hever Moncayo. "Modeling of GPS Degradation Conditions for Risk Assessment of UAS Operations in Urban Environments". *AIAA SciTech.* 2023.

- **Tatiana Gutierrez**, Andrei Cuenca, Nolan Coulter, Hever Moncayo and Brock Steinfeldt. "Development of a Simulation Environment for Validation and Verification of Small UAS Operations". *AIAA SciTech.* 2022.

- D. F, Hever Moncayo, Christoph Aoun, and **Tatiana Gutierrez**. "Comparison of an Adaptive-Immunized and an Adversarial Deep Learning Control Laws to Increase Resiliency in Distributed Cyber-Physical Systems". *AIAA SciTech.* 2022.

Workshops

- **Tatiana Gutierrez**, Nolan Coulter and Hever Moncayo. "Health Management for Distributed Space Robotic System". *AIAA Intelligent Systems Workshop.* 2022.