POLITECNICO DI TORINO

in collaboration with RMIT University, Melbourne

Master’s degree course in Aerospace Engineering

Master’s degree thesis

Autonomous Onboard Health and Usage Management System for Smart Satellites

Supervisors

Dr Alessandro Gardi
Prof.ssa Manuele Battipede
Dr Andoh Michael Afful

Candidate

Mario Sibilla S275666

Ai miei genitori, Emanuele e Tommasa, per avermi guidato facendo luce sui sentieri bui e per avermi sostenuto incondizionatamente, a mia sorella Francesca, la mia stella luminosa e mia complice in ogni cosa, alla mia Sara, che mi ha fatto provare amore come mai l'avevo creduto possibile, a Voi dedico questo nuovo inizio della mia vita.

Index

Abstract

Over the past two decades, CubeSat technology has succeeded in making satellites compact and light-weight, thus affordable to the broader community. More recently, the maturation of Artificial Intelligence (AI) algorithms in conjunction with the significant advances in traditional Fault Detection, Isolation and Recovery (FDIR) techniques for space applications are supporting the development of Intelligent Health and Mission Management (IHMM) systems. These new intelligent systems are expected to provide proprioceptive and self-adaptation capabilities to satellites, thus granting them trusted autonomy.

"Intelligence is the ability to adapt to change", Stephen Hawking said during his Oxford University graduation speech. "Intelligence", "adapt", "change" are the keywords highlighting the important features to be implemented in future systems.

It is well known that space is not a friendly environment because of various unpredictable events, such as external perturbations that affect satellites orbit and attitude, which affect smaller platforms the most. IHMM systems exploit the complementary advantages of traditional physics-based FDIR and contemporary AI techniques to predict degradation of subsystems performance, implementing real-time system health forecasts that accommodate enough time to detect, identify and suddenly recover a possible fault, guaranteeing CubeSat safety and letting them fulfil their operations, while ensuring an acceptable level of functional capability.

The aim of this thesis is to define how IHMM systems shall be designed and integrated in a small satellite to enhance its operational autonomy, with a particular focus on mission-critical systems such as the Attitude Determination and Control System (ADCS) and the Communication system, which crucially enable a satellite to maintain contact with the ground and are subject to various faults. To accomplish this aim, the thesis will introduce the IHMM subsystems' design, followed by a failure analysis of the ADCS components with a possible diagnosis.

An AI-based algorithm will be also implemented in a Digital Twin of the satellite in order to predict the behaviour of the system and to diagnose or recover from the different faulty scenarios.

Lastly, results will be discussed in detail with a view towards future research.

Acknowledgments

I wish to thank my supervisors, Dr. Gardi, Prof. Sabatini, Prof. Battipede and Dr. Afful because of the opportunity to complete my academic journey with an abroad experience, which changed completely my life.

I wish to thank in particular Alex Gardi, for his patience, for helping me during the development of this thesis and for believing in me and in this project since it started.

Thanks to all my colleagues and friends from all over the world, I wouldn't be the same without all of you.

Thanks to my girlfriend Sara, for being next to me in every situation, since we were young.
I will be thankful for having you for all my life. Proud of being loved by you.

I must thank my parents and my sister too because they sustained me in these years of study, without doubting of my abilities and loved me without a limit.
I owe you all, forever.

1. Introduction

Since October 4th 1957, when the Soviet Union successfully launched Sputnik I, space science and technology research have greatly enhanced our understanding of Earth, its atmosphere, and our solar system, as well as outer space, through different space programs and missions [1].

The urgent need to explore space is also justified by the fact that the 21st century population growth has triggered energy and water shortages, and environmental damage [2].

Over all the discoveries and improvements that space research helped to realize, what is profoundly changed is the idea of *unknown space* and the concept of *distance in space exploration*. Starting from the idea of *unknown space*, the human will of observing the sky is anachronistic, but with space observation and exploration missions, it is possible to directly find high resolution images from different telescopes or past missions on the internet. For example, there is the Astronomy Picture of the Day Archive website, offered by NASA, which everyday offers different images to "Discover the cosmos", as it is written on their site [3].

Regarding the *distance in space exploration*, since last not near-Earth manned mission was the Apollo 11, unmanned spacecraft are serving as substitute for astronauts in exploring the solar system and probing the mysteries of the universe.

Multiple technologies have been developed for unmanned spacecraft, experiencing a real boost in studying and producing small satellites. One of the reasons for this boost is the substantial reductions in cost and time, which can be explained by focusing on the advances in microelectronics and integrated technologies of these last decades.

Small satellites include a different variety of satellites with a mass less than 180 kilograms, and they could be classified as: mini (100-180 kg), micro (10-100 kg), nano (1-10 kg), pico (0.01-1 kg) or femto-satellites (0.001-0.1 kg), depending how much they weigh. CubeSat technology, which is the one chosen for this case of study, is a subcategory of nanosatellites, but with a standard size and a form factor. The standard size is a *unit* (1U) and it is basically a cube with the dimensions of 10x10x10 cm, but it could be extended from 1.5 to 2, 3, 6, up to 12 U. [4]

These spacecraft have to function in an extreme environment, because of severe temperature cycling exposure, vacuum, micrometeoroid impacts and high energy ionising radiation, mostly on LEO and GEO orbits. So considering the harsh operating environment of a spacecraft and the inability to repair or replace its malfunctioning equipment, failures are some of the major challenges encountered during a space mission. One of the most critical systems to ensure extended survivability and operability of satellites is the Attitude Determination and Control System (ADCS), since in addition to compromising the payload operation, its failure usually compromises regular power generation and thermal management as well as communication with the ground. The Communication System (Comsys) is also critical in many mission stages since its failure would prevent the continuation of the mission even if the other systems are at least partially healthy.

Degradation of both these subsystems can considerably affect mission performance, and failure can result in the loss of the spacecraft. As shown in [5], the ADCS degrades and fails more frequently, harder (failures with more severe anomalies), and earlier in LEO than in GEO. To overcome this problem, [6] proposed reinforcement learning, a paradigm of machine learning, to derive a discrete neural spacecraft attitude control method without requiring high-performance computing. It has been also developed a quasi-time-optimal control solution to a highly constrained control problem, achieving an important level of pointing accuracy.

While in [7], deep learning was applied for fault detection and identification to detect actuator faults in nanosatellites. Models of the satellite were built using MATLAB, and a simplified model of the reaction wheel was designed. Satellite faults can be detected autonomously, even when they are not detected by the ground station, using the proposed model, and it is expected to be very useful for the autonomous operation of the mega-constellation mission that will utilize nanosatellites in the near future.

So, the implementation of IHMM in ADCS could represent a tangible solution for this problem. IHMM could change drastically also the feasibility of missions, mostly in terms of autonomy.
NASA definition of autonomy is "the ability of a system to achieve goals while operating independently of external control. […] is not the same as artificial intelligence (AI) but may make use of AI methods." [8]. Therefore, it becomes increasingly important to integrate autonomy in robotic space exploration to reach and operate in environments that have never

been explored, where the knowledge about that environment is uncertain, where the spacecraft's interaction with that environment is dynamic, where resources available to the spacecraft are limited, and where the harshness of that environment impacts and degrades the spacecraft's health.

Even in previously characterized environments, like the Moon and Mars, autonomy could improve productivity, increase robustness, and reduce costs, as seen in recent advances in autonomous landing and surface navigation for Mars 2020. [9] It is possible also measuring the autonomy of a system with different type of scales. The Sheridan scale [10] (Fig. 1), for example, defines ten criteria associated to ten levels that classify the level of autonomy:

- Level 1 coincides with full control of the operator,
- Levels from 2 to 4 handle with who decides,
- Levels from 5 to 9 deal with the way to implement the decision,
- Level 10 corresponds to the complete autonomy of the machine that decides if inform the human or not.

1	The computer offers no assistance, human must do it all.
2	The computer offers a complete set of action alternatives, and
3	narrows the selection down to a few, or
4	suggests one, and
5	executes that suggestion if the human approves, or
6	allows the human a restricted time to veto before automatic execution, or
7	executes automatically, then necessarily informs the human, or
8	informs him after execution only if he asks, or
9	informs him after execution if it, the computer, decides to.
10	The computer decides everything and acts autonomously, ignoring the human.

Figure 1 Sheridan Scale, image from [10]

Although autonomous systems are the focus of attention, human operators continue to play a crucial role in the decision-making process during operations, particularly when faulty systems and components are involved. In order to achieve trusted autonomous capabilities, as it was discussed in [11-12] an IVHM system driven by data-driven AI reasoning techniques and models would be required. In this way, the health condition could be monitored in real-time during operation, system faults could be reliably predicted, and resources could be rearranged rapidly to prevent catastrophic events and minimize the impact of single/multiple failures. The actual implementation of the IHMM, which will increase the level of autonomy of the mission, could reduce also the cost of missions near Earth, reducing telemetry, which is handled by the ground stations, and as a consequence reducing the number of people involved. Also the feasibility of long-distance space exploration missions, considering that, for example, there is a time delay of 40 minutes for communication to go and come back from Mars.

According to what was explained before, the integration of IHMM in the satellite design is critically important going forward.

This thesis goal is to develop a viable solution, by exploiting Artificial Intelligence algorithms to assess and predict the degradation of the ADCS performance in CubeSat and similar small-scale platforms, which could compromise the communication affecting the ComSys, and the main contributions are given as follows:

1. One important outcome of this work is to implement in CubeSats the concept of *Trusted Autonomy*, in other words systems capable of independently assessing the situation (own and external), selecting and performing mission tasks, running spacecraft systems, and making changes to operations without human interaction [13].
2. As a difference from literature [6-7, 14-15], this thesis will rely on a Digital Twin of the CubeSat to simulate and study the answer of the system when faults occur. One of the benefits of the Digital Twin technology is that it can create a near-real-time link between the physical and digital worlds. Models based on it are more realistic and holistic, offering deeper and more comprehensive measurements of uncertainty [16].

More in detail, a baseline mission will be considered in this study and for the CubeSat design choices.

The structure of this thesis will be the following: the literature overview in Chapter 2, followed by the methodology explained in the Chapter 3, the verification case and the simulation in the 4th, and at the end it will be the conclusions and the possible future works.

2. Literature overview

This chapter will cover an overview of the concept of autonomous technology and how the idea of FDIR has changed gradually in new and advanced designs and how it was applied in ADCS. Starting from Figure 2, it will be described the fundamental steps and innovations, while the detailed analysis of the studies will be in the chapter itself.

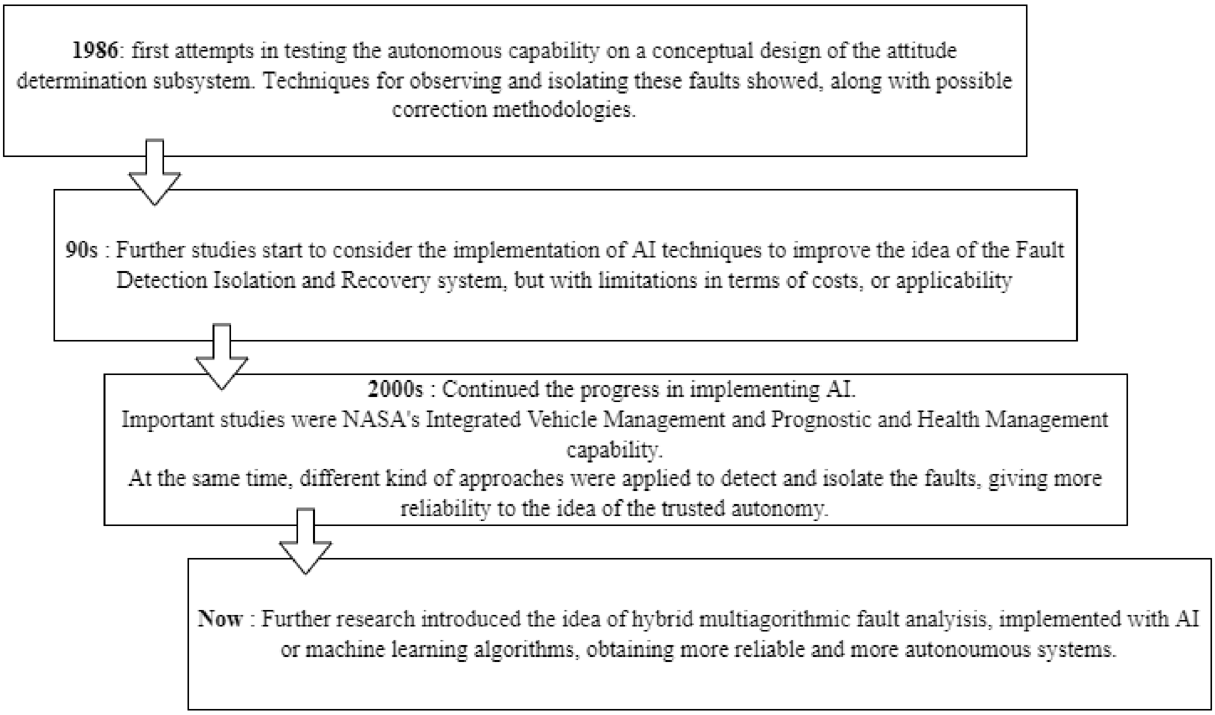

Figure 2 Summary of the fundamental steps covered in the literature overview

One of the first papers that involves the idea of testing the autonomous capability on a conceptual design of the attitude determination subsystem [17] was published in 1986. The presence of specific faults related to this specific system was also considered, while techniques for observing and isolating these faults were showed, along with correction methodologies. It appears clear anyway, that there was not enough confidence in these algorithms, supposing that it is needed that the ground station completes the eventual reconfiguration, after the fault observation and identification phases.

In the late 80s, the interest in improving space technologies to reduce weights and costs arose. Therefore, research in implementing neural networks for the development of instruments for, the spacecraft attitude determination and control were carried out, giving new hopes in this field [18,19].

A fundamental study that needs to be taken into account is part of the 1990 Nasa Conference on Artificial Intelligence for Space Application. In this conference was presented a prototype of a Maintenance and Diagnostic System (MDS) to apply to ADCS in order to improve its Fault Detection Isolation and Recovery (FDIR) system [20]. This project was meant to be for the Space Station Freedom, next converted into the International Space Station in 1993 under Clinton administration. In this paper it is highlighted the need to improve autonomous capability on *remote systems*, in other words that systems that could be unreachable, because physically distant to the crew, without an Extra-Vehicular Activity (EVA) or through robot systems. ADCS is considered a remote system and its maintenance is expensive, so the implementation of the MDS to the FDIR was studied to provide crucial information to support the FDIR in the fault isolation phase and at the same time giving it predictive capabilities.

The interest in developing an autonomous ADCS continued also in the 90s, for example with the 1996 Danish mission Ørsted, in which it was necessary matching the high requirements on autonomy while considering cost requirements that imposed cheap actuators [21]. Among the topics discussed are development of novel algorithms for attitude control based on magnetic torque, attitude determination schemes based on geomagnetic field measurements, and implementation into a supervisory control architecture. In addition, performance degradation was tolerated after the occurrence of a fault and the hardware redundancy was acceptable for few important components. However, the heritage of this paper is a satellite mission applicable idea when autonomy onboard is important, but with the limit of applicability of algorithms for satellite with a little number of sensors.

Another later 90s paper proposed a roadmap for realizing the significant gains that can be realized by enhancing spacecraft autonomy [22]. Important purposes of this paper are evaluated various AI techniques to determine whether they are suitable for use in various spacecraft functions, while defining a command-and-control architecture incorporating virtual prototyping as a means of integrating and migrating reusable code from the ground to the space segment. As a result of the study, the implementation of AI showed a reduction of about 30 % of total mission cost, opening a door for reusable software systems and for automating ground and space segment operations, but highlighting the need to have a standard high-level commanding language to reduce more the costs.

Going through the 2000s, continued the progress in implementing AI algorithms in order to automatize processes of faults diagnosis and their consequences and possible recommendation of solutions to let the human crew be able to pay attention to explore and study space. This was possible thanks to NASA's Integrated Vehicle Health Management (IVHM) that led the way to the future generation of space vehicles [23]. This study discusses IVHM techniques for future space vehicles. It presents how an IVHM could reduce, or even eliminate, many of the costly inspections and operations associated with future space transportation systems but considering that the development and use of an IVHM are heavily dependent on highly reliable sensors and processing software and in that way exposing the urge of continuing studies.

An innovative point of view for the 2000s is offered by [24,25], which are part 1 and part 2 of the 2005 IEEE Aerospace conference. In these two documents is explored the world of *prediction*, through Prognostic and Health Management (PHM) capability and the study of the useful life remaining capabilities. The term prognostics in PHM has a much broader definition than just prognostic functions alone as it includes fault/failure detection, fault / failure isolation, enhanced diagnostics, assessment of material condition, performance monitoring, and life tracking. These papers end admitting that predictive prognostics has several challenges, issues, and lessons to be learned, but with the conviction that this technology will mature and become more widely applied.

Going now more in depth with how the fault diagnosis was implemented in different designs to improve the autonomy of the satellite, it is useful start with the idea of the performance of fault diagnosis, given by a milestone in this field, professor P.M. Frank, in three steps: *residual symptom generation*, *residual evaluation* and *fault analysis* (to determine type, size and cause). He also explained that a fault diagnosis system must have the ability to diagnose as many small faults as possible with a minimum number of measurements and to be extremely robust to unknown inputs [26]. In this paper there is also the classification of different methods of fault detection based on the method of residual generation, such as *signal-based, analytical model-based* and *knowledge-based*. For Analytical Model-based approaches, there are different approaches, such as observer-based methods, which use some form of observer to measure the output of the system and then construct residuals using properly weighted output estimate errors. After that, the residual is examined for the presence of faults utilizing a decision rule based on a simple threshold test and a statistical decision theory. Ideally, the residual should be equal to zero, without faults or disturbances.

For example, in [27] a second order sliding mode observer is applied to the ADCS, which reconstructs the mapping of four reaction wheel faults into three principal axes. A different methodology it is explained in [28] where is implemented a nonlinear observer to calculate modified Rodriguez parameters and angular velocity vector. For the faulty recovery control, this paper presents the need of having a re-configurable algorithm to preserve the closed-loop system's stability and to tune the controller gains in the event of sensor failures. It is proposed also a controller that guarantees the uniformly ultimately boundedness in presence of sensor faults, but with a limit of a known bound of the tracking error.

Another approach is the one offered by [29], where an observer-based H_∞ output feedback fault-tolerant controller is designed, in order to stabilize the attitude system and maintaining the system performance. In this paper there were sensor and actuator faults at the same time. Firstly, virtual observers are used to reduce fault effects, and then a real observer is derived from them. With the newly developed observers, new criteria for designing H_∞ FTC methods have been established, ensuring that the faulty closed-loop attitude systems are asymptotically stable at a given level of disturbance attenuation.

Differently, in [30] for fault detection, isolation, and estimation is investigated a class of nonlinear systems. Specifically, for fault detection, a nonlinear observer is designed to minimize the uncertainty within $H\infty$ framework. Then, a series of nonlinear robust unknown input observers is defined in order to isolate the faulty actuator. Consequently, fault isolation is achieved based on the generalized observer strategy. By using the proposed observer, faults and states can be estimated simultaneously, and input disturbances can be decoupled, and model uncertainty and external disturbances can be attenuated.

An interesting point of view is proposed by [31], which presented a Multiple-Fault design scheme at two distinct levels, system and component. In the system level, two nonlinear observers, so double observers, based on analytical redundancy can diagnose multiple faults of the system, but this level can reveal the fault source roughly. Secondly, at the component level, a bank of Sliding mode observers is designed to determine precisely which actuators of ACDS are malfunctioning; this provides a clear indication of the root cause of the problem, as it is explained in Figure 3.

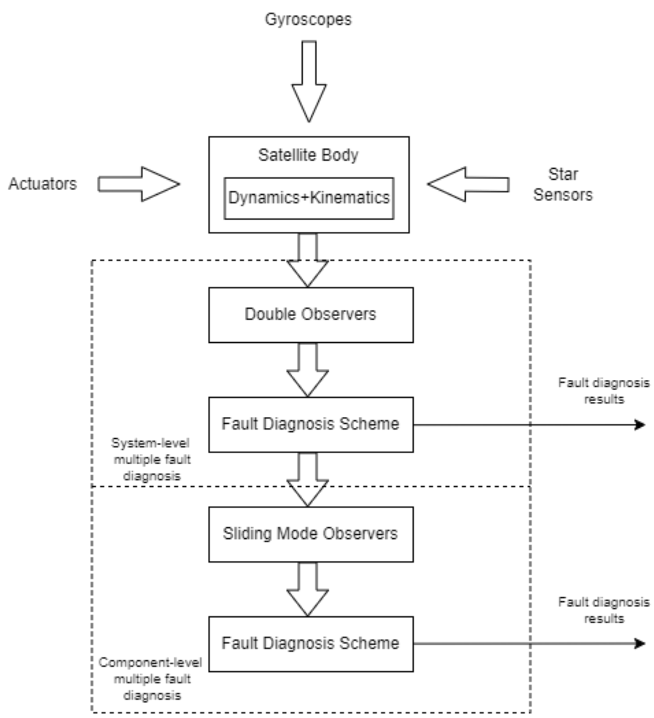

Figure 3 Multiple-Fault Design Scheme, adapted from [31]

Other different solutions to diagnose the occurrence of uncertainties and faults are offered for example by Gao et al. [33], which designed two Extended Kalman filters and a fault diagnosis scheme based on analytical redundancy, in order to use the residuals given by EKFs to identify successfully faults and the element that has caused them.

Pourtakdoust et al. [34] presented an approach in which the presence of external disturbances and sensor faults, the angular velocity and attitude of a rigid gyro-less satellite are estimated and controlled using a modified Square Root Unscented Kalman Filter (MSRUKF). This study also presents and proves a concept that can analytically decouple the unhealthy orientation signal from the healthy one using a proposed optimization process for sensor installation.

Lee et al. [7] offers another possible perspective, where a new method based on deep learning is proposed for detecting and identifying the faults in the reaction wheel. Furthermore, the proposed model enables the satellite to detect faults autonomously.

Guo et al. [32], on the other hand, proposed a solution where after the fault identification, a series of radial basis function neural network (RBFNN)-based observers are designed to isolate and estimate the faults. The design of Fault Detection Observers and Fault Identification Observers is also improved by the use of a series of Disturbance Compensation

Observers to estimate the disturbances in the system. These disturbance estimates are then applied to the fault diagnosis algorithm, making feasible a Multiple Fault Analysis.

Another interesting approach is proposed by Lee et al. [35], where a multi-algorithmic hybrid ADCS for a small satellite has been designed. A hybrid automaton framework was used to implement multiple control and estimation algorithms, as well as condition-based switching strategies. The hybrid automaton has also been described in detail in terms of its states and transition conditions. As it is possible to understand from Figure 4, the switching mechanism is controlled by different thresholds, which define what strategy will be implemented for the specific case.

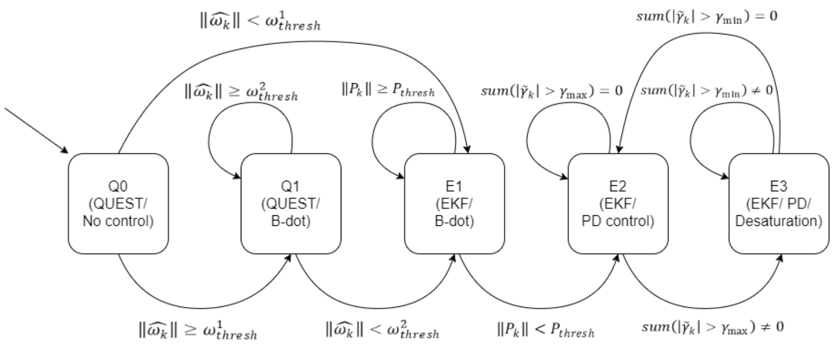

Figure 4 multi-algorithmic hybrid ADCS, adapted from [35]

3. Methodology

3.1 Mission baseline and requirements

The satellite designed for this work is composed by COTS (commercial-off-the-shelf) components.

The Nasa Engineering and Safety Center definition of COTS cited by the 2014 technical update [56] is "*an assembly or part designed for commercial applications for which the item manufacturer or vendor solely establishes and controls the specifications for performance, configuration, and reliability (including design, materials, processes, and testing) without additional requirements imposed by users and external organizations.*"

Based on this definition, the idea of developing a small satellite becomes accessible not only at federal industries or economically developed countries, but it is available also for smaller countries and universities, or research centres.

Moreover, the presence of these kind of components allows to impact less if a fault occurs, differently from a more complex system implemented, which is more expensive and could be fatal for the entire mission. [57]

Referring to the heritage of other missions, this one will be able to check the Earth in an observation mission for remote sensing. The mission considered as a reference for specifications and requirements is the ESA Cryosat-2.

The CryoSat-2 is the replacement mission of the original one that was lost for a launch failure in 2005, but keeping the original mission objectives, such as monitoring the thickness of land ice and sea ice and trying to explain the connection between the rise in sea levels and the melting the polar ice, while giving a contribution to understand how this is influencing the climate change.

Figure 5 CryoSat-2 logo, with the courtesy of ESA

This chapter is divided into the following sections:

- Next part of this paragraph to define the mission requirements and the design preliminaries and choices for this thesis,
- The detailed description of the component models, with a summary of faults and possible diagnosis, in order to simulate efficiently in MATLAB/Simulink environment,
- The attitude representation equations, followed by the Satellite Kinematics and Dynamics,
- The implementation of the applied Unscented Kalman Filter,
- How the Data Analysis will be executed in order to apply the proposed solution,
- How the Fault Detection and the Fault Identification will be carried out through the equations to implement in MATLAB/Simulink,
- A summary of the proposed algorithm, which represents the core contribution of this thesis.

The chosen baseline mission for this thesis is an Earth-Observation Mission in LEO. This choice is the product of the will of analysing an exemplified mission to design IHMM to define an adaptable model for further studies, with the advantage of having more accurate data and information available about the environmental conditions and disturbances.

Since the CryoSat-2 is an Earth-Observation mission in LEO, the orbit specifications and the spacecraft parameters will be considered as the selected for this work.

The spacecraft parameters are listed in Table 1, while the orbital parameters are detailed in Table 2.

Table 1 Spacecraft Parameters, obtained from Earth online Portal [58]

Overall Dimensions	4.60 m x 2.34 m x 2.20 m
Mass (fuel included)	720 kg
Attitude	3-axis stabilized local-normal pointing, with 6 degrees nose-down attitude, using magneto-torquers
Power	2x GaAs body-mounted solar arrays, with 850 W each at normal solar incidence; 78 Ah Li-ion battery

Table 2 Orbital parameters, obtained from Committee on Earth Observation Satellites [59]

Semi-major axis (km)	7095348.56
Eccentricity	0.0005098
Inclination (deg)	92.0369
Right Ascension of the Ascending Node (deg)	288.4013
Argument of Perigee (deg)	159.0008
Mean Anomaly (deg)	201.1410

While talking about the spacecraft configuration and instruments, CryoSat-2 has a rectangular shape, with solar arrays forming a sort of tent. It has also the lower part faces continuously the Earth. The antennas used for radio communication, and the Laser Retroreflector, are mounted on this surface; an emergency antenna for command and monitoring is also fitted on top of the satellite between the solar arrays. The two SIRAL (SAR/Interferometric Radar Altimeter) instrument antenna dishes are mounted on a separate rigid bench in the forward section of the S/C. In addition, a dedicated SIRAL radiator is mounted at the nose tip. [40]

Considering the main system considered in this work, the ADCS, the CryoSat-2 mission is composed of the following elements:

- A cold gas system for attitude control and orbit transfer and maintenance maneuvers,
- Three magnetorquers for compensation of environmental disturbance torques,
- Three star-tracker heads providing autonomous inertial attitude determination for the spacecraft. This makes the sensor system one-failure tolerant, except for the occurrence of simultaneous sun and moon blinding of two heads, to which the system

software is tolerant. Furthermore, the star tracker attitude is useful to be a reference for the orientation of the SIRAL interferometric baseline,

- A DORIS receiver measures the Doppler frequency shifts of UHF and S-band signals transmitted by ground beacons. It has an accuracy < 0.5 mm/s in radial velocity, allowing an absolute determination of the orbit position with an accuracy of 2-6 cm,
- • CESS (Coarse Earth-Sun Sensor) to provide attitude measurements (<5º) with respect to the sun and Earth for initial acquisition and coarse pointing,
- • A set of three three-axis fluxgate magnetometers are used for magnetorquer control and as rate sensors. They provide a measurement range of at least ± 60.000 nT with an accuracy of better than 0.5 % full scale.

Figure 6 CryoSat-2 satellite configuration, courtesy of ESA

On the other hand, Figure 7 will present the CubeSat design choices. In this figure are represented only the systems that could be directly related to this study. As it could be seen, the payload block is also represented and it is composed by instruments for Earth-Observation, such as Hyperspectral Camera[1] and the Lidar[2].

Another important aspect that is highlighted by the legenda is that there are different linkages between the different blocks. In particular, the blue arrows indicate the power line, while the red one indicate the data line.

[1] A hyperspectral imaging camera measures continuous bands of wavelengths within the electromagnetic spectrum and can collect and process the information. A wide range of wavelengths is also covered in the recorded spectra with fine wavelength resolution.

[2] Lidar is a method for determining variable distances by targeting an object with a laser and measuring the time for the reflected light to return to the receiver. Lidar is commonly used to make high-resolution maps.

Looking at the figure it could be noted that, for what concern the Command and Data Handling, there are two double-headed red arrows. These links take to two blocks encircled by a green dashed line, not only the ADCS, which will be involved in implementing AI algorithms in this thesis, but also the Communication System.

Between these systems it stands an inherent bond, justified by the fact that the correct pointing of antennas, which allows communication with ground stations, it is strictly related to external disturbances whom the satellite is affected when in orbit, measured and corrected by ADCS.

So, it appears important consider that the communication system would be influenced, in order to have a complete understanding of the design choices done in this case study.

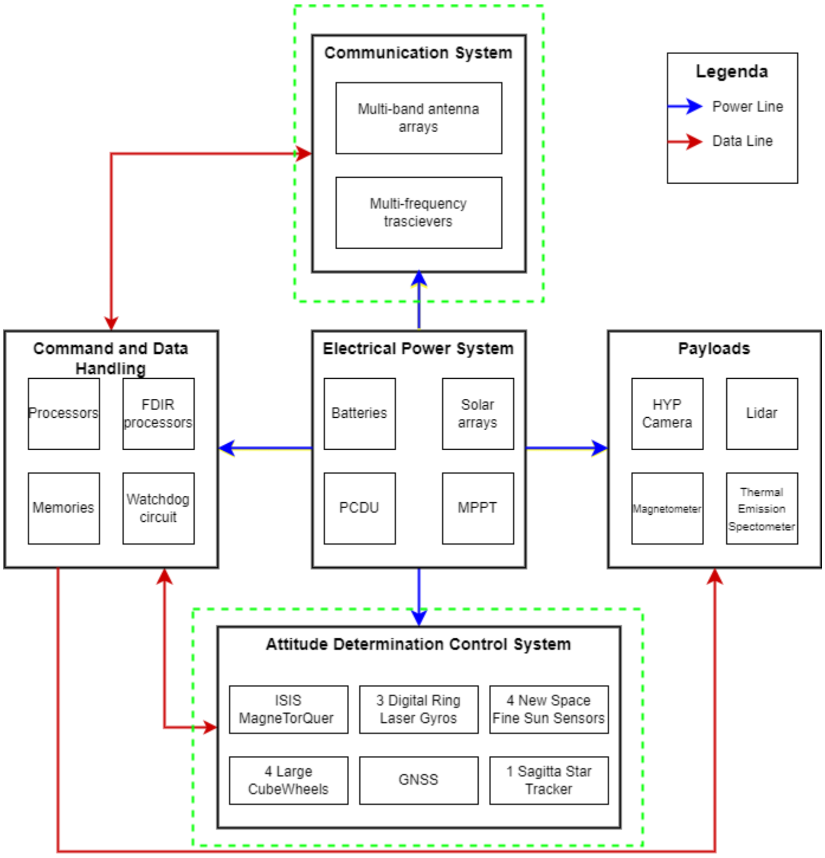

Figure 7 CubeSat design choices

Another important point, it is to have an idea of the mission requirements to model correctly the ADCS.

Table 3 shows the product requirements for the CryoSat-2 mission:

Table 3 ADCS requirements, from [40]

Requirement	Value
Cross-track pointing knowledge for SARIn[3] mode	< 10 arcsec
Pointing accuracy per axis in the nominal Earth-pointing phase of the mission	< 0.2º
Pointing stability for 0.5 s in the nominal Earth-pointing phase of the mission	< 0.005º

These requirements will represent constraints to take into account for the simulation phase of this thesis, considering them as requirements also for this case study.

3.2 Component models

As an innovative point of view, in this thesis, a hybrid approach is proposed to detect and identify attitude control failures based on an analytical Model-based approach.

The detection system would be designed as an intelligent system which combines knowledge of spacecraft dynamics control and knowledge of the components behaviour through a digital twin of the system and artificial intelligence algorithms.

The aim of these steps is to allow the system to detect and report anomalies in real time with high accuracy, in order to improve pointing accuracy, while increasing satellite autonomous capabilities.

To efficiently use the digital twin approach, in order to simulate accurately the system behaviour when a fault occurs, it will be used a Model-based approach and so it will be modelled the main components of the instruments used in ADCS:

- A brushless DC motor for the reaction wheel,
- A baffle for the star tracker,
- An air core and three torque rods for the magnetorquer board,
- A Global Positioning System receiver.

[3] SARIn refers to the SAR interferometric mode

Before proceeding with the actual models, the classification of the components, their possible faults, and a possible strategy to diagnose them are listed in Tables from 4 to 6.

Table 4 Faults and Diagnosis for Actuators

Component	Fault	Diagnostic
Brushless DC Motor [36]	Broken rotor bar, Static and Dynamic air gap irregularities, Dynamic eccentricity, Winding short, Bearing and gearbox failure.	• Temperature measurements, • RF emission monitoring, • Noise and vibration monitoring, • Motor Current Signature Analysis (MCSA), • AI & NN based techniques, • Electromagnetic field monitoring.
Inverter	Loss of one or more of the switches of a phase, Short circuit of a switch, Opening of one of the lines to the machine.	
Rotor	Damaged rotor magnets, Damaged Hall sensors.	
Stator	Breakdown of the winding insulation.[4]	
Magnetorquer [46]	Broken wire or bad soldering, Component burned, Short circuit to the power voltage, Misalignment of the magnetorquer, Short circuit between the magnetorquer and the power voltage, Faulty supply voltage.	• Analysis of the residual based on Biot-Savart Law, • Discrepancy between estimated and actual actuator torques, • Decreasing control voltage range.

[4] Caused by large electrical voltage stresses, electro-dynamic forces produced by winding currents, thermal aging from multiple heating and cooling cycles, and mechanical vibrations from internal & external sources. [36]

Table 5 Faults and Diagnosis for the Star tracker

Component	Fault	Diagnostic
Baffle [47] (Star tracker)		
Random errors	Shot noise of the flux from the star,	Decreasable with the increase of exposure time.
	Random dark signal,	Reduceable by cooling the sensor.
	Readout noise of the sensor.	Select the right mode of operation or reading out with a lower frequency.
Systematic errors	Image pixelization,	Calculate the correction to the measured coordinates for a certain Point Spread Function.
	The effect of "hot" pixels,	Exclusion from the consideration of stars which images contain "hot" pixels.
	Bias non-uniformity.	To eliminate this error the map of bias needs to be stored and taken into account while image processing.

As it possible to notice for the star tracker, all faults founded are affecting the imaging process and not the mechanical part, which is the baffle. The baffle could be affected to other disturbances that are not considered as proper diagnosable faults, such as micro-meteoroids impact and possibility of shifting of the structure because of vibrations due to the launch or the deployment.

Micrometeoroids with velocities of tens of kilometres per second pose a significant environmental hazard to spacecraft. As meteoroids impact, their damage is different depending on their size, density, porosity, speed, and direction. A repeated impact of micron-sized to submm-sized particles can cause gradual degradation of spacecraft surfaces through erosion or cratering, which may affect for example mirrors, lenses, and sensors. Bigger particles can also perforate baffles or insulation layers [48]. For these reasons, in this thesis, won't be considered the faults from the baffle, because they are unpredictable, talking about micro-meteoroids impacts, or previously taken into account when the baffle was optimized in the design phase.

Lastly, Table 6 is referred to the GPS receiver and its possible faults.

Table 6 Faults and Diagnosis for GPS receiver

Component	Fault	Diagnostic
GPS receiver [52]	Noise and Resolution,	Reduceable by using appropriate filtering techniques
	Ephemeris Prediction,	Evaluable through an equation function of Along Track (ATK), Across Track (XTK) and Radial (RAD)
	Clock Offset,	Negligible in most positioning applications, residual ineliminable because the corrections are periodic
	Group Delays,	Estimated on the ground before launch and corrections in the navigation message
	Multipath errors.	Techniques such as the Narrow Correlator, the Double Delta/Strobe Correlator, or the Vision Correlator by Fenton and Jones, are useful, but not capable of eliminating the errors completely
	Antenna obscuration	It is necessary to determine the Line of Sight of the satellite with respect to the antenna phase centre
	Radiofrequency interference Intentional and Unintentional (jamming)	Filtering and suppression techniques are applied for detecting or anti-jamming

The mathematical models of the components with their equations will be discussed in the next paragraphs for a better understanding of the physics behind, and they will be useful for the implementation in MATLAB/Simulink environment.

3.2.1 Brushless DC motor model

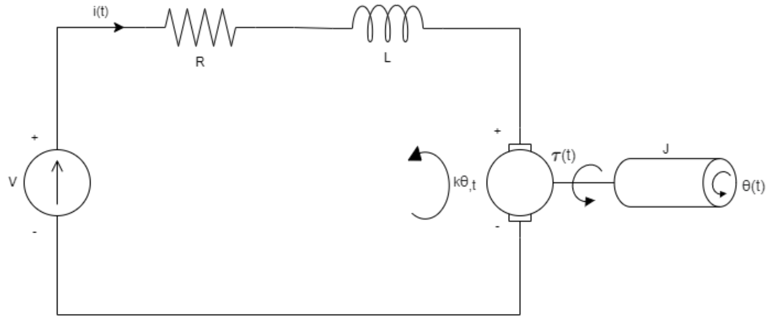

Figure 8 BLDC motor model circuit

This R-L circuit represents the BLDC motor functioning, using an armature to convert the current in electro-mechanical force that generate a torque [36]. This motor includes also permanent magnets, and this induces a back electro-magnetic frequency (emf) in the armature. The variables are:

- R, the resistance
- L, the inductance[5]
- k, the back-emf constant
- θ,t the angular velocity of the motor shaft
- J motor moment of inertia

To obtain a model, it is useful to solve the circuit, applying the Kirchhoff's law to voltage:

$$Ri(t) + L\frac{di}{dt} = V - k\frac{d\vartheta}{dt} \tag{3.1}$$

Now, considering the mechanical part, the generate torque is proportional to the current $i(t)$:

$$J\frac{d^2\vartheta}{dt^2} = ki(t) \tag{3.2}$$

Anyway, this equation is approximative, because it should be considered that there are not losses, but it is not realistic.

[5] Note that the inductance should be the L-M where L is the self-inductance and M is the mutual inductance

To ensure more grip on reality, it is possible to add a damping coefficient to multiplicate the angular velocity that model the loss:

$$J\frac{d^2\vartheta}{dt^2} - b\frac{d\vartheta}{dt} = ki(t) \tag{3.3}$$

So, the set of the equations appears to be:

$$\begin{cases} Ri(t) + L\frac{di}{dt} = V - k\frac{d\vartheta}{dt} \\ J\frac{d^2\vartheta}{dt^2} - b\frac{d\vartheta}{dt} = ki(t) \end{cases} \tag{3.4}$$

Now, the transfer function will be:

$$H(s) = \frac{k}{LJs^3+(JR-bL)s^2+(k^2-Rb)s} \tag{3.5}$$

In order to be more accurate, it should be considered not the ideal circuit but the non-ideal one [37].

Starting from the resistor, it is modelled as a circuit with a lumped leaked inductance L_{lead} is considered to be in series with the ideal bulk resistance R, that is in parallel with the parasitic capacitance $C_{parasitic}$.

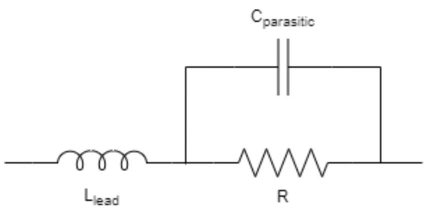

Figure 9 Non-Ideal Behaviour Circuit for Resistor, adapted from [37]

While C parasitic is the combination of a lead and a leakage capacitance.

$$C_{parasitic} = C_{lead} + C_{leakage} \tag{3.6}$$

The total impedance will be determined by calculating the sum between the impedance of the RC part and the lead inductor:

$$Z_{RC} = \frac{R}{1+j\omega RC_{par}} \tag{3.7}$$

$$Z_L = j\omega RL_{lead} \tag{3.8}$$

With this model the impedance of the equivalent circuit will be the sum of these two:

$$Z_R = \frac{j\omega L_{lead} + R(1-\omega^2 L_{lead} C_{par})}{1 + j\omega R C_{par}} \tag{3.9}$$

To exemplify this expression, it will be assumed that:

Assumption 1. The range of frequencies considered is limited to only low frequencies.

With this assumption, it could be considered that $\omega^2 L_{lead} C_{par} \ll 1$ and $\omega R C_{par} \ll 1$, and $L_{lead} \ll R$, so after mathematical manipulation the impedance of the resistor will be:

$$Z_R = R \tag{3.10}$$

Applying this model in MATLAB environment, it is possible to notice that for low frequencies the circuit is working as a simple resistor:

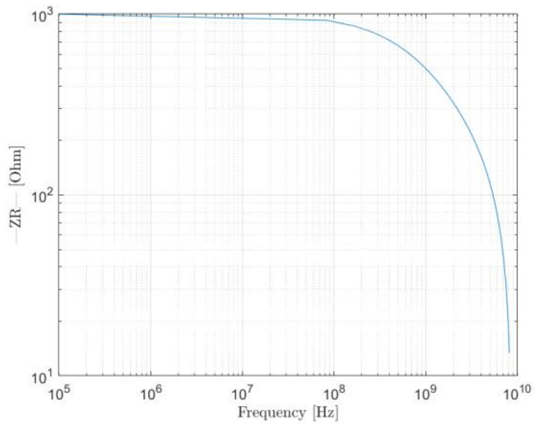

Figure 10 Resistor model implemented in MATLAB environment

For the inductor then, the real circuit will be modelled as:

- A series inductance L_{lead} and a parallel C_{lead} with the ideal inductance because of the wire leads of the inductor,
- A parasitic resistance in series with the ideal inductance to consider the large amount of wire in the inductor coil,

- A parasitic capacitance in parallel, due to the individual windings of the coil being close to one another.

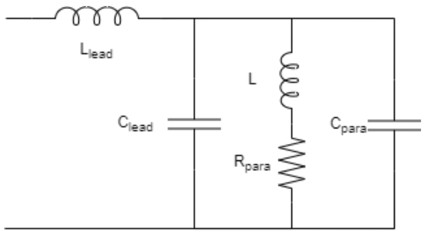

Figure 11 Complete Non-Ideal Behaviour Circuit for Inductor, adapted from [37]

But it is possible to simplify more this first model, because:

- $L_{lead} \ll L$ so, it will therefore be neglected,
- $C_{lead} \ll C_{parasitic}$, same as before.

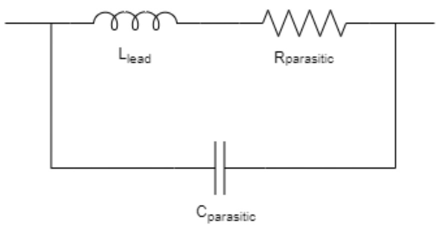

Figure 12 Simplified Non-Ideal Behaviour Circuit for Resistor, adapted from [37]

With this model the impedance of the inductor will be:

$$\frac{1}{Z_L} = \frac{1}{j\omega L + R_{par}} + j\omega C_{par} \tag{3.11}$$

This will lead to:

$$Z_L = \frac{j\omega L + R_{par}}{1 - \omega^2 L C_{par} + j\omega R_{par} C_{par}} \tag{3.12}$$

For the *Assumption 1*, the impedance could be written as:

$$Z_L = R_{par} + j\omega L \tag{3.13}$$

Applying this model in MATLAB environment, it is possible to notice that for low frequencies the circuit working as the model:

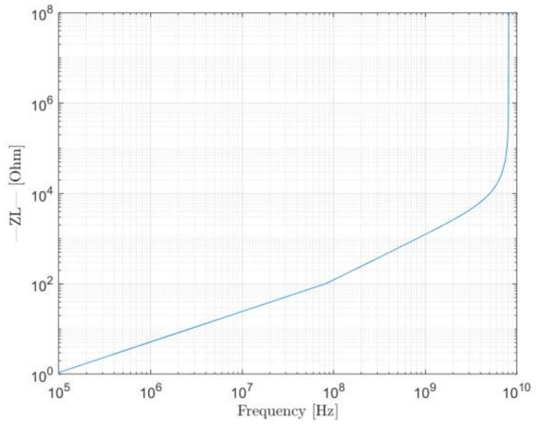

Figure 13 Inductor model implemented in MATLAB environment

3.2.2 Baffle

There are many factors that limit the performance of star sensors, including stray light level on the detector, non-uniformity, and slopes of the corresponding irradiances. A baffle prevents direct lightning from the Sun or the Earth from striking optical surfaces.

Arnoux, in [38], designed an optimization of a star sensor baffle, reducing the length and improving the global attenuation and decreasing the stray light irradiance slopes at the detector level, while the baffle attenuation alone has been slightly degraded.

The optimized model, which is shown in the figure, was designed following different steps.

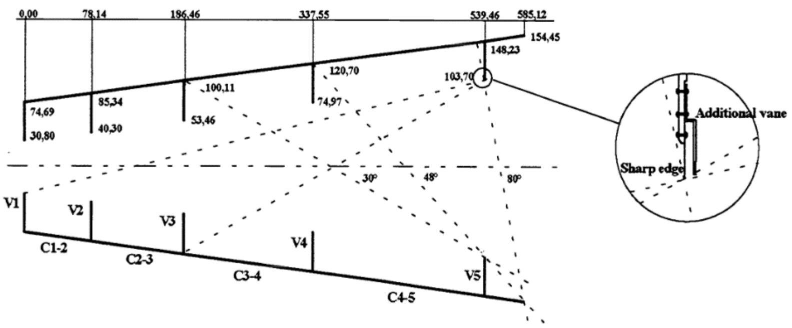

Figure 14 Optimized baffle design, image from [38]

Starting from the numbers of vanes that will be five, it is defined that the middle vane shall not be illuminated by the Sun. Furthermore, the Baffle is split in two sections by the middle vane.

The first section length, d_2 is defined as

$$d_2 = \frac{2a[\tan\varphi_S - \tan\varphi_T] - (r+\varepsilon)[\tan\varphi_T - \tan\alpha_2]}{[\tan\varphi_T - \tan\alpha_2][\tan\beta - \tan\varphi_S] - [\tan\varphi_S - \tan\varphi_T][\tan\alpha_1 - \tan\alpha_2]} \tag{3.14}$$

While the total length of the baffle L

$$L = \frac{2a[\tan\beta + \tan\varphi_S] - (r+\varepsilon)[\tan\alpha_1 - \tan\alpha_2]}{[\tan\varphi_T - \tan\alpha_2][\tan\beta + \tan\varphi_S] - [\tan\varphi_S - \tan\varphi_T][\tan\alpha_1 - \tan\alpha_2]} \tag{3.15}$$

And the entrance port radius b will be

$$b = L\tan\varphi_T - a \tag{3.16}$$

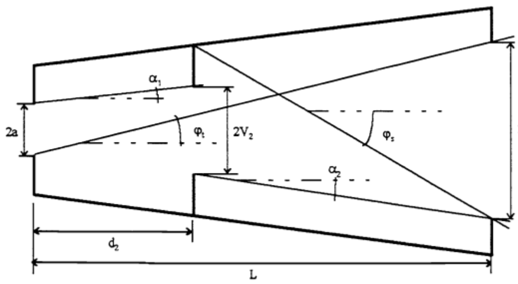

Figure 15 Parameters definition, image from [38]

After that, it is possible to calculate the second vane position and the depth of the first one.

$$Z_1 = \frac{LV_2 + bd_2 - (a+r)(L-d_2)}{V_2 + b + (L-d_2)\tan\beta} \tag{3.17}$$

$$U_1 = a + r + Z_1\tan\beta \tag{3.18}$$

$$d_1 = \frac{2aZ_1}{U_1 + a - Z_1\tan\alpha_1} \tag{3.19}$$

$$V_1 = a + d_1\tan\alpha_1 \tag{3.20}$$

$$U'_0 = \frac{V_1 L + bd_1}{L - d_1} \tag{3.21}$$

$$U_0 = a + r \tag{3.22}$$

In order to solve these equations, it need to solve the last two, which should be $U'_0 = U_0$.

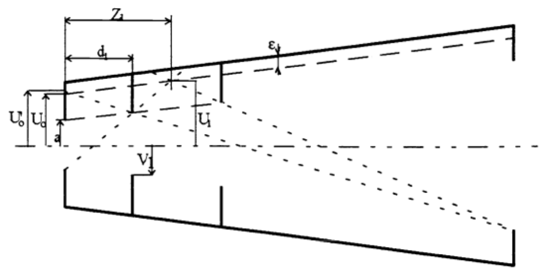

Figure 16 First two vanes depth design, image from [38]

It is also possible implementing the same method $U'_4 = U_4$ to calculate the 4th vane position and the depth of the last vane:

$$U_3 = \frac{(V_2+a)(a+r)+ad_2 \tan\beta}{V_2+a-d_2 \tan\beta} \tag{3.23}$$

$$Z_3 = \frac{(2a+r)d_2}{V_2+a-d_2 \tan\beta} \tag{3.24}$$

$$d_3 = \frac{LU_3+bZ_3-(L-Z_3)(V_2-d_2 \tan\alpha_2)}{U_3+b+(L-Z_3)\tan\alpha_2} \tag{3.25}$$

$$U'_4 = \frac{V_3+a}{d_3}(L-a) \tag{3.26}$$

$$U_4 = a + r + L\tan\beta \tag{3.27}$$

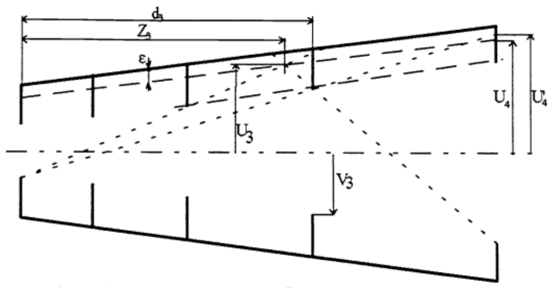

Figure 17 Fourth and fifth vane depth design, image from [38]

3.2.3 Magnetorquer

The magnetorquer considered in this work is composed of two torque rods and one air core torquer, so it will be important to define the main design parameters [39] such as: generated dipole, mass, power consumption and occupied volume and interference.

Generated dipole is one of the most important features for the fact that is strictly related to the efficiency of the control torque and, for that reason, should be determined with an accurate trade-off between the different requests for the mission.

Mass is mostly affected by the number of turns and the area of the coil for air core, but for torque rods is affected by the presence of the metal core in it.

Also, the power consumption is a fundamental parameter that is strictly connected to the total resistance of the wire, because of Second Ohm's Law. Anyway, it is also related to the temperature of the conductor, tending to decrease while the temperature will increase.

By now, it should be necessary to divide the phenomenology of the two different actuators to model accurately both.

Air core torquer

Starting from generated dipole, the formula is:

$$m = nIS \tag{3.28}$$

Where m is the magnetic dipole intensity[6], S is the coil area and n is the number of turns.

For the magnetic field there is a formula to define it for a rectangular shape coil, where l_1 and l_2 are the dimensions:

$$B_z = nI\frac{2\mu_0\sqrt{(l_1^2+l_2^2)}}{\pi(l_1h_2)} \tag{3.29}$$

The torque will be:

$$\vec{T} = \vec{m} \times \vec{B} \tag{3.30}$$

Torque rods

Starting from the magnetic dipole, this case will highlight more difficulties in designing that kind of magnetorquer, because of intrinsic properties and demagnetizing factor that should be taken into account.

[6] Unit is A*m^2

The magnetic dipole expression will be the sum of two different elements, the dipole due to the solenoid and the one inducted by the core's magnetization, in that way it could be better modelled the torque rod behaviour:

$$m = NIS + V_C M \tag{3.31}$$

Where N, I and S are specifically the number of turns, the current, and the solenoid area, while V_C and M are the core volume and the magnetization of the core. Referring at [5], the final expression for *m* is

$$m = NI\pi G \tag{3.32}$$

Where G is

$$G = r^2 + \frac{r^2(\mu_r - 1)}{(1 - N_d + \mu_r N_d)} \tag{3.33}$$

So, it is a parameter representative of the shape of the core and its magnetic properties.

With *r* and N_d defined as:

$$r = \sqrt[3]{\frac{V}{\pi\left(\frac{l}{r}\right)}} \tag{3. 34}$$

$$N_d = \frac{4\left[ln\left(\frac{l}{2}\right) - 1\right]}{\left(\frac{l}{r}\right)^2 - 4\, ln\left(\frac{l}{r}\right)} \tag{3. 35}$$

Note that N_d is specific for cylindrical core, with *r* and *l* as the radius and the length.

At the same time the expression of B is related at the current and the core shape:

$$B = \frac{\mu_0 \mu_r NI}{l(1 - N_d + \mu_r N_d)} \tag{3.36}$$

As the previous type of magnetorquer, the torque will be

$$\vec{T} = \vec{m} \times \vec{B} \tag{3.37}$$

3.2.4 GPS receiver

The GPS receiver is a component of the satellite that allows to improve the precision of attitude measurement, since it is based on the Global Navigation Satellite Systems.
As Sabatini et al. [52] documented in their review, GNSS errors are continuously studied and

their remediation techniques improved in order to mitigate the degradation of signal, to achieve better performances and to make the system more trustworthy.

As explained in Table 6 in the previous section to implement the GPS receiver, it is necessary to consider the presence of possible faults, but since the mission considered for this study is a LEO mission, it will not be necessary considering all the faults listed in the paper.

The first type of faults that will be considered, it is related to the Noise and Resolution, but it is not mathematically modellable, since it is reduceable with filtering techniques or with more precise receivers.

Another error is the *ephemeris prediction error*, which can be described with the (3.38):

$$ERR = RAD\ cos\alpha + ATK\ sin\,\alpha\ cos\,\beta + XTK\ sin\,\alpha\ sin\,\beta \qquad (3.38)$$

where α and β is the angle between the Line Of Sight user-satellite and the satellite vertical and β is the angle between the ATK direction and the satellite target containing the Line Of Sight and the satellite vertical. It is relevant to notice that United States Department of Defence precise ephemeris is observed from ground stations, while the non-DoD agencies or organizations use models to predict precisely the position.

For the *Clock Offset* and the *Group Delays errors*, the first one it is reduceable with a correction with polynomial coefficients in order to reduce the error, while the second ones are fixable with an estimation before launch and corrections while in space.

Multipath errors are a kind of error that is not negligible and it is an error source that needs to be considered, but can be modelled and deleted only in static points, with consecutive observations.

Other types of sources of errors like the *propagation errors*, are excluded because of the presence of the satellite in LEO, so the signal for satellite to satellite will not be delayed because of the atmospheric effects of ionosphere and troposphere.

Different approach for the *antenna obscuration*, which can be modelled. In order to determine the obscuration, it needs to determine the LOS of the GNSS satellite with respect to the antenna phase centre. The formula is:

$$T_E^a = T_b^a * T_N^b * T_E^N \qquad (3.39)$$

where T_b^a is the transformation matrix between the aircraft body frame and the antenna frame, T_N^b is the transformation matrix from ENU (East-North-Up) to body frame and T_E^N is the ECEF (Earth Centred Earth Fixed) to ENU transformation matrix.

Last error that will be considered is *radiofrequency interference*, which could be intentional or unintentional and it causes a degradation of the accuracy of the navigation or a complete loss of the receiver tracking. For these kinds of error there are different anti-jamming or jamming detection techniques, also for civil use.

For more information about sources of errors, these are presented more in depth in the Ref.[52].

3.3 Attitude representation

In order to have a representation of the satellite in orbit, it will be used a body-fixed frame with axes aligned to the principal axes of inertia and the origin in the centre of mass, but also a reference frame with origin at the center of the earth and axes pointing to the center of the earth, along the orbital velocity.

The attitude representation is the expression of the orientation of the satellite body-fixed frame toward reference frame and it could be showed by different types of representation including direction cosine matrix, Euler angles, quaternions, and Modified Rodriguez Parameters.

The direction cosine matrix is very inefficient because it is difficult to enforce the six constraints, while Euler angles have issues related to trigonometric functions, and 4-element quaternion vector has ambiguity problems. [41]

So, it will be used MRP vector to represent the attitude of the satellite. The MRPs can be considered as a normalized version of the Euler parameters. Let's start with denoting with Φ the principal angle and with $\hat{e}$ the principal axis associated with Euler's theorem [42]. The Euler parameters are defined by:

$$\begin{cases} q_0 = cos(\phi \ / \ 2) \\ q_i = \hat{e}_i \, sin(\phi \ / \ 2) \ (i = 1,2,3) \end{cases} \qquad (3.40)$$

MRP eliminate the Euler parameter constraint[7] and reduce the number of coordinates from four to three, introducing σ:

$$\sigma_i = \frac{q_i}{1+q_0} \quad (i = 1,2,3) \tag{3.41}$$

Furthermore, the MRP are related to the principal angle and the principal axis through:

$$\sigma = \hat{e}\, tan(\phi \,/\, 4) \tag{3.42}$$

It is important to notice that MRPs have a singularity in [0,2π] range, avoidable using the MRP shadow set [43]:

$$\sigma_i^S = \frac{\sigma_i}{\sigma^T\sigma} = \hat{e}\, tan\left(\frac{\phi-2\pi}{4}\right) \tag{3.43}$$

The shadow points have important properties, because they go singular at the zero rotation and to zero at ±2π principal rotation, so basically the opposite behaviour of σ.

3.4 Kinematics and Dynamics

So, the kinematic equation can be expressed through this set of differential equations [44]:

$$\begin{cases} \dot{\sigma} = G(\sigma)\omega \\ \sigma(0) = \sigma_0 \end{cases} \tag{3.44}$$

Where ω is the angular velocity of the body-fixed frame in the reference one but expressed in the body frame and G is defined as:

$$G(\sigma) = \frac{1}{2}\left(I_{3X3} - S(\sigma) + \sigma\sigma^T - 1 + \frac{\sigma^T\sigma}{2} I_{3X3}\right) \tag{3.45}$$

S(.) is the 3x3 skew-symmetric matrix, defined as follow:

$$S(\sigma) = \begin{bmatrix} 0 & \sigma_3 & -\sigma_2 \\ -\sigma_3 & 0 & \sigma_1 \\ \sigma_2 & -\sigma_1 & 0 \end{bmatrix} \tag{3.46}$$

At the same time, the dynamics of the rotational motion of a rigid body can be described through this set of equations in body-fixed frame:

---

[7] $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$

$$\begin{cases} J\dot{\omega} = S(\omega)J\omega + u \\ \omega(0) = \omega_0 \end{cases} \tag{3.47}$$

Where J is the satellite inertia matrix and u is the applied torque vector in body axes [45].

Considering that $x = [\sigma^T \ \omega^T]^T$ is the state variable, $y(t)$ is measured by sensors, it is possible to express the state-space model of the studied ADCS, based on attitude kinematics and dynamics:

$$\begin{cases} \dot{x}(t) = \phi(x)x(t) + Bu \\ y(t) = Cx(t) \end{cases} \tag{3.48}$$

Where,

$$\phi = \begin{bmatrix} 0_{3X3} & G(\sigma) \\ 0_{3X3} & J^{-1}S(\omega)J \end{bmatrix}, B = \begin{bmatrix} 0_{3X3} \\ J^{-1} \end{bmatrix}, C = I_{6X6} \tag{3.49}$$

3.5 Unscented Kalman Filter

In order to calculate the satellite attitude, it will be implemented an attitude estimation algorithm, specifically a Kalman Filter.

Since the problem analysed is nonlinear, the choice can be limited between an Extended Kalman Filter, an Unscented Kalman Filter or a Particle Filter, which are the mostly applied for local estimation.

As it showed and explained in [44], if the dynamical models have Gaussian noise the UKF is more accurate than the EKF, because it stores the second order moments, while it doesn't need a local approximation.

Another reason for using UKF, instead of EKF, is because MRP based on EKF has the big limit that can be applied to a specific range in which MRP vector is nearly linear.

The Unscented Kalman Filter is described as it follows.

Starting from the calculation of the root mean square S_{k-1}^{+} of covariance matrix by Cholesky method, where P_{k-1}^{+} is the covariance matrix:

$$P_{k-1}^{+} = S_{k-1}^{+} {S_{k-1}^{+}}^{T} \tag{3.50}$$

After that, Sigma point is evaluated as:

$$x_{k-1}^{+(i)} = \hat{x}_{k-1}^{+} + \sqrt{n}S_{k-1;i}^{+}, i \leq n \tag{3.51}$$

$$x_{k-1}^{+(i)} = \hat{x}_{k-1}^{+} - \sqrt{n}S_{k-1;i-n}^{+}, i > n \tag{3.52}$$

Considering that *n* is the matrix dimension.

The Sigma points will be propagated, with τ_s as a time interval, by:

$$x_{k}^{-(i)} = x_{k-1}^{+(i)} + f(x_{k-1}^{+(i)}, t_k)\tau_s \tag{3.53}$$

The state and error covariance after propagating, will be described by:

$$x_k^{-} = \frac{1}{2n}\sum_{i=1}^{2n} x_k^{-(i)} \tag{3.54}$$

$$P_k^{-} = \frac{1}{2n}\sum_{i=1}^{2n} (x_k^{-(i)} - \hat{x}_k^{-})(x_k^{-(i)} - \hat{x}_k^{-})^T + Q_{k-1} \tag{3.55}$$

With Q_{k-1} as the system noise covariance.

The measurements update will be, from the new Sigma point:

$$P_k^{-} = S_k^{-}S_k^{-T} \tag{3.56}$$

$$x_{k}^{-(i)} = \hat{x}_{k}^{-} + \sqrt{n}S_{k;i}^{-}, i \leq n \tag{3.57}$$

$$x_{k}^{-(i)} = \hat{x}_{k}^{-} - \sqrt{n}S_{k;i-n}^{-}, i > n \tag{3.58}$$

The average measurement innovation is defined by:

$$\begin{cases} \delta z_k^{-(i)} = z_k - h(\hat{x}_k^{-(i)}, t_k) \\ \delta z_k^{-} = \frac{1}{2n}\sum_{i=1}^{2n} \delta z_k^{-(i)} \end{cases} \tag{3.59}$$

Also, the covariance of innovation is:

$$C_{\delta z,k}^{-} = \frac{1}{2n}\sum_{i=1}^{2n} (\delta z_k^{-(i)} - \delta z_k^{-})(\delta z_k^{-(i)} - \delta z_k^{-})^T + R_k \tag{3.60}$$

The next step is calculating the gain matrix, state and error covariance matrix update:

$$K_k = [\frac{1}{2n}\sum_{i=1}^{2n} (\delta z_k^{-(i)} - \delta z_k^{-})\left(\delta z_k^{-(i)} - \delta z_k^{-}\right)^T] + (C_{\delta z,k}^{-})^{-1} \tag{3.61}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \delta z_k^- \quad (3.62)$$

$$P_k^+ = P_k^- - K_k C_{\delta z,k}^- {K_k}^T \quad (3.63)$$

Furthermore, in [55] there are also the shadow-set transformation and all the equations for the shadow state.

3.6 Data Analysis

As it was presented in Fig.6, the new concept presented in this thesis for the ADCS algorithm includes the presence of sensors measurement data analysis, in order to detect possible anomalies that will trigger the fault investigation path, which will be useful to detect and identify possible faulty components, preventing potential problems before they occur.

As part of data analysis and data-driven anomaly detection, it will be analysed the similarity of the monitoring parameters based on distance measurements. An appropriate similarity measure can properly reflect the gradual and small changes in the monitoring series, and thus, can help identify abnormalities more efficiently, classifying the telemetry data through an anomaly detection algorithm, such as the k-Nearest Neighbor (KNN) classification [48][49].

For a better understanding of the time series pattern and finding the similarity between two unequal sequences, dynamically warping the time axis is necessary. This is possible through the Dynamic time warping (DTW) technique, as explained in [50] and as it will presented in the following part of this paragraph.

First, it defines a time series Q and a time series C, each of which has a length of n and m, respectively:

$$Q = q_1, q_2, \dots, q_i, \dots, q_n \quad (3.63)$$

$$C = c_1, c_2, \dots, c_j, \dots, c_m \quad (3.64)$$

To compare the two time series, it is necessary to align them. Through DTW it is possible to construct a matrix that has as i^{th} and j^{th} element the alignment between the two points q_i e c_j, obtained through the calculation of their Euclidean Distance[8].

---

[8] $d(q_i, c_j) = (q_i - c_j)^2$

The Warping Path, indicated as *W*, establishes a mapping between the two series because it is defined as:

$$W = w_1, w_2, \dots, w_k, \dots, w_K \tag{3.65}$$

Where K is ranging between *max(m,n)* and *m+n-1*.

It is fundamental to impose all the constraints that influence W. It is required that:

- The warping path starts in one corner cell and it will finish in the diagonally opposite one,
- A step to be allowable has to be in the adjacent cell, including the diagonally one,
- The points will be monotonically spaced in time.

It is needed to have the warping path that satisfies the conditions, but that minimizes the warping cost at the same time, defined as:

$$DTW(Q, C) = min\left(\frac{\sqrt{\sum_{k=1}^{K} w_k}}{K}\right) \tag{3.66}$$

K is useful to compensate, in case of different lengths.

In order to solve (3.51), it is possible to define the cost matrix α:

$$\alpha\ (i, j) = d(i, j) + min\{\alpha(i, j-1), \alpha(i-1, j-1), \alpha(i-1, j)\} \tag{3.67}$$

Considering the constraints for the frame elements[9], $\alpha\ (i, j)$ allows to define the warping path calculating the actual map and defining the minimum warping cost DTW(Q,C).

As it is proposed and tested in [49], with the framework in figure 18, it is possible to detect anomalies offline and online with two different types of algorithms.

[9] $\alpha(0,0)=0$, $\alpha(i,0)= \alpha(0,j)=+\infty$

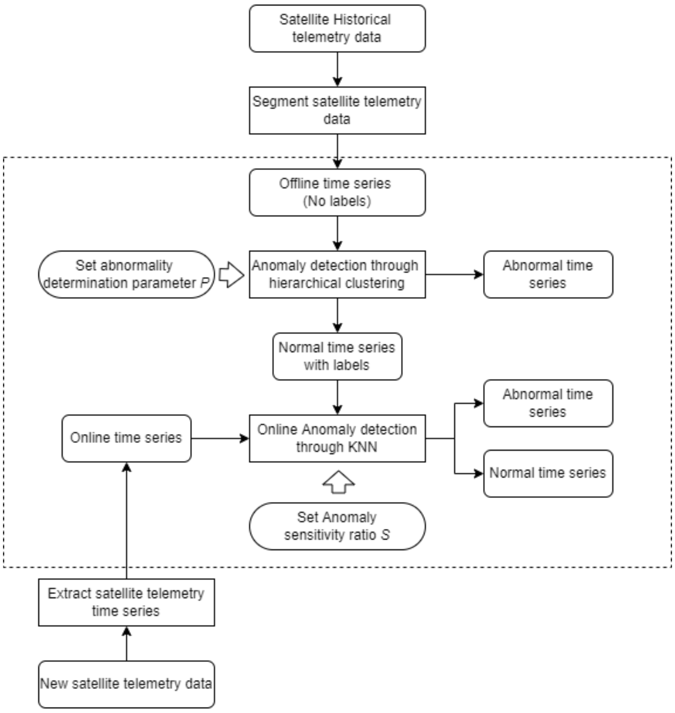

Figure 18 Framework of anomaly detection for telemetry data, adapted from [49]

The next step will be completed through the KNN classification, which will allow to detect possible anomalies online.

In order to have a better understanding of this algorithm, it will be defined the equations as it was done in [49], in which is possible to find the specific framework for the KNN algorithm too.

KNN classification is one of the most used algorithms for data mining, which estimates if a data point is to be a member of a group or another depending on what group the data points nearest to it are in.

To calculate the average distance inside a class *j*:

$$\overline{s_j} = \begin{cases} \frac{2}{n_j(n_j-1)} \sum_{x_{jk} \in X_j} \sum_{x_{jt} \in X_t} dist(x_{jk}, x_{jt}), n_j > 1 \\ min(\overline{S}), n_j = 1 \end{cases} \tag{3.68}$$

Where:

- n_j is the number of samples in j,
- x_{jk} and x_{jt} are the k$_{th}$ and the t$_{th}$ sample in j,

- X_j and X_t are the training data sets,
- $\mathrm{dist}(x_{jk}, x_{jt})$ is the distance between these two calculate with the DTW as explained before,
- $\overline{S}$ is the series of average distance inside each class.

Following the framework for KNN, explain in depth in [49], the most important part is the one about the anomaly judgement.

To obtain it, it needs to calculate the d_{min}, which is the minimum distance between x' (the test time series to be detected) and the samples in class l'. This will be fundamental for the judgement:

$$\begin{cases} d_{min} > S \cdot \overline{s_{l'}}\ , \ anomaly\ detected \\ d_{min} \leq S \cdot \overline{s_{l'}}\ , \ normal\ series \end{cases} \qquad (3.69)$$

At the same time, the satellite historical telemetry data will be analysed, while the hierarchical clustering will be applied to the unlabelled time series to detect possible anomalies. For this part of the algorithm, the main goal is to delete the anomalous class of data in order to keep the normal series. This will be done through the *P* parameter, the *abnormal determination parameter*, to determine the clusters. If this $P > \frac{n_i}{N}$, the class considered is anomalous, being n_i the number of members in the class and N the total number of members.

To have a better understanding of the hierarchical clustering algorithm, refers to Fig. 16, adapted from [54], which explains the steps for the Nearest Neighbor Chain Algorithm.

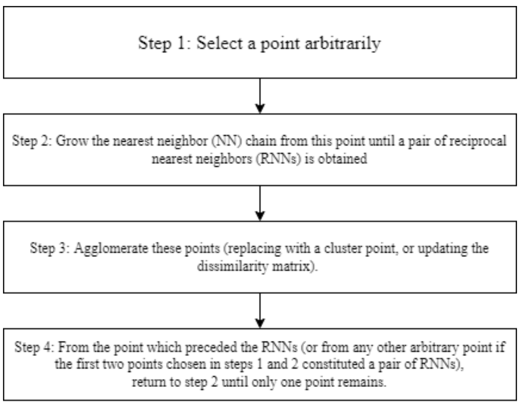

Figure 19 Steps Nearest Neighbor Chain Algorithm, adapted from [54]

3.7 Fault Detection

As it was previously explained, once the presence of an anomaly in the data is detected, it will be triggered the fault detection part of the algorithm.

In order to detect any sensor or actuator fault without restrictions, it will be implemented the observer-based multiple fault diagnosis algorithm presented in [31] that, as explained in the literature overview, it is applied in two different levels, the system and the component one.

Starting from the system level, it will be set up a double observer, one for actuators faults based on the satellite dynamics, while the other for star sensors based on the kinematics.

According to (3.46), the state-space model for the kinematics will be:

$$\begin{cases} \dot{x}(t) = \phi_\sigma(x)x(t) \\ y(t) = C_\sigma x(t) \end{cases} \tag{3.70}$$

Where $x = \sigma$.

It is possible to design the observer, as follows:

$$\begin{cases} \dot{\hat{x}}(t) = \phi_\sigma(\hat{x})\hat{x}(t) + L_1(y(t) - \hat{y}(t)) \\ \hat{y}(t) = C_\sigma \hat{x}(t) \end{cases} \tag{3.71}$$

Where the state estimation is $\hat{x}(t)$ and the output estimation is $\hat{y}(t)$.

The state estimation error is defined as $e(t) = x(t) - \hat{x}(t)$ and the output estimation error is $\varepsilon_1(t) = y(t) - \hat{y}(t)$.

The goal of designing the observer is to reach an observer gain L_1 that allows state estimation $\hat{x}(t)$ to converge asymptotically to the state variable $x(t)$, under the condition of free fault.

It is possible to obtain, through the definition of state estimation error and output estimation error, the state error equation:

$$\dot{e}(t) = (\phi_\sigma - L_1 C_\sigma)e(t) \tag{3.72}$$

$$\varepsilon_1(t) = C_\sigma e(t) \tag{3.73}$$

In order to have the error dynamics of the observer to be robustly stable to the unknown vector without any fault, Theorem 1 should be respected.

Theorem 1. For the given constant γ_1 and δ_1, if existing a matrix M_1 and a positive-definite, symmetric matrix P_1 such that:

$$\Pi_1 = \begin{bmatrix} \Lambda_1 + C^T C & C^T - P_1 L_1 \\ 0 & \left(1 - {\delta_1}^2\right) \end{bmatrix} < 0 \tag{3.74}$$

$$\Lambda_1 = {M_1}^T P_1 + P_1 M_1 + {\gamma_1}^2 P_1 P_1 + I \tag{3.75}$$

In this way, the state estimate error asymptotically goes to zero in a fault free condition, and the observer gain will be $L_1 = -M_1 C^{-1}$.

For the detailed proof procedure, referring to [51].

Moreover, it will be considered a fault free situation if $\|\varepsilon_1(t)\| < \lambda_1$, where λ_1 represents the threshold for the fault diagnosis. In the other cases, it will be detected the presence of a fault in the sensor.

Now again, according to (3.46), the state-space model for the dynamics will be:

$$\begin{cases} \dot{x}(t) = \phi(x)x(t) + Bu \\ y(t) = Cx(t) \end{cases} \tag{3.76}$$

Where $x = \omega$.

It is possible to design the observer, as follows:

$$\begin{cases} \dot{\hat{x}}(t) = \phi_\omega(\hat{x})\hat{x}(t) + B_\omega u + L_2(y(t) - \hat{y}(t)) \\ \hat{y}(t) = C_\omega \hat{x}(t) \end{cases} \tag{3.77}$$

Where the state estimation is $\hat{x}(t)$ and the output estimation is $\hat{y}(t)$.

The state estimation error is defined as $e(t) = x(t) - \hat{x}(t)$ and the output estimation error is $\varepsilon_2(t) = y(t) - \hat{y}(t)$.

The goal of designing the observer is to reach an observer gain L_2 that allows state estimation $\hat{x}(t)$ to converge asymptotically to the state variable $x(t)$, under the condition of free fault.

It is possible to obtain, through the definition of state estimation error and output estimation error, the state error equation:

$$\dot{e}(t) = (\phi_\omega - L_2 C_\omega)e(t) \tag{3.78}$$

$$\varepsilon_2(t) = C_\omega e(t) \tag{3.79}$$

In order to have the error dynamics of the observer to be robustly stable to the unknown vector without any fault, Theorem 2 should be respected.

Theorem 2. For the given constant γ_2 and δ_2, if there are a constant $\beta_1 > 1$, and positive-definite, symmetric matrices Q_1 and S, such that

$$(1-\beta_1)C^T C + {\gamma_2}^2 Q_1 Q_1 + \frac{1}{\delta_z^2} Q_1 E E^T Q_1 + I = -S \qquad (3.80)$$

In this way, the state estimate error asymptotically goes to zero in a fault free condition, and the observer gain will be $L_2 = \frac{1}{2}\beta Q_1^{-1} C^T$.

For the detailed proof procedure, referring to [51].

Moreover, it will be considered a fault free situation if $\|\varepsilon_2(t)\| < \lambda_2$, where λ_2 represents the threshold for the fault diagnosis. In the other cases, it will be detected the presence of a fault in the actuator.

This configuration of double observers is used to run in parallel, so the logic of the diagnosis will be:

Table 7 Diagnostic logic based on Double Observers, adapted from [31]

Diagnostic logic	*Presence of the fault*
$\|\varepsilon_1(t)\| < \lambda_1$ and $\|\varepsilon_2(t)\| < \lambda_2$	Fault free
$\|\varepsilon_1(t)\| \geq \lambda_1$ and $\|\varepsilon_2(t)\| < \lambda_2$	Sensor fault
$\|\varepsilon_1(t)\| < \lambda_1$ and $\|\varepsilon_2(t)\| \geq \lambda_2$	Actuator fault
$\|\varepsilon_1(t)\| \geq \lambda_1$ and $\|\varepsilon_2(t)\| \geq \lambda_2$	Both sensor and actuator fault

Considering the component level, it will be designed a bank of sliding mode observers, as done in [31].

In case of actuators fault, the dynamics model with the fault will be in this form:

$$\begin{cases} \dot{x}(t) = \phi_\omega(x)x(t) + B_\omega(u+f) \\ y(t) = C_\omega x(t) \end{cases} \qquad (3.81)$$

Where $f(t) = [f_1(t), f_2(t), f_3(t)]$ is the fault function and it satisfies the assumption to be bounded, for each axis: $\|f(t)\| \leq \rho(t)$.

It will be designed three sliding mode observers, one for each axis, to produce estimations and diagnose the actuator fault in that specific axis.

For better describing the model, another way could be:

$$\begin{cases} \dot{x}(t) = \phi_\omega(x)x(t) + B_\omega u + B_{fi}f_i \\ y(t) = C_\omega x(t) \end{cases} \tag{3.82}$$

So, it will be defined $B_f = [B_{f1}, B_{f2}, B_{f3}] = \mathrm{B}$, the *fault distribution matrix*.

As a consequence of this, the sliding mode observer will be defined as follow:

$$\begin{cases} \dot{\hat{z}}_i(t) = F_i z_i(t) + T_i B_\omega u + T_i B_\omega \mu_i(t) + H_i y(t) \\ \hat{x}_i(t) = z_i(t) + N_i y(t) \\ \hat{y}(t) = C_\omega \hat{x}_i(t) \end{cases} \tag{3.83}$$

Where $\hat{x}_i(t)$ and $\hat{y}_i(t)$ are the estimated vectors in the i-th axis, $\mu_i(t)$ is the non-linear control input, $\mathrm{z}_i(\mathrm{t})$ is the state variable of the observer, $\mathrm{F_i}, T_\mathrm{i}, H_\mathrm{i}, N_\mathrm{i}$ are matrices that are designed to allow the state estimation $\hat{x}_i(t)$ to converge asymptotically to the state variable $x(t)$, under the condition of the i-th actuator fault.

The state estimation error is defined as $e_i(t) = x(t) - \hat{x}_i(t)$ and the output estimation error is $\varepsilon_{3i}(t) = y(t) - \hat{y}_i(t)$.

Furthermore, it is necessary to set these conditions:

$$\begin{cases} T_i = I - N_i C \\ F_i = K_i C \\ H_i = -K_i(I - N_i C) \end{cases} \tag{3.84}$$

To make the i-th observer estimation sensitive to the fault in order to decouple signals, these constraints will be applied:

$$\begin{cases} T_i B_{fi} \neq 0 \\ T_i B_{fj} = 0, if\ j \neq i \end{cases} \tag{3.85}$$

How to determine all the matrices is explained in depth in [31].

Considering **Theorem 2** and the condition that the fault function is norm-bounded, it will be considered this other theorem:

Theorem 3. It will be defined the nonlinear input control variable, such as:

$$\mu_i = \begin{cases} -\rho \frac{F_i \varepsilon_{3i}(t)}{\|F_i \varepsilon_{3i}(t)\|}, & \varepsilon_{3i}(t) \neq 0 \\ 0, & \varepsilon_{3i}(t) = 0 \end{cases} \tag{3.86}$$

For the detailed proof procedure, referring to [53].

It is also possible, to reduce the buffeting, modifying the control input to make it continuous:

$$\mu_i = -\rho \frac{F\varepsilon_{3i}(t)}{\|F\varepsilon_{3i}(t)\| + \delta} \tag{3.87}$$

Where δ is a small positive number.

This configuration of sliding mode observers is used to run in parallel, so the logic of the diagnosis will be:

Table 8 Diagnostic logic based on Sliding Mode Observers, adapted from [31]

Diagnostic logic	*Presence of the fault*
$\mu_1 = 0, \mu_2 = 0, \mu_3 = 0$	Fault free
$\mu_1 = 1, \mu_2 = 0, \mu_3 = 0$	X-axis actuator fault
$\mu_1 = 0, \mu_2 = 1, \mu_3 = 0$	Y-axis actuator fault
$\mu_1 = 0, \mu_2 = 0, \mu_3 = 1$	Z-axis actuator fault
$\mu_1 = 1, \mu_2 = 1, \mu_3 = 0$	X-axis and Y-axis actuator fault
$\mu_1 = 1, \mu_2 = 0, \mu_3 = 1$	X-axis and Z-axis actuator fault
$\mu_1 = 0, \mu_2 = 1, \mu_3 = 1$	Y-axis and Z-axis actuator fault
$\mu_1 = 1, \mu_2 = 1, \mu_3 = 1$	All actuators fault

3.8 Fault Identification

In order to identify the fault, it will be implemented a hierarchical clustering method that allows the cluster centre to be specified.

As it showed in [54], it is possible to summarize all the hierarchical methods through the *Lance-Williams dissimilarity update formula*.

This formula allows to check the dissimilarity between two points in an agglomerated cluster and all the other points, and it is defined as:

$$d(i \cup j, k) = \alpha_i\, d(i,k) + \alpha_j\, d(j,k) + \beta\, d(i,j) + \gamma |d(i,k) - d(j,k)| \tag{3.88}$$

Table 9, adapted from [54] gives the specifications of the values of α_i, β, γ to choose in the formula for the hierarchical clustering method decided.

It is important to specify for better understanding of this table that:

- $|i|$ is the number of objects in cluster i,
- g_i is a vector of dimension m, where m is the set of attributes,
- $\|.\|$ is the norm in Euclidean metric,
- $\alpha_i = \alpha_j$.

Table 9 Specifications of Hierarchical Clustering Methods, adapted from [54]

Hierarchical Clustering Method	Lance-Wiliams Dissimilarity Update formula values	Coordinates of Center of Cluster, which Agglomerates Clusters i and j	Dissimilarity between Cluster Centers g_i and g_j
Single link (nearest neighbour)	$\alpha_i = 0.5$ $\beta = 0$ $\gamma = -0.5$		
Complete link (diameter)	$\alpha_i = 0.5$ $\beta = 0$ $\gamma = 0.5$		
Group average (average link)	$\alpha_i = \frac{\lvert i\rvert}{\lvert i\rvert + \lvert j\rvert}$ $\beta = 0$ $\gamma = 0$		
McQuitty's method	$\alpha_i = 0.5$ $\beta = 0$ $\gamma = 0$		
Median method	$\alpha_i = 0.5$ $\beta = -0.25$ $\gamma = 0$	$g = \frac{g_i + g_j}{2}$	$\lVert g_i - g_j\rVert^2$
Centroid	$\alpha_i = \frac{\lvert i\rvert}{\lvert i\rvert + \lvert j\rvert}$ $\beta = \frac{\lvert i\rvert\lvert j\rvert}{(\lvert i\rvert + \lvert j\rvert)^2}$ $\gamma = 0$	$g = \frac{\lvert i\rvert g_i + \lvert j\rvert g_j}{\lvert i\rvert + \lvert j\rvert}$	$\lVert g_i - g_j\rVert^2$
Ward's method (minimum variance, error sum of squares)	$\alpha_i = \frac{\lvert i\rvert + \lvert k\rvert}{\lvert i\rvert + \lvert j\rvert + \lvert k\rvert}$ $\beta = \frac{\lvert k\rvert}{\lvert i\rvert + \lvert j\rvert + \lvert k\rvert}$ $\gamma = 0$	$g = \frac{\lvert i\rvert g_i + \lvert j\rvert g_j}{\lvert i\rvert + \lvert j\rvert}$	$\frac{\lvert i\rvert\lvert j\rvert}{\lvert i\rvert + \lvert j\rvert}\lVert g_i - g_j\rVert^2$

For this case study, the approach which will be implemented is a cluster centre method, with a *stored data approach*, which steps are described in:

1. Examination of all dissimilarities and formation of cluster from two closest points,
2. Replacing two clustered points with a point that represents its centre of gravity,
3. Restarting form step 1, until all points are in one cluster.

Considering that the method applied will be the *Median method*, the Lance-Williams dissimilarity update formula will be:

$$d(i \cup j, k) = \frac{d(i,k)}{2} + \frac{d(j,k)}{2} - \frac{\beta\, d(i,j)}{4} \tag{3.89}$$

So, the new cluster centre will be distant from the point k:

$$\left\| k + \frac{i+j}{2} \right\|^2 \tag{3.90}$$

With this algorithm, a dendrogram and a graph representing the clusters will be created, in order to try to identify the presence of possible anomalies in the data, which highlight the presence of possible faults, to allow to identify them.

3.9 Proposed algorithm

In this paragraph the proposed algorithm will be explained and discussed, to give a summary of the studied approach showed in the previous paragraphs.

Starting from the Figure 20, a standard ADCS framework is presented.

Within this framework, it is possible to follow the following steps:

1. The sensors give as output their measurements and telemetry and those will be the input of the Attitude Determination block, composed by the UKF;
2. The output of the UKF block will result in the Attitude Error signal, while it is compared to the satellite desired attitude;
3. This signal will be the command input for the controller block, that will calculate the necessary torque to apply to the satellite;
4. Before the application of the calculated torque, it will be considered the presence of the Orbital Environment Disturbance Torques, which are not negligible;

5. This will result in the application of the total torque, which will cause the change of the attitude of the satellite, and it will close the loop, because of the sensors measuring the new attitude.

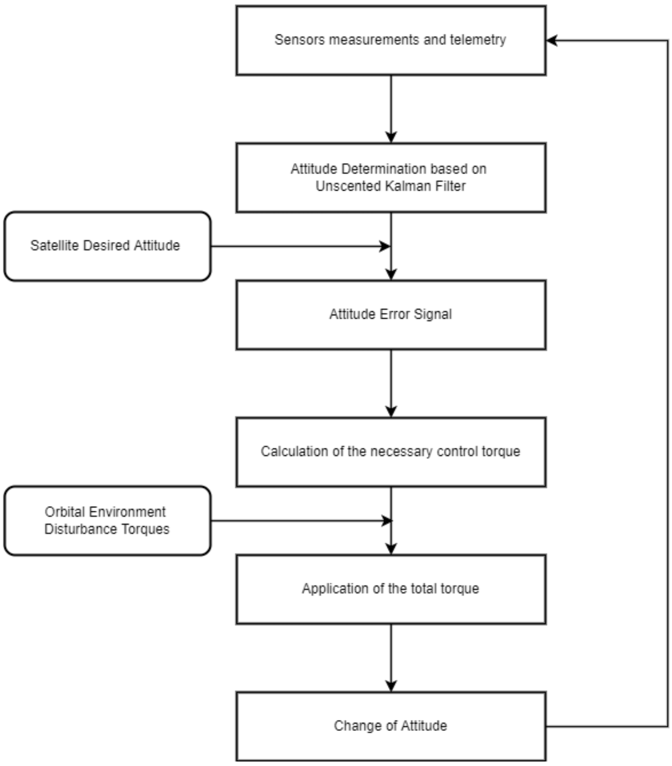

Figure 20 Standard ADCS framework

On the other hand, in Fig. 21, it is proposed the ADCS framework for this thesis.

As it possible to notice, there are some differences from the standard loop presented in Fig. 17, and these will be analysed in the following part of this paragraph.

The first difference in the proposed approach is the presence of a branch from the sensor measurements and telemetry block.

This branch will be divided into two different paths, as it possible to notice from the scheme. The red path represents the *data integrity verification loop*, while the blue one represents the *health monitoring and diagnostic loop*.

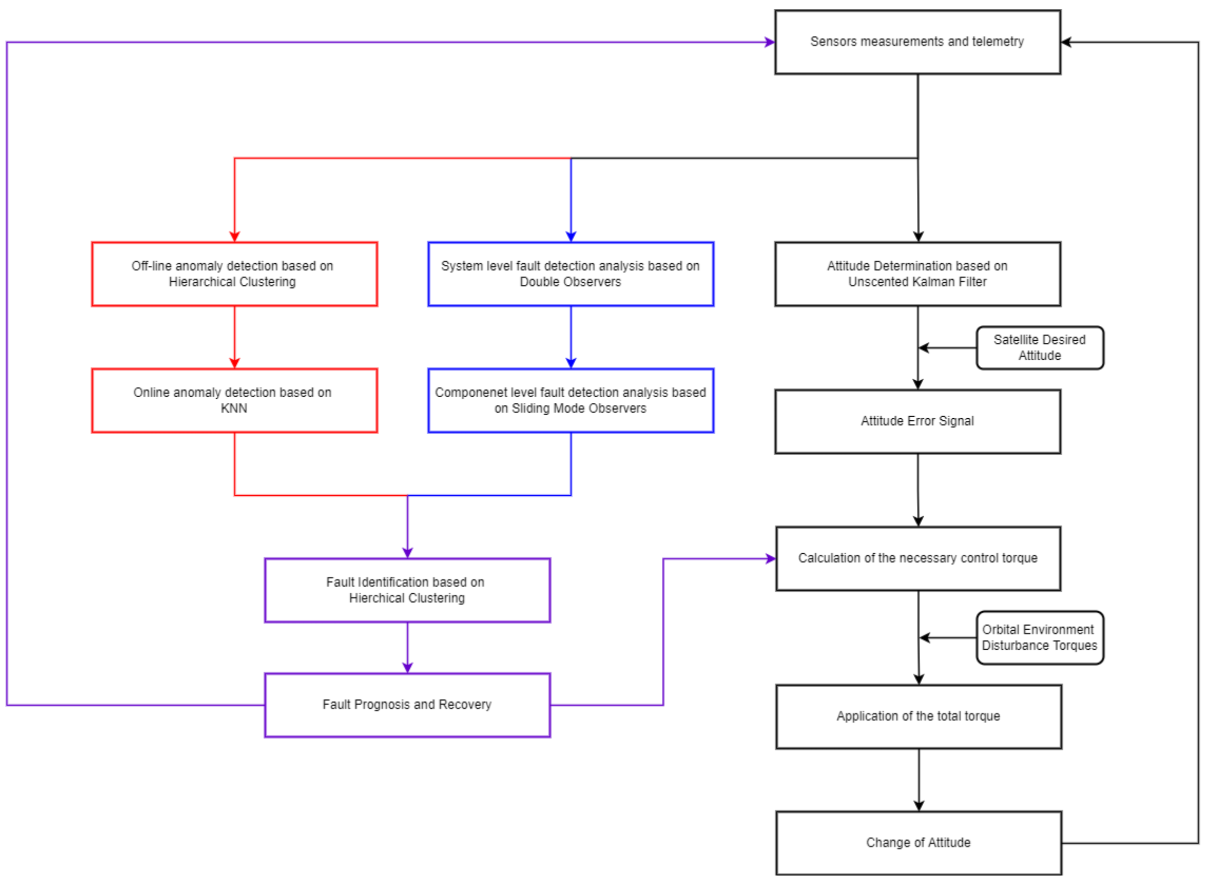

Figure 21 Proposed ADCS framework

Starting from the red path, as it was explained in *3.6*, a data analysis of the sensors' telemetry will be executed online through the KNN algorithm and offline through a Hierarchical Clustering, to detect possible anomalies between the data sets and discard them.

The blue path, instead, will implement two different types of fault detection analysis on two different levels of the satellite, with two different approaches. As showed in *3.7*, the fault detection will be executed on the system level through the Double Observers approach, while on the component level it will be executed based on the Sliding Mode Observers one.

As it is shown in the figure, the two path will merge in a single block, coloured in purple, through which it will be conducted the fault identification based on hierarchical clustering, as it possible to see in *3.8*.

Lastly, there is the Fault Prognosis and Recovery block, branching in two different parts of the framework.

The first one, referring to the Prognosis part, will go into the sensors' measurements and telemetry block, in order to delete the anomalies in data.

The second one will merge into the controller block, for the Recovery part of the block, if the algorithm will detect any faults at the component level, it will be necessary to deactivate the faulty actuator.

This algorithm will therefore monitor the health status of the satellite through its data and prognose possible faults with the Fault Detection and Identification, making the system more autonomous, robust and reliable, while at the same time operating Fault Prognosis and Recovery.

4. Simulation and verification

In this section, a three-axis stabilized satellite ADCS is implemented in MATLAB/Simulink environment to give numerical proof of the effectiveness of the algorithm proposed.

In order to complete the simulation, the implemented ADCS model will be introduced and explained and afterwards the results of the Data Analysis and of the Fault Detection and Identification will be showed and commented.

First, these are the parameters considered for the simulation:

Table 10 Simulation parameters

Satellite parameters	
Mass	720 kg
Dimensions	4.6x2.2x2.34 m
Inertia of the satellite	$I_x = 1269.6$ kg m^2; $I_y = 290.4$ kg m^2; $I_z = 1560$ kg m^2
Orbital angular velocity	0.001 rad/s
Initial attitude (sigma)	[-0.3134 0.3663 0.3619]
Controller parameters	
ρ_i	0.1
δ_i	0.0001

Disturbance torques are also implemented as follow:

$$T_{dx} = A_x \sin \omega_d t \tag{3.91}$$

$$T_{dy} = A_y \sin \omega_d t \tag{3.92}$$

$$T_{dz} = A_z \sin \omega_d t \tag{3.93}$$

Where, $A_x = 1.5 * 10^{-5}\ Nm$, $A_y = 1.6 * 10^{-5}\ Nm$, $A_z = 1.4 * 10^{-5}\ Nm$ and $\omega_d = 0.02\ rad/s$.

4.1 ADCS model

In this model, as it is noticeable from Fig. 22, the desired position is given as an input in order to calculate the signal error from the position measured.

The next steps, as it is explained in the paragraph *3.9*, are the *controller block* commands that become inputs for the *actuator blocks*, which will produce the necessary torque. The result of the controlled torque given by the actuators will be summed at the *Environmental*

Disturbance Torques block that were introduced before, giving as output the total torque that will be computed in the *Dynamics block*.

After this block, the sensors in the Sensors block will result in the sensors measurements and they will be the input for the *Unscented Kalman Filter block* that will give the measured position back, to compute the error with respect to the desired one.

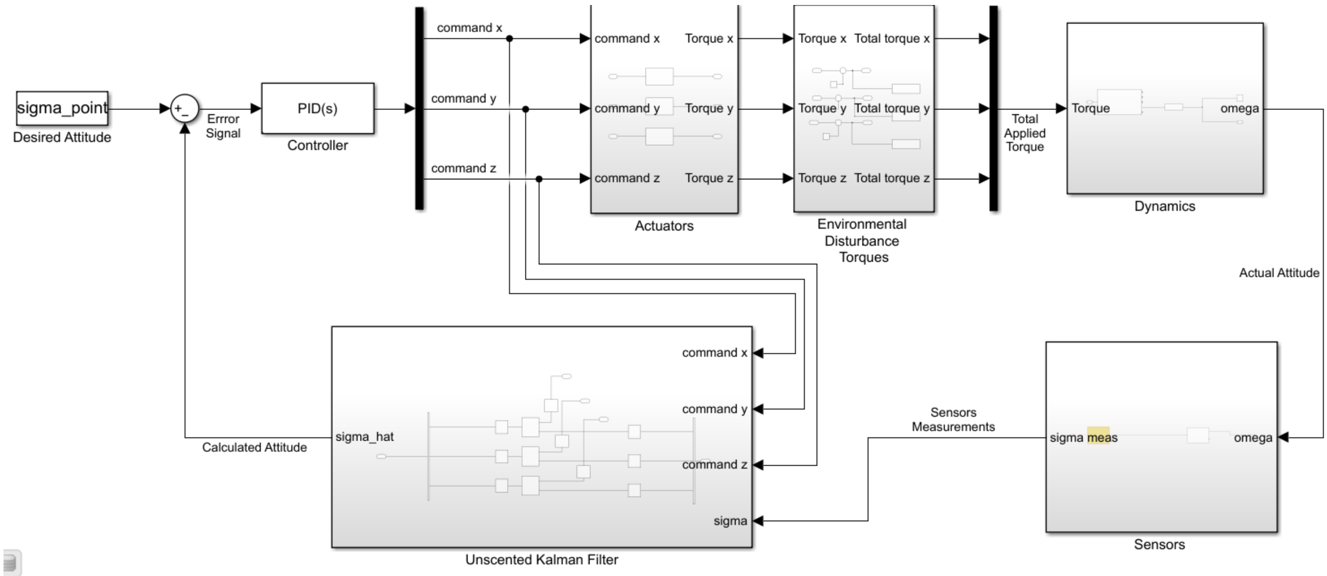

Figure 22 ADCS model in MATLAB\Simulink

4.2 Attitude model

For the desired attitude, the model used is the following:

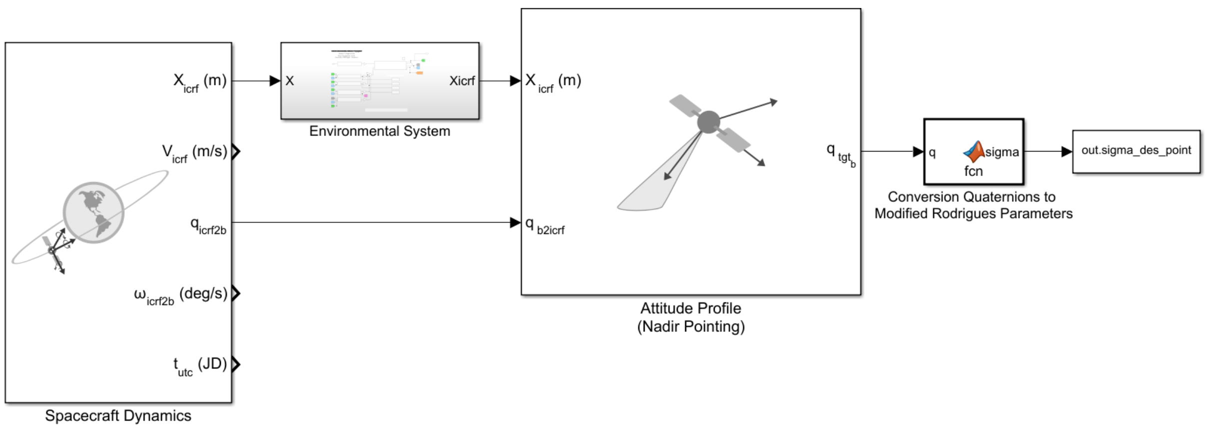

Figure 23 Desired attitude model

Through the *Spacecraft Dynamics Block*, from the Simulink *Aerospace Blockset*, are calculated the ICRF (International Celestial Reference Frame) position and velocity, the quaternions, the angular velocity in body frame and the time UTC in Julian Days.

For this case study, in order to obtain an improved accuracy of the dynamics, it will be implemented the *Environmental Model block* offered by [60], which as it represented in Figure 24 consider the disturbances depending on the position given as input, calculated form the orbit propagator, included in the *Spacecraft Dynamics block*.

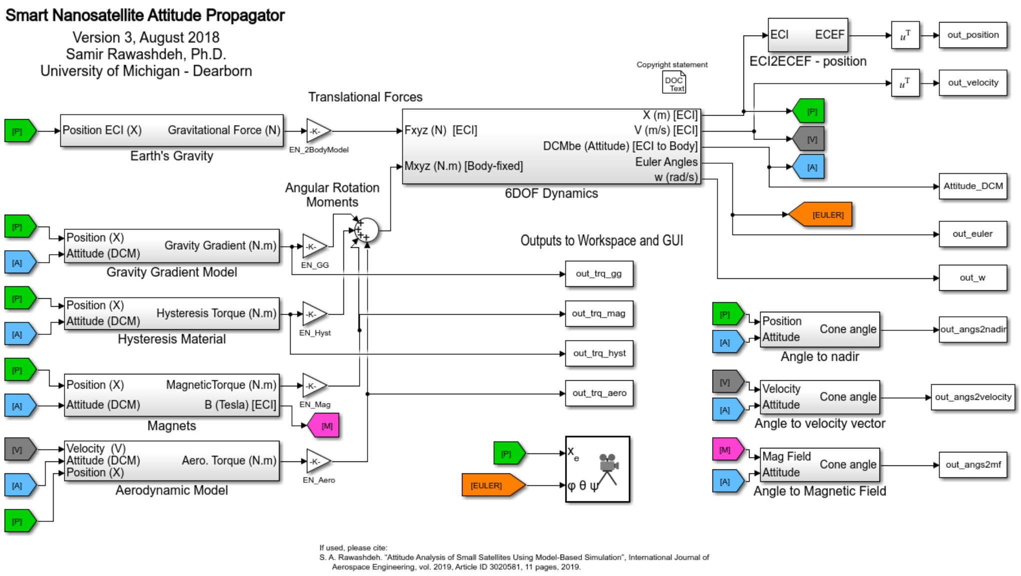

Figure 24 Smart Nanosatellite Attitude Propagator, [60]

After this, the updated position and the quaternions will be the input of the *Attitude Profile block* that will give as output the specific quaternions.

These will be converted in MRPs as it is proposed in this thesis.

4.3 Actuator models

Since the orbital parameters used as reference for the numerical simulation are from the mission CryoSat-2, in this case study only the reaction wheels are represented as actuators.

For ease of calculation are considered only 3 RWs, so one for axis.

The reaction wheel model is implemented in Simscape, as follow:

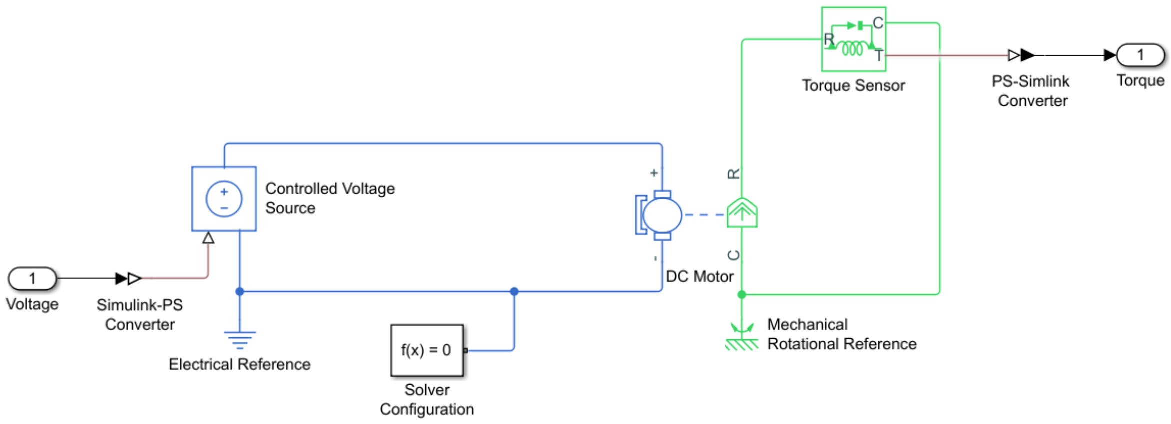

Figure 25 DC Motor model on Simscape

As it clear from the figure, the blue part of the scheme is the electric one, while the green part is the mechanical.

The voltage is given by the controller as the input that is converted to the Simscape environment. After the solver has given the solution to the configuration, the *DC Motor* will give the controlled torque to apply.

The parameters used for the DC motor are:

Table 11 Simulation parameters for DC motor model

Electrical parameters	
Armature resistance	4 Ohm
Armature inductance	$2.75 * 10^{-5}$ H
Back-emf constant	$0.072 * 10^{-3}$ V/rpm
Mechanical parameters	
Rotor inertia	$3.2284 * 10^{-6}$ cm^2*g
Rotor damping	$3.5077 * 10^{-6}$ N*m*s/rad

4.4 Sensors Model

For the Sensors block, it is used a *three-axis gyroscope block* from the *Aerospace Blockset* for measurements and its output is computed in a block that calculate the sigmas from the angular velocities through the *MRP Kinematics block*.

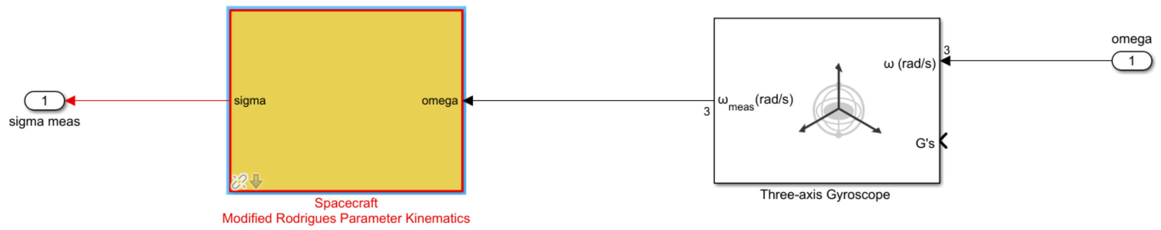

Figure 26 Sensors Model

This specific block is a modified version of the original, referenced as [61].

As it possible to see from the comparison of the two blocks, the one at right, modified for this thesis, presents the implementation of the shadow set too, avoiding singularities problems.

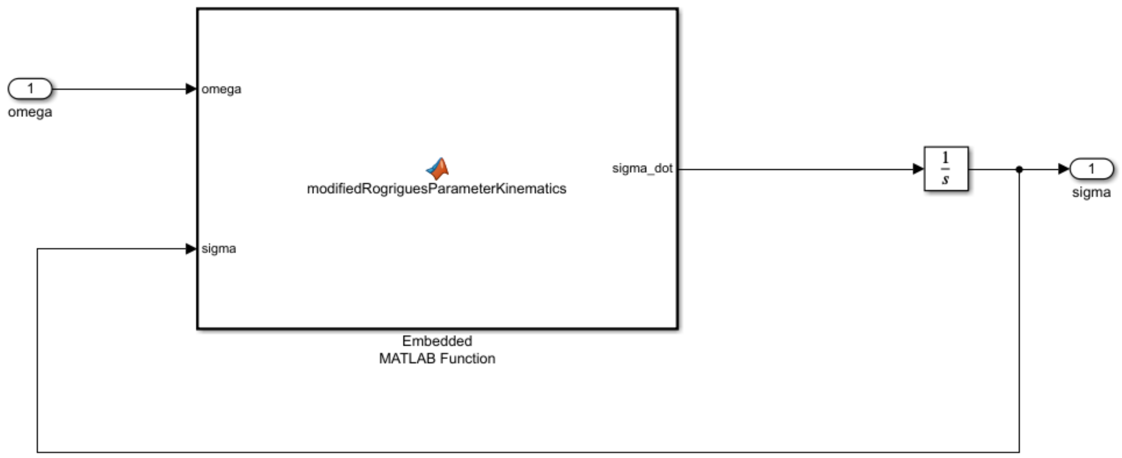

Figure 27 MRP Kinematics original version from [61]

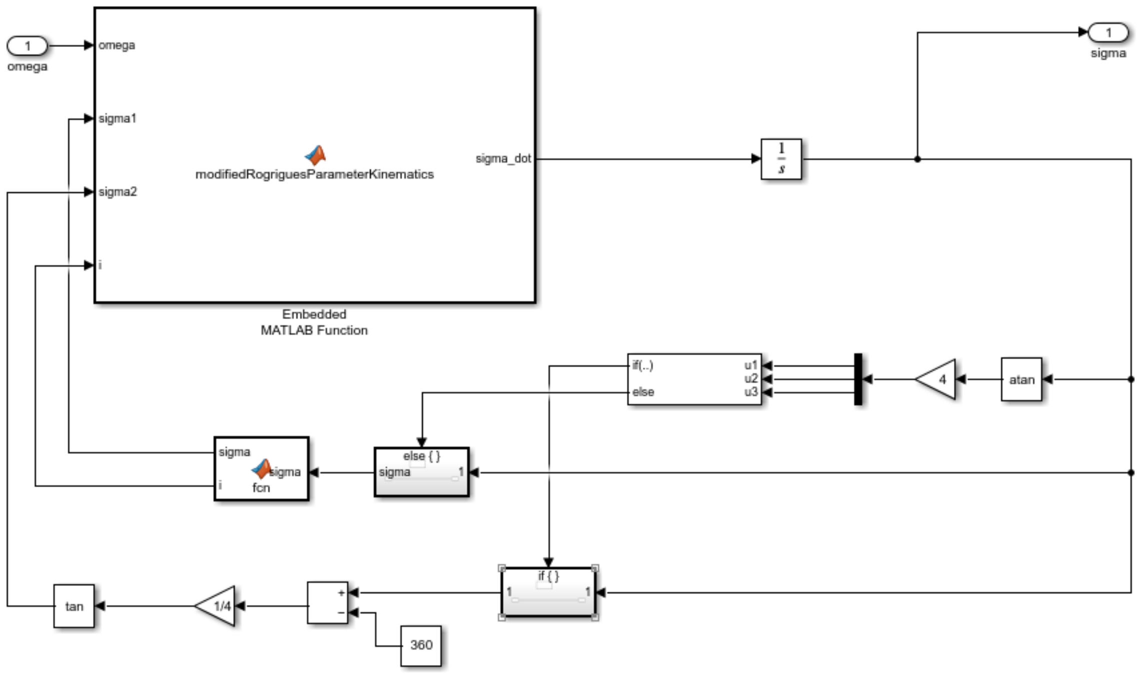

Figure 28 MRP Kinematics update version with shadow set

The choice between the normal or the shadow set is implemented through an *if block* that checks the angle in order to define the range to switch in the other set.

4.5 Fault detection

As explained in the previous chapter, the fault detection will be carried out through Double Observers for the System level and through a bank of three Sliding Mode Observers for the Component level.

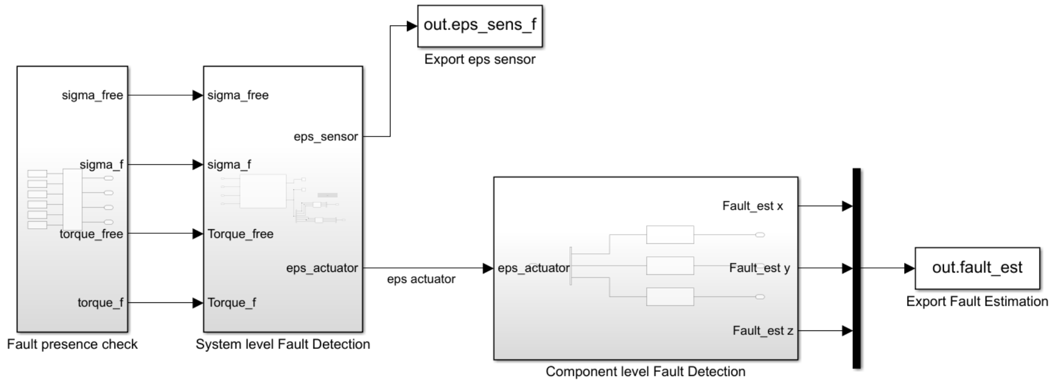

Figure 29 Fault Detection

The *Fault presence check block* is a simple check for the algorithm to give in input the right variables, if there is a fault or not.

Specifically, the *System level Fault Detection block* is composed as follows:

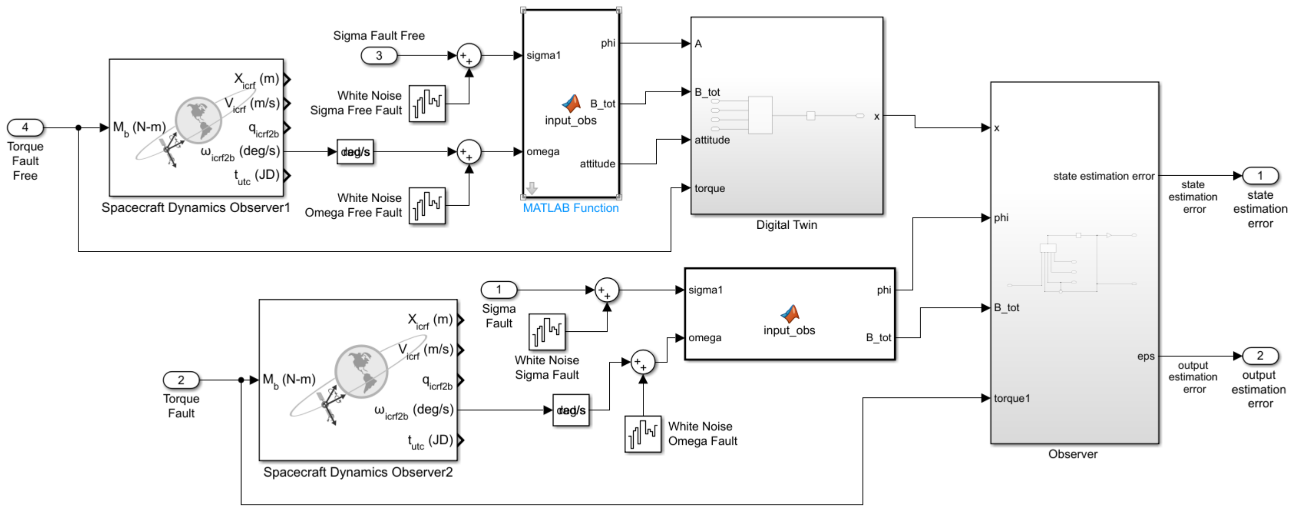

Figure 30 Fault Detection System level

As it is explained in the dedicated paragraph, the Double Observers Fault Detection algorithm is implemented, using the equations and satisfying the theorems imposed with the matrices chosen.

For this simulation, the following threshold will be considered: $\lambda_1 = 3 * 10^{-7}$ $\lambda_2 = 2 * 10^{-7}$.

In order to have an idea of the effectiveness of this algorithm, it is injected a fault in all of three actuators and in all of three sensors.

For the actuators, the fault injected is a pulse type one, with these characteristics:

$$\begin{cases} u_a(t) = u_d(t), \ t < 780s \\ u_a(t) = u_d(t) + f_a(t), \ t \geq 780s \end{cases} \tag{3.94}$$

Where u_a is the actuator output, u_d is the desired actuator output and $f_a(t)$ is the bias fault of the actuators and it is defined as:

$$\begin{cases} f_a(t) = 0, \ t < 780s \\ f_a(t) = 0.003, \ t \geq 780s \end{cases} \tag{3.95}$$

The pulse-type fault of an actuator is common to occur as a control line fault or a bearing failure [62].

At the same time, it is injected a fault in all of the three sensors, another pulse-type, expressed as:

$$\begin{cases} \sigma_{out}(t) = \sigma_d(t), \ t < 780s \\ \sigma_{out}(t) = \sigma_f(t), \ t \geq 780s \end{cases} \tag{3.95}$$

Where $\sigma_{out}(t)$ is the sensor output, $\sigma_d(t)$ is the desired sensor output and $\sigma_f(t) = [0,0,0]^T$ is the output when it faults.This kind of fault indicates that the sensor has failed to detect the correct attitude information.

In Figures 31 and 32, it is possible to observe the difference between the free fault condition and the faulty one:

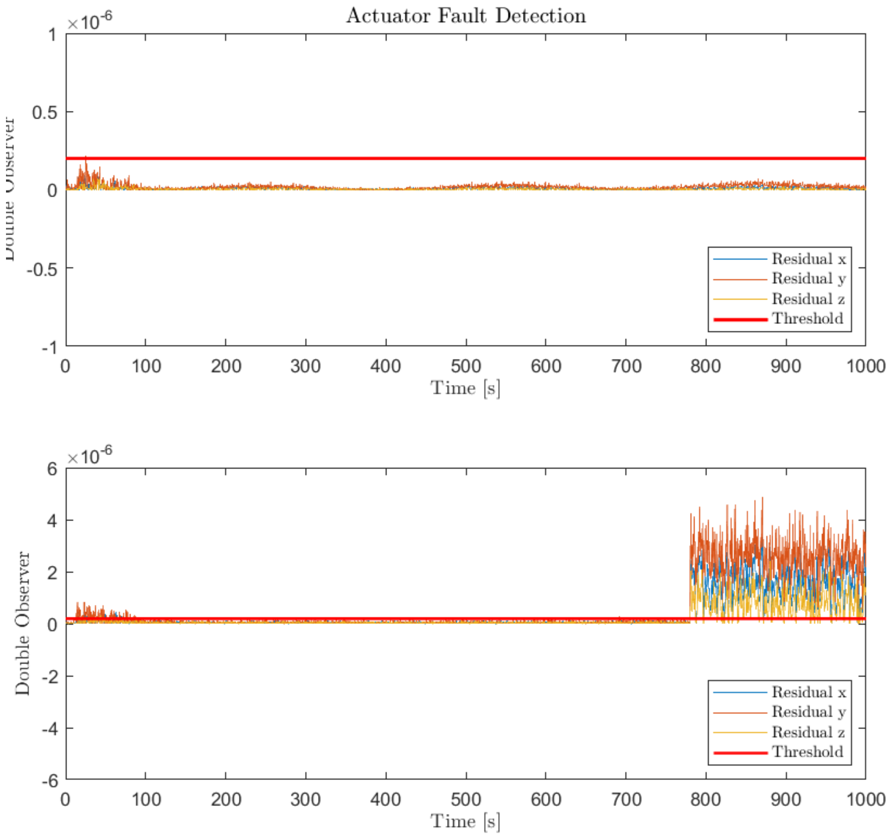

Figure 31 Actuator Fault detection at System level

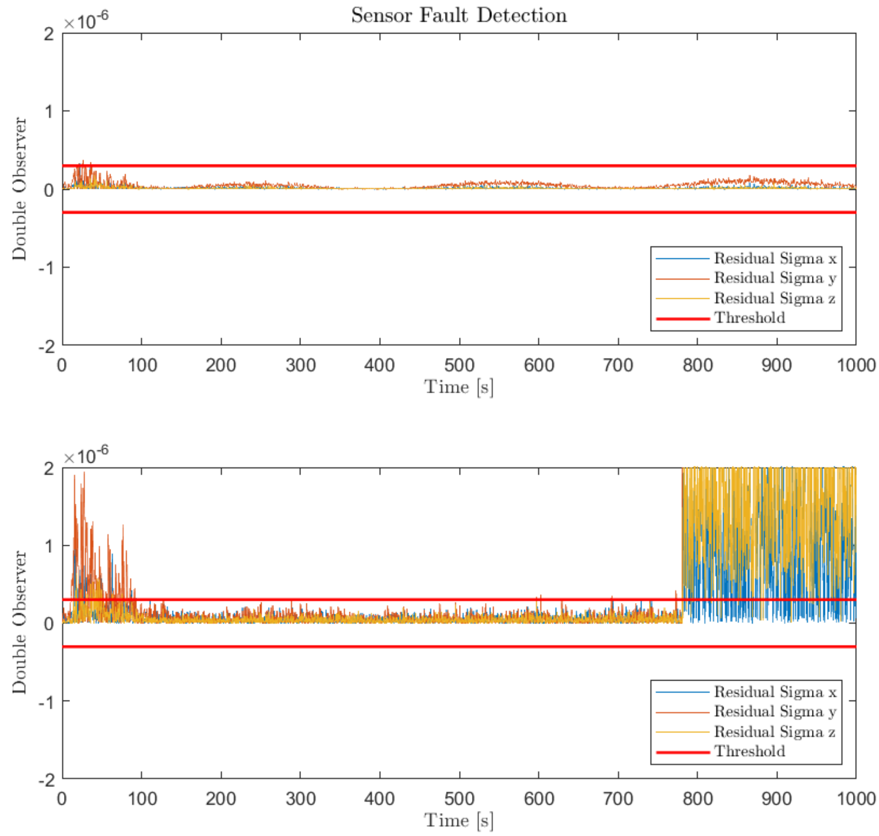

Figure 32 Sensor Fault detection at System level

After the fault is detected in the actuators, the algorithm continues to the *Component level Fault Detection block*, composed as:

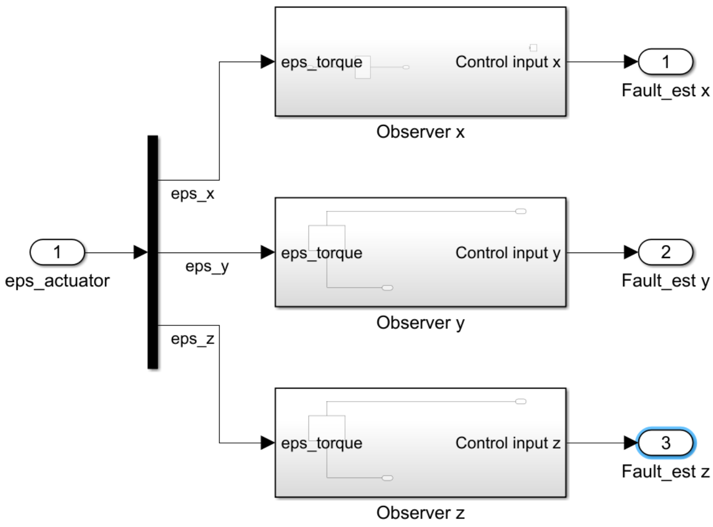

Figure 33 Fault Detection Component level

In this block, the actuator output estimation error, obtained from the previous fault detection, will be the input for the bank of Sliding Mode observers that will compute the difference between the estimation of the fault and the real one.

Figure 34 show the computed comparison between the estimated fault and the real one:

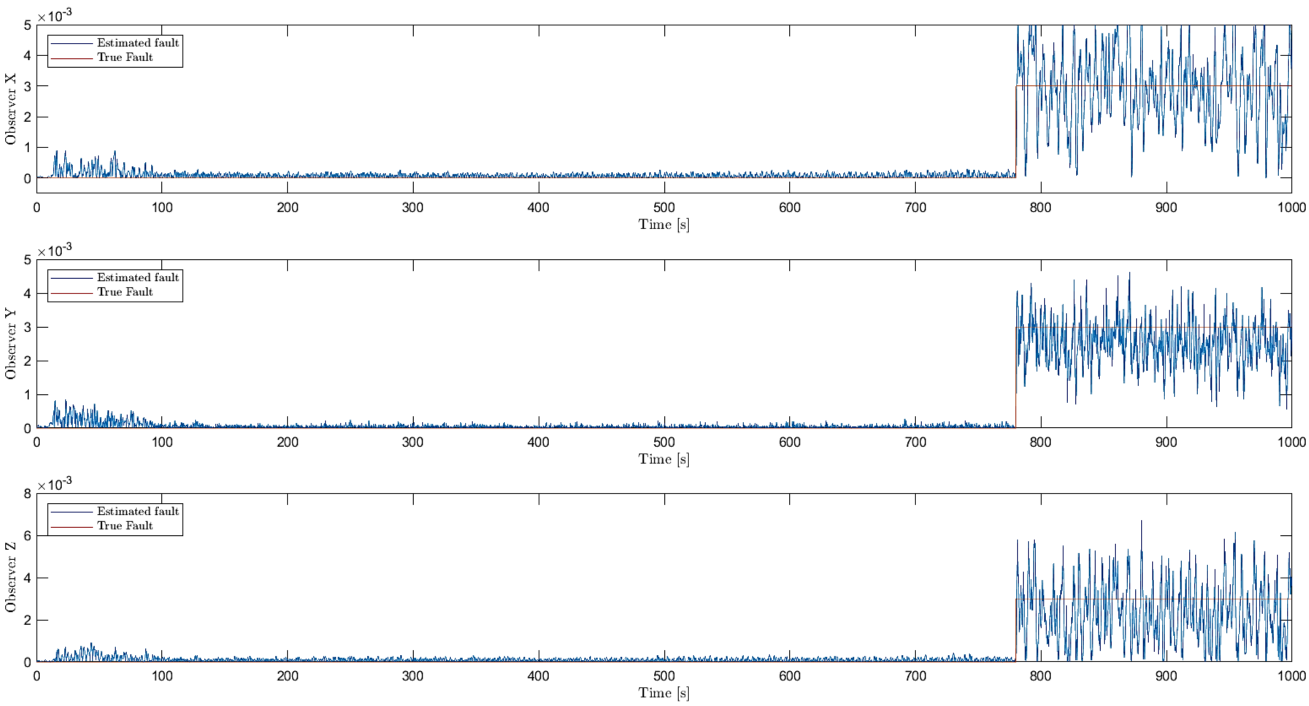

Figure 34 Actuator Fault detection at Component level

4.6 Data Analysis

To execute the data analysis, two different approaches are applied and computed, the KNN algorithm and the hierarchical clustering, in order to find anomalies in data obtained from the sensors.

To compute the KNN, it was necessary to implement a code on MATLAB, using the built-in functions: *pdist* [63], *linkage* [64] and *dendrogram* [65].

Using these functions, it was possible to obtain this important comparison, between the free fault condition and after the fault injection:

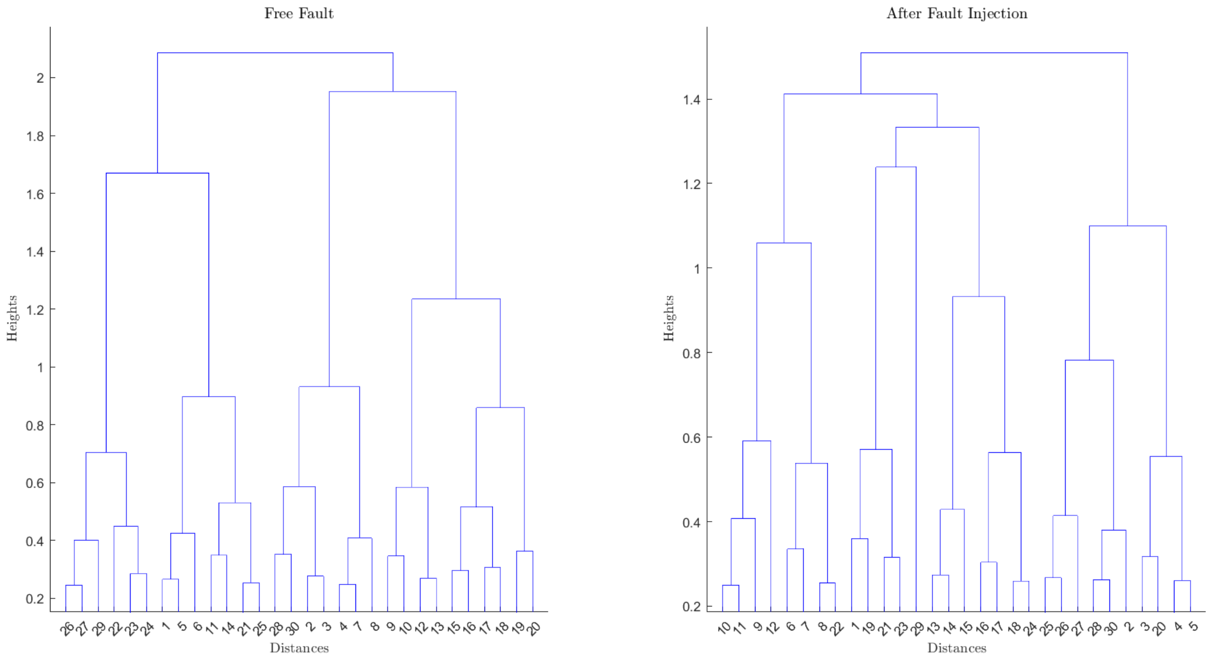

Figure 35 Dendrograms of sensors for KNN

As it possible to notice, the heights between the different clusters are more heterogeneous, indicating the presence of a fault in the sensor or the presence of anomalies in the telemetries.

To prove the effective presence of anomalies, it was computed a graph through a function obtained from [66], which give as output:

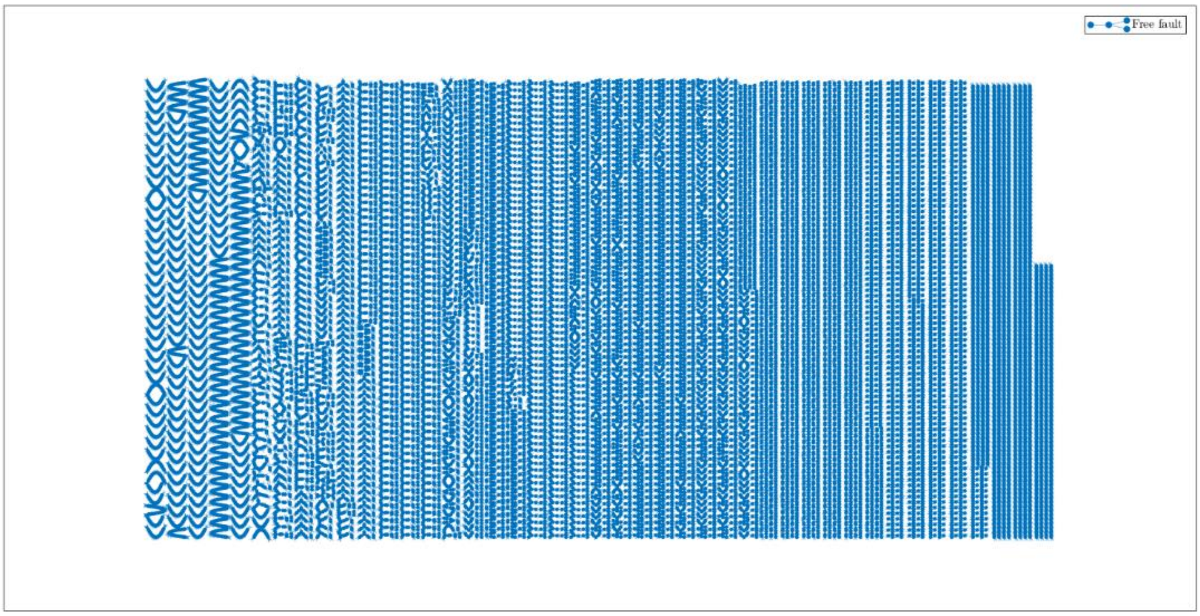

Figure 36 Free fault sensor graph from KNN

This figure represents the free fault configuration showing an organised pattern.

Different from this, it is Figure 37, which show the faulty configuration, such as the one with all the sensor faulty.

In this one, it is clear the presence of a cluster at the bottom of the left side, which show the presence of an anomaly.

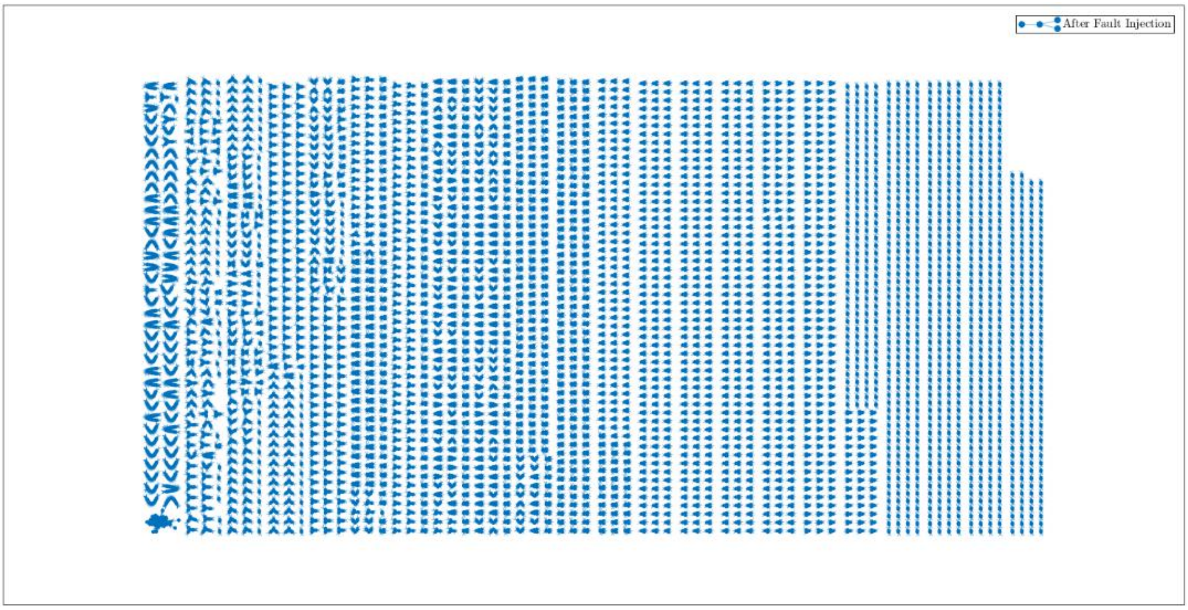

Figure 37 Faulty sensor graph from KNN

The second step, for Data Analysis is executed implementing the hierarchical clustering.

In the next two figures, it will be presented the output of the sensors and the results of the clustering, in a fault free condition and after the injection of the fault.

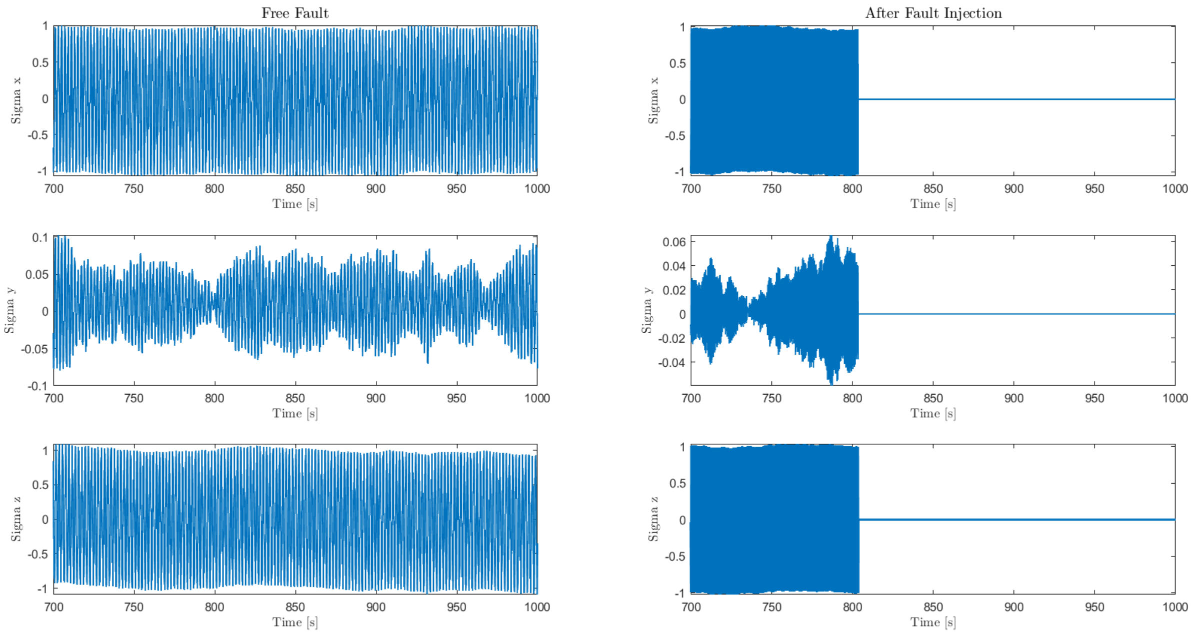

Figure 38 Sigma Measured vs Time

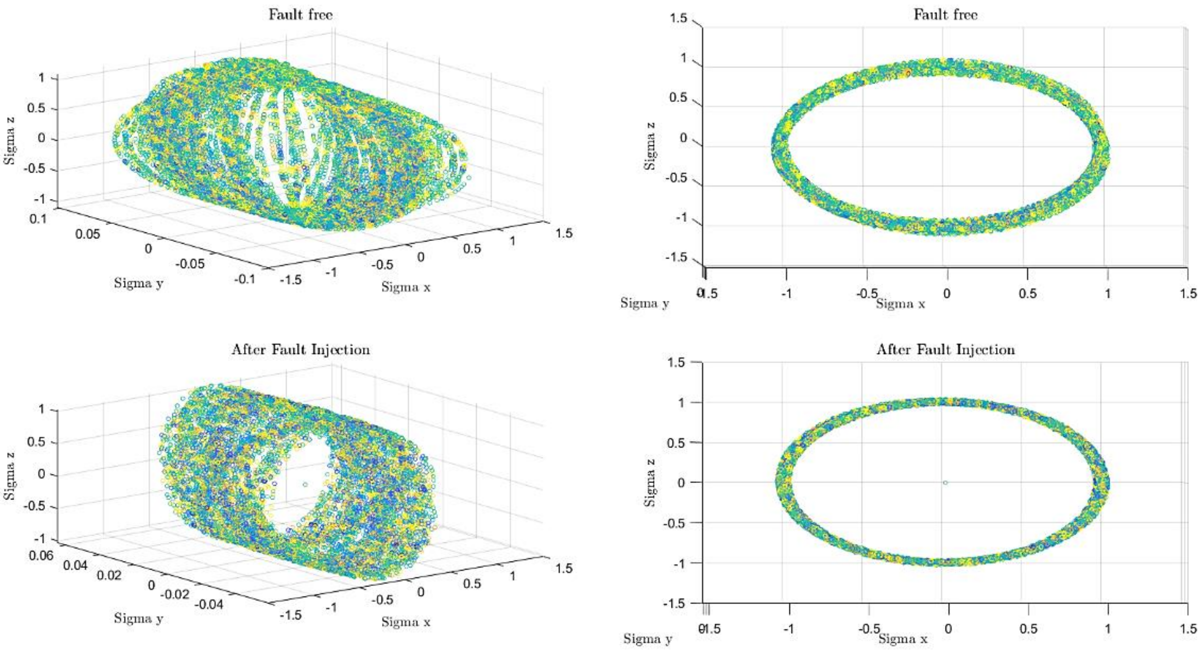

Figure 39 Clustering graphs for Data Anlysis

As it is evident from the comparison of these two figures, the faulty condition presents a cluster in the centre of the axis, indicating a clear anomaly also from the clustering technique.

4.7 Fault Identification

As explained previously, the fault identification will be carried out through a hierarchical clustering.

Analysing the data from the fault detection analysis, it is computed the clustering in two different configurations, in a free fault situation and in a faulty one.

As it highlighted from Figure 40, the fault injected for the fault detection is clearly present, in the dependency of time:

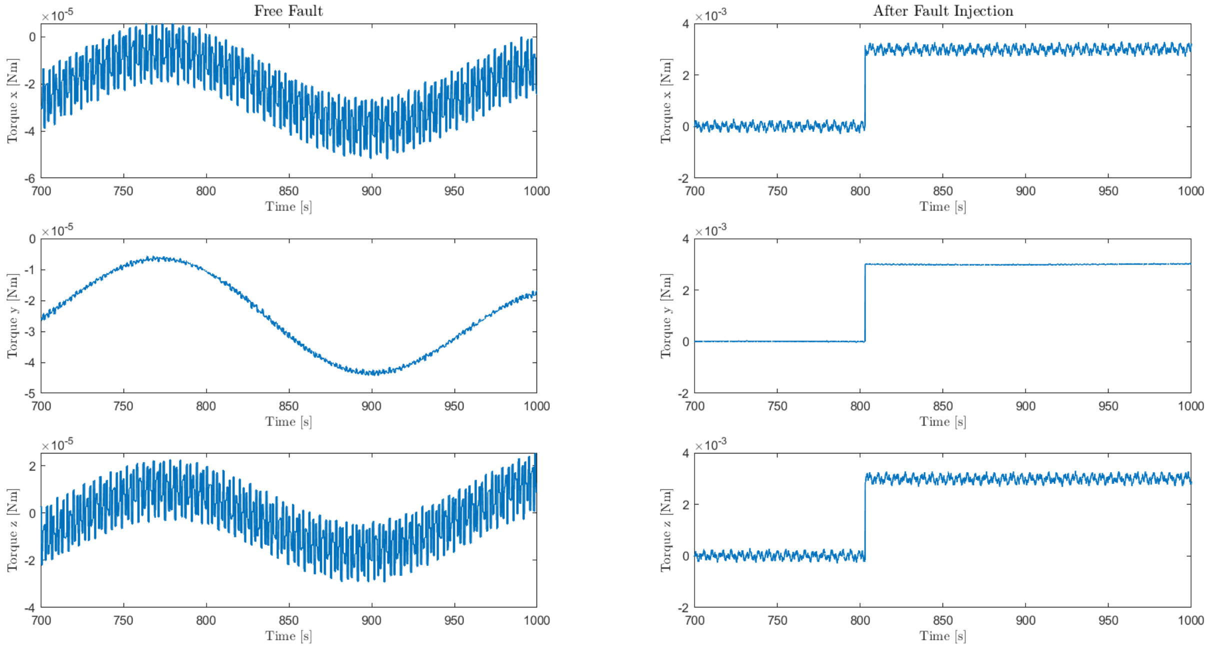

Figure 40 Torque vs Time

To identify the fault, it will be calculated the dendrogram before, and after it will be graphed the clusters. All these figures will be compared to the fault free condition, in order to give a complete understanding:

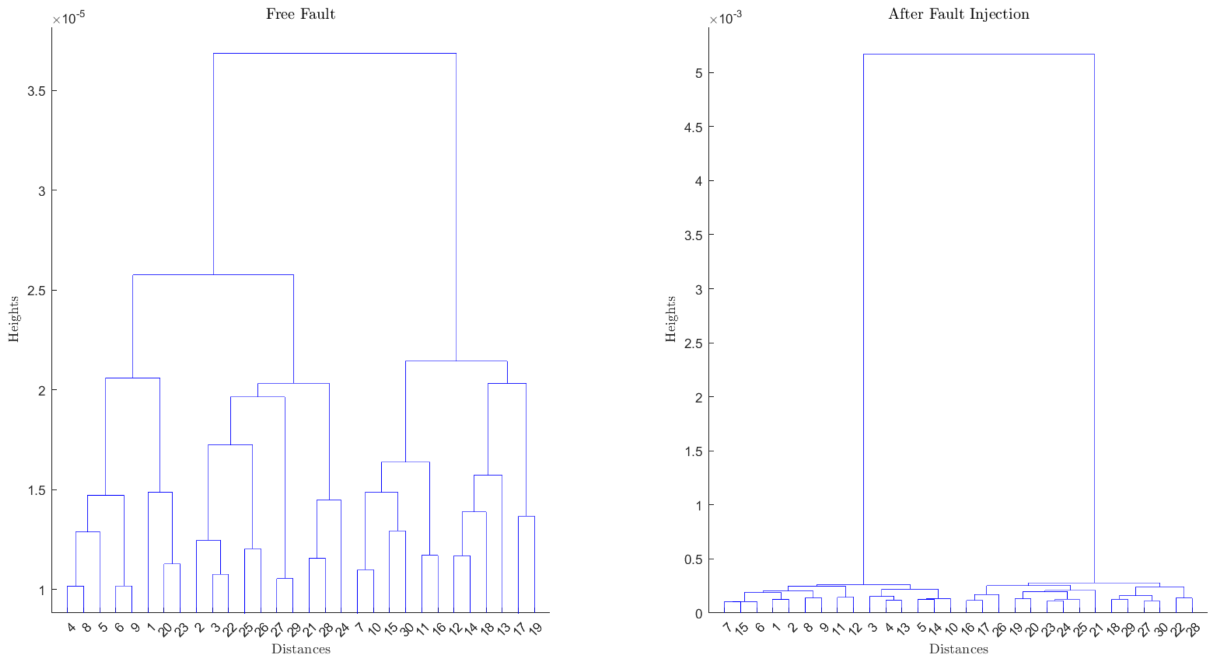

Figure 41 Dendrograms for actuators faults

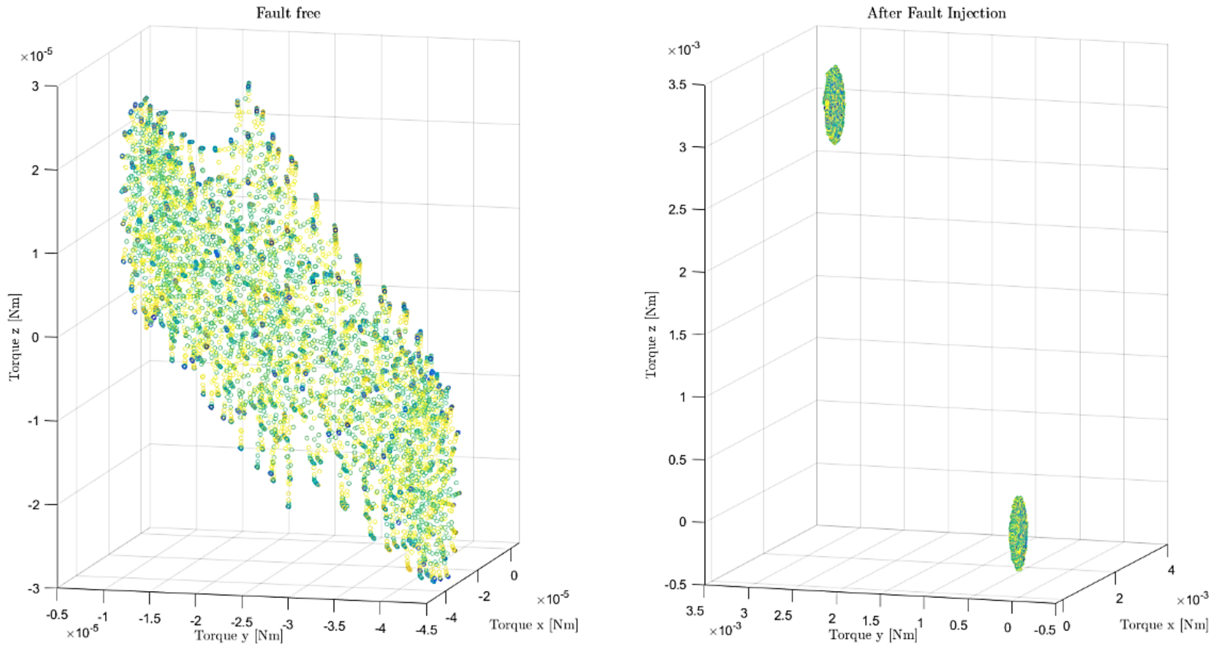

Figure 42 Clustering for Fault Identification

As it represented in Figures 41 and 42, the presence of the fault appears evident as in the dendrogram, where the difference between the heights of the last two clusters is 5 times more, as in the cluster representation, where the two clusters are completely detached.

5. Conclusions

This thesis aimed to develop a viable autonomous algorithm for a Small-Sat, in view of next developments towards trusted autonomy in space.

The proposed approach, based on a combination of physics-based models and Artificial Intelligence (AI) algorithms has satisfied the set objectives, thus providing an effective solution to make the ADCS of a satellite more autonomous.

As substantiated in the presented verification activities, the techniques of fault detection are effective, at both levels, in providing an accurate analysis of the presence of possible faults.

Additionally, the data analysis results help to identify the presence of anomalies in the sensors telemetries. As demonstrated, the anomalies can be detected as errors, following the specific fault detection for the sensors or it can be anomalies of the data, checking the distances provided by the MATLAB code with the abnormal determination parameter.

The strength of the proposed algorithm relies on the ability of the system to become reliable and robust, also in terms of trusted autonomy. As presented and demonstrated by numerical simulations, this approach provides:

- A real-time analysis of data that helps to forecast the possible degradation of the components, analysing the patterns obtained from the KNN and comparing with the healthy one. At the same time, these results allow to eventually prognose the onset of faults, activating the fault detection, identification and recovery mechanism,
- Another scientific reliable proof that relying on a Digital Twin version of the CubeSat offers a valid opportunity of a more accurate representation of the models, being closer to reality and achieving a better estimation of the behaviour of the systems.

As a consequence of these achievements, this work aim is to propose an innovative point of view on the trusted autonomy field, showing how it could be possible to design and implement an efficient and reliable algorithm to a satellite system, obtaining promising results in prognosis of anomalies in data, before the actual degradation of the components or the occurrence of faults.

In this scenario, the satellite does not necessarily need to rely on a ground station, for example, being able to preserve and monitor its health status continuously, becoming reliable in its own autonomous functioning.

Based on these conclusions future research should focus on:

- Further improve the accuracy of physics models, as applicable/required,
- Extending the analysis to additional kinds of faults for the specific actuators and sensors,
- Conducting experimental flight testing and/or adopting data from real missions, to prove the effectiveness of the algorithm in real operations.

This list is obviously not complete, but its aim is to be a first direction for future investigation.

References

[1] *A brief story of space exploration*, https://aerospace.org/article/brief-history-space-exploration, **2018**

[2] Kalam, A. P. J., *The future of space exploration and human development*, The Frederick S. Pardee Center for the Study of the Longer-Range Future Boston University, **2008**

[3] *https://apod.nasa.gov/apod/astropix.html*

[4] *What are SmallSats and CubeSats?*, https://www.nasa.gov/content/what-are-smallsats-and-cubesats, **2015**

[5] Wayer, J. K., Castet, J.-F., Saleh*,* J. H., *Spacecraft attitude control subsystem: Reliability, multi-state analyses, and comparative failure behavior in Leo and Geo*, Acta Astronautica, 85-92, **2013**

[6] Elkins, J., Sood, R., Rumpf, C., *Autonomous Spacecraft Attitude Control Using Deep Reinforcement Learning*, 71st International Astronautical Congress (IAC), **2020**

[7] Lee, K-H., Lim S., Cho, D-H., Kim, H-D., *Development of Fault Detection and Identification Algorithm Using Deep learning for Nanosatellite Attitude Control System*, International Journal of Aeronautical and Space Sciences, **2019**

[8] NASA Autonomous Systems – Systems Capability Leadership Team, *Autonomous systems taxonomy*, NASA Technical Reports Server, https://ntrs.nasa.gov/citations/20180003082, **2018**

[9] Toupet, O., Del Sesto, T., Ono, M., Myint, S., vander Hook, J., and McHenry, M., *A Ros-Based Simulator for Testing the Enhanced Autonomous Navigation of the mars 2020 Rover*., IEEE Aerospace Conference, 1–11. **2020**

[10] Coppin, G., Legras, F., *Autonomy Spectrum and Performance Perception Issues in Swarm Supervisory Control*, Proceedings of the IEEE. 100. 590-603, **2012**

[11] Ranasinghe, K., Kapoor, R., Gardi, A., Sabatini, R., S., Wickramanayake, V., Ludovici, D., *Vehicular Sensor Network and Data Analytics for a Health and Usage Management System*, *Progress in Aerospace Sciences*, Sensors, **2020**

[12] Ranasinghe, K., Sabatini, R., Gardi, A., Bijjahalli, S., Kapoor, R., Fahey, T., Thangavel, K., *Advances in Integrated System Health Management for mission-essential and safety-critical aerospace applications*, Progress in Aerospace Sciences, Volume 128, **2022**

[13] *Trusted Autonomy for the Moon and Mars*, https://www.nasa.gov/offices/oct/feature/Trusted_Autonomy_for_the_Moon_and_Mars, **2019**

[14] Venkatesh, M., Kumar, K.D., *A novel data-driven method for fault detection and isolation of control moment gyroscopes onboard satellites*, Acta Astronautica, **2021**

[15] Guo, X.-G., Li, Q., Ahn, C.-K., Yang, Y.-H., *Multiple-fault Diagnosis for Spacecraft Attitude Control Systems using RBFNN-based Observers*, Aerospace Science and Technology, **2020**

[16] Parrott, A., Warshaw, L., *Industry 4.0 and the digital twin*, https://www2.deloitte.com/us/en/insights/focus/industry-4-0/digital-twin-technology-smart-factory.html, **2017**

[17] B. H. Kawauchi, *Redundancy Management for Autonomous Spacecraft Attitude Control Systems*, 1986 American Control Conference, pp. 497-506, **1986**

[18] Alvelda, P., San Martin, A., *Neural Network Star Pattern Recognition for Spacecraft Attitude Determination and Control*, Advances in Neural Information Processing Systems 1, **1988**

[19] van Bezooijen, R.W.H., *A Star Pattern Recognition Algorithm for Autonomous Attitude Determination*, IFAC Proceedings Volumes, Volume 22, Issue 7, **1989**

[20] Toms, D., Hadden, G.D., Harrington, J., *Attitude Determination and Control System (ADCS) Maintenance and Diagnostic System (MDS)*, NASA Conference Publication, Edition 3073-3075, **1990**

[21] Bak, T., Wisniewski, R., Blanke, M., *Autonomous attitude determination and control system for the Orsted satellite*, 1996 IEEE Aerospace Applications Conference. Proceedings, **1996**

[22] Powell, G. Jr., Lane, M., Indseth, R., *A layered architecture for autonomous satellite command and control*, AIAA 1998-4384. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, **1998**

[23] Baroth, E., Powers, W., Fox, J., Prosser, B., Pallix, J., Schweikard, K., Zakrajsek, J., *IVHM (Integrated Vehicle Health Management) techniques for future space vehicles*, AIAA 2001-3523. 37th Joint Propulsion Conference and Exhibit, **2001**

[24] Hess, A., Calvello, G., Frith, P., *Challenges, issues, and lessons learned chasing the "Big P". Real predictive prognostics. Part 1*, 2005 IEEE Aerospace Conference, **2005**

[25] Hess, A., Calvello, G., Frith, P., Engel, S. J., Hoitsma, D., *Challenges, Issues, and Lessons Learned Chasing the "Big P": Real Predictive Prognostics Part 2*, 2006 IEEE Aerospace Conference, **2006**

[26] P.M. Frank, *Analytical and Qualitative Model-Based Fault Diagnosis – A survey and Some New Results*, European Journal of Control, **1996**

[27] Jiang, T., Khorasani, K., *A fault detection, isolation and reconstruction strategy for a satellite's attitude control subsystem with redundant reaction wheels*, 2007 IEEE International Conference on Systems, Man and Cybernetics, **2007**

[28] Nasrolah, S. S., Abdollahi, F., *Sensor fault detection and recovery in satellite attitude control*, Acta Astronautica, **2018**

[29] Liang, X., Wang, Q., Hu, C., Dong, C., *Observer-based H∞ fault-tolerant attitude control for satellite with actuator and sensor faults*, Aerospace Science and Technology, **2019**

[30] Nemati, F., Safavi, S. M., Zemouche, A., *A nonlinear observer-based approach to fault detection, isolation and estimation for satellite formation flight application*, Automatica, **2019**

[31] Gao, S., Zhang, Z., Zhang, W., He, X., Lu, X., *Observer-based multiple faults diagnosis scheme for satellite attitude control system*, IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, **2017**

[32] Guo, X.-G., Tian, M.-E., Li, Q., Ahn, C.-K., Yang, Y.-H., *Multiple-fault Diagnosis for Spacecraft Attitude Control Systems using RBFNN-based Observers*, Aerospace Science and Technology, **2020**

[33] Gao, S., Zhang, Z., Zhang, W., He, X., Lu, X., *Fault Diagnosis for Satellite Attitude Control System with Using Extended Kalman Filter,* Proceedings of the 38th Chinese Control Conference, **2019**

[34] Pourtakdoust, S.H., Mehrjardi, M.F., Hajkarim, M.H., *Attitude estimation and control based on modified unscented Kalman filter for gyro-less satellite with faulty sensors*, Acta Astronautica, **2022**

[35] Lee, D., Park, H., Romano, M., Cutler, J., *Development and experimental validation of a multi-algorithmic hybrid attitude determination and control system for a small satellite*, Aerospace Science and Technology, **2018**

[36] Lakshmi, V. Jaya, Ramesh, M.V., *Diagnosis methods of stator short circuit faults of BLDC Motor Using Adaptive Neuro-fuzzy Inference (ANFIS) system and Wavelet*, IOSR Journal of Electrical and Electronics Engineering, **2015**

[37] Rothwell, E., Notes from *Electromagnetic compatibility course*, Michigan State University, **2016**

[38] Arnoux, J-J.P., *Star sensor baffle optimization: some helpful practical design rules*, Proc. SPIE 2864, Optical System Contamination V, and Stray Light and System Optimization, **1996**

[39] Bellini, N., Tortora, P., Puig-Suari, J., *Magnetic Actuators for Nanosatellite Attitude Control*, Alma Mater Studiorum, Universita' Di Bologna, Scuola Di Ingegneria E Architettura, Sede di Forlì, pp. 11-16, 68-69, 99-103, **2013/2014**

[40] European Space Agency, *CryoSat Mission and Data Description,* Doc. No. CS-RP-ESA-SY-0059, **2007**

[41] Landis Markley, F., *Spacecraft Attitude Representations*, Aatborg University, Dept. of Control Eng. Colloquium, **1999**

[42] Tsiotras, P., *Stabilization and Optimality Results for the Attitude Control Problem*, Journal of Guidance, Control, and Dynamics, Volume 19, **1996**

[43] Schaub, H., Junkins, J. L., *Stereographic orientation parameters for attitude dynamics: A generalization of the rodrigues parameters*, Journal of the 425 Astronautical Sciences 44 (1) **1996**

[44] Konatowski, S., Kaniewski, P., Matuszewski, J., *Comparison of Estimation Accuracy of EKF, UKF and PF Filters*, Annual of Navigation, **2016**

[45] Sidi, M.J., *Spacecraft Dynamics and Control: a Practical Engineering Approach*, vol 7, Cambridge university press, **1997**

[46] Bolgár, D., Biniakos, N., Nicolae, A.-C., Larsen, J. A., *Fault Tolerant Attitude Control of a Pico-Satellite Equipped with Reaction Wheels and Magnetorquers*, Project Technical Faculty of IT and Design Department of Electronics Systems Control and Automation, Aalborg University, **2018**

[47] Tuchin, M., Biryukov, A., Nickiforov, Prokhorov, M., Zakharov, A., *On Random and Systematic Errors of a Star Tracker*, Sternberg Astronomical Institute of Lomonosov Moscow State University, **2013**

[48] Lhermitte, S., Verbesselt, J., Verstraeten, W. W., Coppin, P., *A Comparison of Time Series Similarity Measures for Classification and Change Detection of Ecosystem Dynamics*, Remote Sensing of Environment, Vol. 115, No. 12, pp. 3129-3152, **2011**

[49] Liu, D., Pang, J., Xu, B., Liu, Z., Zhou, J., Zhang, G., *Satellite Telemetry Data Anomaly Detection with Hybrid Similarity Measures*, International Conference on Sensing, Diagnostics, Prognostics, and Control, **2017**

[50] Keogh, E., Pazzani, M., *Derivative dynamic time warping*, Proceedings of the 1st SIAM International Conference on Data Mining, San Jose, CA, USA, **2001**

[51] Wang, R., Chengn, Y., Xu, M., *Analytical redundancy-based fault diagnosis scheme for satellite attitude control systems*, Journal of the Franklin Institute, Volume 352, Issue 5, **2015**

[52] Sabatini, R., Moore, T., Ramasamy, S., *Global navigation satellite systems performance analysis and augmentation strategies in aviation*, Progress in Aerospace Sciences, Volume 95, **2017**

[53] Wu, L.N., Zhang, Y.C., Zhao, S.L., *Sliding mode observer used in satellite attitude control system fault diagnosis*, Journal of Harbin Institute of Technology, **2011**

[54] Murtagh, F., Contreras, P., *Algorithms for hierarchical clustering: An overview*, Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery., **2012**

[55] Zhang, M., Hu, B., Li, K., *A Non-singular Attitude Filtering Based On UKF Using Modified Rodrigues Parameters*, 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), **2018**

[56] https://www.nasa.gov/nesc/technicalupdates

[57] Poghosyan, A., Golkar, A., *CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions*, Progress in Aerospace Sciences, vol. 88, **2017**

[58] https://earth.esa.int/eogateway

[59] http://database.eohandbook.com/database/missionsummary.aspx?missionID=529

[60] S. A. Rawashdeh. *Attitude Analysis of Small Satellites Using Model-Based Simulation*, International Journal of Aerospace Engineering, **2019**

[61] Acquatella, P., *Spacecraft Attitude Kinematics in terms of Modified Rodrigues Parameters library block*, MATLAB Central File Exchange, **2022**

[62] Wang, R., Cheng, Y., Xu, M., *Analytical redundancy based fault diagnosis scheme for satellite attitude control systems*, J. Frankl. Inst., **2015**

[63] https://it.mathworks.com/help/stats/pdist.html

[64] https://it.mathworks.com/help/stats/linkage.html

[65] https://it.mathworks.com/help/stats/dendrogram.html

[66] Vannoy, T., https://github.com/tvannoy/knn-graphs/releases/tag/v1.0, GitHub, **2022**