Democratic and Popular Republic of Algeria

Ministry of Higher Education and Scientific Research

University Kasdi Merbah, Ouargla

Faculty of new information and communication technologies

Department of Electronics and Telecommunications

Dissertation submitted to the Department of Electronics and Telecommunications in Candidacy for the Degree of ***Doctor*** 3rd Cycle LMD

In: Automatics

Speciality: Automatics & Industrial Computing

By: Hicham HENNA

Title

Attitude fault-tolerant control applied to microsatellite

Presented publicly on Apr 20th, 2024, in front of jury members:

Mr.	Boubakeur ROUABAH	MCA	UKM Ouargla	President
Mr.	Houari TOUBAKH	MCA	UKM Ouargla	Supervisor
Mr.	Mohamed Redouane KAFI	Professor	UKM Ouargla	Co-Supervisor
Mr.	Mohamed Abdelbasset MAHBOUB	MCA	UKM Ouargla	Examiner
Mr.	Labib Sadek TERRISSA	Professor	UMK Biskra	Examiner
Mr.	Zakaria LAMMOUCHI	MCA	UHL ElOued	Examiner
Mr.	Noureddine ZERHOUNI	Professor	femto-st Besançon	Invited

Academic Year: 2023 / 2024

Attitude fault-tolerant control applied to microsatellite

Hicham Henna

Supervised by Dr. Toubakh Houari

Co-supervised by Dr. Kafi Mohamed Redouane

Department of electronics and telecommunications

Faculty of new technologies of information and communication

Electrical engineering laboratory

Kasdi Merbah University, Ouargla

April, 2024

A dissertation submitted in partial fulfilment of the requirements for the degree of Ph.d. in automatics and industrial computing.

https://www.univ-ouargla.dz/index.php/en/

First edition, April 30, 2024

Declaration by Postgraduate Students

(a) Authenticity of Dissertation

I affirm that I am the rightful author of this Dissertation and that it is a product of my own intellectual endeavors and all the assistance received in preparing this thesis have been acknowledged.
None of the content within this work has been utilized in support of an application for another degree or qualification from any university or institution of higher education.
I absolve Kasdi Merbah University (Ouargla) from any claims by third parties related to copyright infringement, breach of confidentiality, defamation, or any other infringement of third-party rights.

(b) Research Code of Practice and Ethics Review Procedures

I confirm my adherence to the University's Research Ethics Review Procedures.
In my capacity as a Ph.D. student, I acknowledge and agree that my dissertation will be accessible to the public through the Kasdi Merbah University Institutional Repository.

Faculty/Institute/Centre/School	Faculty of new technologies of information and communication
Degree	Ph.d. in automatics and industrial computing
Title	Attitude fault-tolerant control applied to microsatellite
Candidate (Id.)	Hicham Henna (19/FNTIC-AII/02)
Date	April 30, 2024

I dedicate this dissertation to my mother, my wife Assia, and my cherished children, who have consistently served as my boundless wellsprings of inspiration, wisdom, and affection. Additionally, dedications extend to my brothers, sister, and their respective families.

Acknowledgements

Above all, I extend my gratitude and appreciation to the Almighty **Allah**, the Most Gracious and Most Merciful, who has bestowed upon me numerous blessings and provided the opportunity to undertake and complete this research study, enabling me to pursue my aspirations. Without God's facilitation, none of this would have been possible.

I wish to express my utmost gratitude to my supervisor, ***Dr. Toubakh Houari***, for granting me the opportunity to successfully complete this PhD research. His expertise and guidance have been indispensable, and without them, the accomplishment of this research would not have been possible. I am also thankful to my co-supervisor, ***Dr. Kafi Mohamed Redouane***, for his invaluable assistance, feedback, and insightful comments.

I am eager to convey my appreciation to a pivotal figure in this research, ***Prof. Sayed-Mouchaweh Moamar***. His invaluable comments and support have been instrumental, and without them, completing this dissertation would undoubtedly have been a challenging endeavour.

I would like to seize this opportunity to express my gratitude to ***Prof. Djemai Mohamed*** for his valuable comments. Additionally, I extend my thanks to ***Chikouche Abdelwahab*** for his encouragement, assistance, and responsiveness to all administrative matters.

I am deeply thankful to numerous colleagues at Kasdi Merbah University and the Algerian Space Agency who provided technical and personal advice, offering crucial support throughout my studies. Mere words cannot adequately convey my appreciation to each of you. Special acknowledgement goes to The ***Algerian Space Agency*** for their financial and logistical support, which greatly contributed to the success of this research.

I would also like to recognize the unwavering support of my mother, whose prayers have been an essential part of my achievements. Lastly, this insightful journey would not have been possible without the love, dedication, and sacrifices of my wife. She has been an enthusiastic cheerleader, offering continuous encouragement, and support. My heartfelt thanks also extend to my beloved children: Tahar, Mohamed, Amna, Idris, and Sara, who have been an endless source of love, motivation, and patience.

Abstract

Spacecraft play a pivotal role in various aspects of our daily lives, including mapping, disaster monitoring, and telecommunications. However, akin to any technical apparatus or industrial system, they are susceptible to faults and failures throughout their operational life. Moreover, the repair of damaged components is a rare option in certain missions, such as those involving the Hubble telescope and the International Space Station.

To address these challenges, researchers have embraced two key paradigms: fault diagnosis and fault-tolerant control. Both rely on the availability of a physical model that comprehensively captures system dynamics. Our research in this thesis revolves around two primary axes. Firstly, we employ a hybrid approach, combining a model-based method (Kalman filter) with a data-driven method to enhance gyro fault assessment. This is followed by the reconfiguration of satellite attitude control parameters. Secondly, we delve into the application of reinforcement learning, a cutting-edge artificial intelligence method, to optimize attitude fault-tolerant control.

It is noteworthy that our scientific contribution, particularly in the second aspect, focuses on refining the reward function by incorporating the similarity between the torques generated by the reinforcement learning agent and the conventional control system. Simulation results underscore the efficacy of the proposed methods in this thesis, substantiated through comparisons with the latest advancements documented in scientific publications.

Keywords: Fault-tolerant control, attitude control, reinforcement learning, data-driven methods, fault diagnosis.

ملخص

تلعب الاقمـــار الاصطناعية دورا هاماً في حياتنا اليومية في عدة مجالات كتصوير الخرائط، متابعة الكوارث، والاتصالات. لكنها مثل أي جهاز تقني او نظام صناعي قد تتعرض للأعطاب في مرحلة ما من حياتها الوظيفية. يضاف الى ذلك أن تصليح الاعطاب غير متاح موضعيا الا نادرا في بعض المهمات مثل التلسكوب هابل ومحطة الفضاء الدولية.

في سبيل مواجهة هذه التحديات، اعتمد الباحثون منظومتي كشف الاعطاب (***fault diagnosis***) والتحكم مع سماحية للأعطاب (***fault tolerant control***). آخذين بعين الاعتبار أن هاتين المنظومتين تعتمدان في الحالة السائدة على النموذج الرياضي (***model***) لديناميكية النظام، لقد اعتمد عملنا البحثي في هذه الاطروحة على محورين: (أ) التهجين بين طريقة متعلقة بالنموذج (مرشح كالمان) وأخرى معتمدة على البيانات بغرض تحسين تقييم عطب الجيرو سكوب، متبوع بإعادة برمجة اعدادات نظام التحكم بتوجه القمر الاصطناعي (***attitude control***)، (ب) تطبيق التعلم المعزز (***reinforcement learning***) كواحدة من أحدث طرائق الذكاء الاصطناعي بغرض توفير نظام تحكم مع سماحية أمثل للأعطاب. تجدر الإشارة الى ان مساهمتنا العلمية فيما يتعلق بالنقطة (ب) آنفة الذكر ركزت أساساً على تكييف دالة المكافأة كون التعليم المعزز مستنبطاً من التصرفات البيولوجية للكائنات الحية وذلك بتضمين نسبة التشابه بين اشارتي العزم الناتجتين عن العامل الذكي (***agent***) من جهة، وعن منظومة تحكم كلاسيكية من جهة أخرى. نتائج المحاكاة المتحصل عليها أثبتت نجاعة الطرائق المقترحة في هذه الاطروحة وذلك بمقارنتها مع أحدث الأنظمة الواردة في المنشورات العلمية.

كلمات مفتاحية: تحكم مع سماحية للأعطاب، نظام توجيه، تعليم معزز، طرائق باستعمال البيانات، كشف الاعطاب.

Contents

List of Figures

List of Tables

List of Abbreviations

AC	Actor-Critic	65
ACS	Attitude control system	4
ADS	Attitude determination system	11
ADCS	Attitude determination & control system	8
AI	Artificial intelligence	8
AOCS	Attitude & orbit control system	xi
CNN	convolutional neural network	27
CNES	Centre National d'Etudes Spatiales	28
CRD	Coarse reconfiguration database	54
DCM	Direction cosine matrix	13
DDPG	Deep deterministic policy gradient	65
DNN	deep neural networks	27
ED	Euclidean distance	78
EGE	expert guided exploration	75
EKF	extended Kalman filter	20
ESA	European Space Agency	16
FDD	fault detection and diagnosis	6
FDI	fault detection and isolation	17
FDIR	fault detection, isolation, and recovery	28
FTC	fault tolerant control	5
GSE	Gyro stellar estimator	42
IMU	Inertial Measurement Unit	17
IRU	Inertial Reference Unit	17

JAXA	Japan Aerospace Exploration Agency	17
JSpOC	Joint Space Operations Center	17
KF	Kalman filter	22
LEO	low earth orbit	15
MDP	Markov decision process	65
MDPs	Markov decision processes	65
ML	Machine Learning	xi
MMAE	multiple model adaptive estimation	24
MRPs	Modified Rodrigues parameters	14
NASA	National Aeronautics and Space Administration	16
PG	policy gradient	73
PID	proportional, integral, and derivative	11
RL	reinforcement learning	12
RUL	Remaining Useful Life	21
RWs	reaction wheels	17
SMC	sliding mode control	12
SMO	sliding mode observer	20
SST	star trackers	22
SVM	Support Vector Machine	27
SW	sliding window	98
TD3	Twin Delayed DDPG	65
VSADC	variability-based self-adaptive dynamical classification	30
VGSADC	Variability Gradient-Based Self Adaptive and Dynamical Classification	49

1

Introduction

An informal, non-mathematical overview of the space missions will be provided in this introduction. In this review, we will concentrate on a particular mission type and how it contributes to the development and sovereignty of states. A brief description of the spacecraft subsystem that will be extensively investigated in this thesis will be provided. Finally, we will summarize the remaining chapters of this dissertation.

1.1 | Space era and space missions

The space era and space exploration represent profound milestones in human history, driven by our innate curiosity, scientific endeavours, and the desire to expand our understanding of the universe. Space exploration revolves around discovering, studying, and utilising celestial bodies beyond Earth, including moons, planets, asteroids, and comets.

The space era began with the launch of the Soviet satellite Sputnik 1 in 1957, marking the first artificial object to orbit the Earth. This historic event ignited a "space race" between the United States and the Soviet Union, culminating in remarkable achievements like the Apollo missions that landed humans on the moon. Since then, numerous nations and private organizations have joined the pursuit of space exploration, transforming it into a global endeavour.

The significance of space exploration lies in its profound impact on various aspects of human civilization. Firstly, it advances our scientific knowledge by enabling us to conduct experiments and observations in the unique environment of space. From studying cosmic radiation and gravitational forces to investigating the universe's origins, space exploration allows us to push the boundaries of our understanding and challenge existing theories. Secondly, space exploration has tremendous technological implications. To overcome the challenges of space travel, we have developed groundbreaking innovations in areas such as rocket propulsion, materials science, telecommunications, robotics, and life support systems. These advancements have benefited space missions and found applications in diverse fields on Earth, leading to improvements in satellite technology, weather forecasting, global communications, medical imaging, and more.

Additionally, space exploration inspires and captivates the human spirit. The images of Earth from space, like the iconic "Blue Marble" photograph, have fostered a greater awareness of our planet's fragility and the need for responsible stewardship. Space missions have also kindled a sense of wonder and imagination, encouraging generations to dream and aspire towards new frontiers. They are powerful reminders of the incredible potential of human ingenuity and collaboration. Furthermore, space exploration holds the promise of addressing pressing global challenges. It opens avenues for resource utilization, such as mining asteroids for rare minerals or extracting water from lunar ice, potentially alleviating resource scarcity on Earth. However, space exploration faces numerous challenges regarding satellite design, deployment, and operation. These challenges stem from the unique environment of space, technological constraints, and the need to ensure long-term functionality and reliability. Let's delve into some of the profound challenges associated with satellites in space exploration:

- Launch and Deployment: The first major challenge lies in successfully launching satellites into space and deploying them in their intended orbits. This process requires precise calculations, timing, and propulsion systems to ensure accurate positioning. Any errors during launch or deployment can result in the satellite being placed in the wrong orbit or rendered non-functional.

- Space Debris: The growing issue of space debris poses a significant challenge for satellite operations. Space debris comprises residuals of rocket stages, defunct satellites, and other fragments from previous spacecraft. Collisions with even small pieces of debris can cause catastrophic damage to satellites. Mitigating this risk requires careful monitoring of space debris, manoeuvring satellites to avoid potential collisions, and designing satellites with shielding and protective measures.

- Radiation and Space Weather: Satellites are exposed to harsh radiation and space weather conditions, which can degrade their components and affect their functionality. Solar flares, cosmic rays, and charged particles from the Sun's coronal mass ejections can damage or disrupt sensitive electronics onboard satellites.

Shielding and robust design are crucial to safeguard satellites against these hazards.

- Power Generation and Management: Satellites rely on power sources like solar panels or onboard batteries to operate in space. Generating and managing power efficiently is challenging due to variations in solar radiation levels, eclipse periods, and power storage limitations. Optimizing power systems and ensuring reliable energy supply throughout the satellite's operational lifespan is essential.

- Communication and Data Transfer: Establishing reliable communication links between satellites and ground stations is vital for data transfer, telemetry, and control. The vast distances involved, signal delays, and potential interference introduce complexities in maintaining consistent and high-bandwidth communication. Advanced communication protocols, antenna design, and signal-processing techniques are employed to address these challenges.

- Thermal Management: Satellites experience extreme temperature variations in space, ranging from intense heat when exposed to sunlight to extreme cold in the shadowed regions. Managing these thermal cycles is crucial to prevent damage to sensitive components and ensure the satellite's operational integrity. Effective thermal insulation, temperature control systems, and proper heat dissipation mechanisms regulate the satellite's temperature.

- Longevity and Maintenance: Satellites are expected to operate for extended periods, ranging from several years to decades. Ensuring their longevity and reliability presents a significant challenge. Designing satellites with robust and redundant systems, implementing fault detection and correction mechanisms, and enabling remote diagnostics and repairs contributes to prolonging their operational lifespan.

- Cost and Resource Limitations: Developing and launching satellites into space can be prohibitively expensive. The costs involve research and development, manufacturing, testing, launch services, and ongoing maintenance. Balancing performance, capabilities, and cost-effectiveness poses a constant challenge, particularly for missions with limited budgets.

The last two points above are the main satellite operations challenges to be addressed in this dissertation. Overcoming these challenges requires interdisciplinary expertise, continuous innovation, and collaborative efforts among space agencies, research institutions, and private companies. As technology advances, addressing these challenges becomes critical to unlock space exploration's full potential and maximise satellite operations' scientific, commercial, and societal benefits.

Since we can launch spacecraft into orbit successfully, our interest in space has grown steadily for various reasons. The latter include, to name a few, scientific research, surveillance, and remote sensing. However, one crucial aspect of most space missions makes them unique: once in orbit, there is no longer any access to physical repair. Satellites used for remote sensing make up a significant portion of artificial earth-orbiting objects. They are mainly utilized for mapping, meteorological, and environmental monitoring. Figure 1.1 illustrates a satellite of this sort. For more technical details about this platform, the reader is referred to [1].

However, one should know that a space mission is a mix of several disciplines such as:

- Mission design and management.
- Flight dynamics.
- Guidance, navigation, and control.
- Communications architectures and networks.
- Launch procedures.
- In-orbit satellite operations.

One of the most crucial elements in the list above is the **Guidance, Navigation, and Control**. The latter is responsible for guaranteeing precise orientation of the platform and/or payload to fulfil mission requirements such as earth and cosmos imaging, sun pointing for battery charging, etc. In this dissertation, we will discuss the Attitude control system (ACS) in terms of the possible faults that may occur in its various parts, and in particular, we will discuss how to detect them by means of fault detection methods, as well as reduce their impact on performance.

1.2 | Satellite attitude control

The control of spacecraft attitude is crucial in meeting mission-pointing requirements, encompassing scientific modes and thruster-pointing. In previous spacecraft mission designs, passive spin stabilization was incorporated to maintain relative stability along one axis by rotating the spacecraft around that particular axis—usually the axis with the most significant moment of inertia. Spin stabilisation was predominantly employed

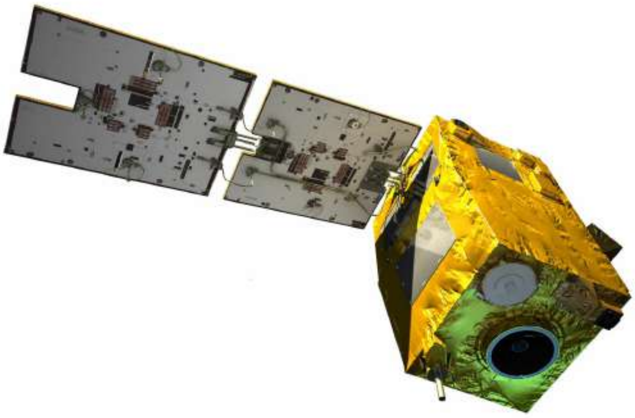


Figure 1.1: Algerian remote sensing satellite ALSAT-2A (Credit: Algerian space agency)

due to limited control actuation and the absence of sophisticated computer technology for implementing complex control algorithms. While spin-stabilized spacecraft offer high stability, achieving this requires meticulous balancing, demanding precision in the construction and positioning of each component.

In the contemporary era, advancements in sensors, actuators, and computer processors have facilitated the construction of three-axis stabilized spacecraft. Nevertheless, spinners continue to find use in numerous missions. Furthermore, the theory of attitude control law has undergone extensive exploration and development, ensuring control stability even in nonlinear attitude dynamics. However, the challenge arises when controlling spacecraft during large-angle slewing manoeuvres. These challenges encompass the highly nonlinear nature of the governing equations, constraints and limits on control rates and saturation, and inadequate state knowledge resulting from sensor failure or omission. Effective control during substantial angular manoeuvres may employ open-loop or closed-loop techniques. [2].

1.3 | What is fault-tolerant control of satellite attitude?

Fault tolerant control (FTC) for aerospace systems refers to designing and implementing control strategies that enable continued safe and reliable operation even in the presence of faults or failures. FTC aims to detect, isolate, and accommodate faults, minimizing their impact on system performance and ensuring the system's ability to maintain its desired functionality and stability. Here is an explanation of fault-tolerant control for aerospace:

- ***Fault Detection:*** The first step in FTC is to detect the occurrence of faults in the system. This involves monitoring various system components, such as sensors, actuators, and subsystems, for deviations from their expected behaviour or performance. Fault detection algorithms analyze sensor measurements, compare them to expected values, and detect anomalies or discrepancies that may indicate the presence of a fault.

- ***Fault Isolation:*** Once a fault is detected, the next step is to isolate the faulty component or subsystem responsible for the observed abnormal behaviour. Fault isolation techniques aim to identify the specific location and type of fault, distinguishing it from other possible sources of system degradation. This is crucial for taking appropriate corrective actions and implementing fault accommodation strategies accurately.

- ***Fault Accommodation:*** After fault detection and isolation, fault accommodation techniques are employed to mitigate the effects of the identified fault. Different strategies can be employed depending on the severity and nature of the fault. For instance, in the case of an actuator fault, redundant actuators may be utilized to compensate for the faulty one. Alternatively, control reconfiguration techniques can be applied to adapt the control strategy to the faulted condition or switch to backup control modes. The aim is to maintain system performance and stability within acceptable limits despite the presence of faults.

- ***Redundancy and Diversity:*** Redundancy plays a vital role in fault-tolerant control for aerospace systems. The system can continue operating even if one or more components fail by providing duplicate or backup components, such as sensors or actuators. Redundancy can be implemented at various levels, including hardware redundancy (duplicate physical components) and software redundancy (redundant algorithms or control strategies). Diversity, which involves employing different technologies or design approaches for redundant components, further enhances fault tolerance by reducing the likelihood of common-mode failures.

- ***Fault Management and Decision Making:*** FTC systems require intelligent decision-making capabilities to determine the most appropriate course of action when faults occur. This involves assessing the severity of the fault, evaluating the available fault accommodation options, and selecting the optimal strategy based on predefined criteria or objectives. Advanced algorithms and decision-making frameworks, such as model-based reasoning, fault detection and diagnosis (FDD), and adaptive control techniques, are employed to ensure efficient and effective fault management.

Overall, FTC for aerospace systems aims to enhance the safety, reliability, and availability of aircraft, spacecraft, and other aerospace platforms. By incorporating fault detection, isolation, and accommodation techniques, along with redundancy and intelligent decision-making capabilities, aerospace systems can continue to operate within acceptable performance bounds, even in the presence of faults or failures. This ensures the protection of human life, preserves valuable assets and supports the successful completion of critical missions in the aerospace domain.

1.4 | Machine learning in industrial processes

ML is one of the most rapidly expanding subfields of computer science and has numerous applications. ML has revolutionized industrial processes by enabling automation, optimization, and predictive capabilities. It leverages algorithms and statistical models to enable computers to learn from data, identify patterns, make predictions, and take actions without explicit programming. The following is a non-exhaustive list of industrial uses of machine learning [3]:

- ***Energy Sector***: where ML can be used to predict the proportion of oil or gas moved through pipelines from offshore drilling locations or to quantify the uncertainties in reservoir utilization using previous reservoir data. ML can also be used for oil price forecasting and problem detection or classification in renewable energy systems, which have significant economic effects.

- ***Basic Materials Sector***: in the *chemicals* business, machine learning can play a significant role in production, drug creation, toxicity prediction, and compound classification. Paper production, industrial metals, and mineral classification are all *Basic Resources* subfields made more efficient by incorporating machine learning techniques.

- ***Healthcare Sector***: Machine learning can help reduce diagnosis and therapeutic errors unavoidable in human clinical practice. The superiority of machine learning in this sector comes from the accurate models learned from the vast amount of healthcare data. ML is typically beneficial in Health sub-fields like: *neuroscience*, *cardiovascular*, *cancer*,*obesity*, and *diabetes*.

- ***Telecommunications Sector***: The excessive growth of telecommunication-related data during the 21st century forcibly pushes corporations toward using Artificial intelligence (AI) to deal with the huge amount of, but not limited to, social media and customers services data. One of the most famous applications is the detection of spam emails. Additionally, the fast-growing number of mobile network customers makes it compulsory to adopt self-organizing networks that use ML techniques to operate.

- ***Information Technology Sector***: This sector includes multiple industries, usually associated with computer science, such as computer hardware, software, the Internet, and semiconductors. Fraud detection as a Cybersecurity sub-field is one of the most famous sectors where it has been reported that online payment companies such as PayPal and Visa use ML techniques like artificial neural networks and deep learning.

It's important to note that successfully implementing ML in industrial processes requires access to high-quality data, scalable computing infrastructure, and expertise in data analytics and model development. Additionally, data privacy, security, and ethical use of ML should be addressed to ensure responsible and trustworthy deployment in industrial settings.

1.5 | Thesis outline and contributions

The dissertation acknowledges that the Attitude determination & control system (ADCS) is susceptible to faults, failures, and malfunctions. Novel technologies are explored to address these challenges. Nevertheless, the prevalence of model-based methods persists in current practice, making complete avoidance of their use impractical. Therefore, this dissertation leverages classical approaches to enhance data-driven and intelligent methods implementation. The remainder of this dissertation is structured as follows:

Chapter 2 extensively examines the central themes of this research, specifically the Attitude Control System (ACS) and the fault diagnostic and fault-tolerant control aspects associated with ACS. The initial section introduces the constituent elements of ACS, such as sensors and actuators, which are susceptible to malfunctions and irregularities. Subsequent sections delve into using model-based techniques to address fault-related issues. The final two sections comprehensively review the state-of-the-art techniques employed in FDD and FTC within attitude control.

Chapter 3 introduces a hybridized approach that combines data-driven and model-based Kalman filtering techniques to enhance the stability of fault diagnosis. The primary contribution of this chapter lies in demonstrating that the utilization of variability-based data-driven methods in conjunction with the Kalman filter, based on ACS sensor residuals, yields superior (FDD) performance. Additionally, incorporating supplementary reconfiguration of the estimator/controller enhances the system's response following sensor fault incidents. This enhancement improves the fault-tolerant control loop's capability to provide superior performance in the event of a fault occurrence.

In Chapter 4, there is a notable advancement in improving existing machine learning methodologies by developing a novel training approach for reinforcement learning agents. This innovative technique, rooted in the concept of reward shaping, involves directing the agent towards a policy space characterized by high returns. This is achieved by integrating a similarity measure between the agent's output and a reference control. The resultant trained agent demonstrates enhanced performance, particularly in scenarios involving faults and uncertainties in system parameters. Chapter 5 offers a comprehensive overview of the research's findings and summarises the work conducted. It also presents suggestions for potential future research directions in alignment with this study.

Moreover, this thesis incorporates an appendix (Appendix A) containing the reinforcement learning environment and the code utilized for reward shaping. These components were applied in Chapter 4 to validate the expert-guided exploration strategy. Additionally, the code for the reinforcement learning setup is included.

1.6 | List of publications

Journal papers

- Henna H, Toubakh H, Kafi MR, Gürsoy Ö, Sayed-Mouchaweh M, Djemai M. Satellite fault tolerant attitude control based on expert guided exploration of reinforcement learning agent (2024). *Journal of Experimental & Theoretical Artificial Intelligence*.

- Henna H, Toubakh H, Kafi MR, Sayed-Mouchaweh M, Djemai M. Hybrid supervision scheme for satellite attitude control with sensor faults (2024). *CEAS Space Journal*.

Conference papers

- Henna H, Toubakh H, Kafi MR, Sayed-Mouchaweh M. Towards Fault-Tolerant Strategy in Satellite Attitude Control Systems: A Review. *In Annual Conference of the PHM Society*. 2020 Nov 3 (Vol. 12, No. 1, pp. 14-14).

- Henna H, Toubakh H, KAFI MR, Sayed-Mouchaweh M. Unsupervised data-driven approach for fault diagnostic of spacecraft gyroscope. *In Annual Conference of the PHM Society*. 2022 Oct 28 (Vol. 14, No. 1).

- H. Henna, E. Gasmi, H. Toubakh, M. R. Kafi, M. Djemai and M. Sayed-Mouchaweh. Fault Tolerant Attitude Estimation for Satellite at Low Earth Orbit. *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, Rome, Italy, 2023, pp. 275-280, doi: 10.1109/CoDIT58514.2023.10284329.

2

Attitude control system, fault diagnostic and fault tolerant control

2.1 | Introduction

This chapter will explore the investigated system's mathematical modelling, architecture, and recurring challenges. Then, from an automatics perspective, we will detail the system's abnormal conditions and how they were resolved. In the final section of this chapter, the procedures for reconfiguration in the event of errors will be described. Often known as FTC, these methods allow mission continuity in atypical situations with degraded but acceptable performance.

2.2 | Attitude control system

The ACS stabilises the satellite's orientation concerning some reference point or trajectory. ACS achieves this by:

- *Attitude sensing*: often known as the Attitude determination system (ADS). Using the appropriate sensors (see Figure 2.1) and a filtering algorithm for noise attenuation, ADS gives precise body frame orientation relative to the reference frame.

- *Control computation*: the navigation system typically provides the pointing error between (i) the attitude calculated by the ADS and (ii) the desired attitude (reference). Then, an inbuilt algorithm calculates the relative control signal (torques) to adjust for inaccuracy. These algorithms range from the traditional proportional, integral, and derivative (PID) control to highly modern and resilient approaches

(a) Star tracker (Credit: Jena-Optronik, DLR , ESA) measure the relative attitude *w.r.t* the inertial frame.

(b) Gyro (Credit: Northrop Grumman) measure the angular rate of body frame *w.r.t* the inertial frame.

(c) Magnetometer (Credit: NASA(Pioneer 10 project)) it gives the magnitude and direction of the earth magnetic field.

(d) Sun sensor (Credit: NewSpace Systems) provides the sun-vector angle relative to the sensor normal.

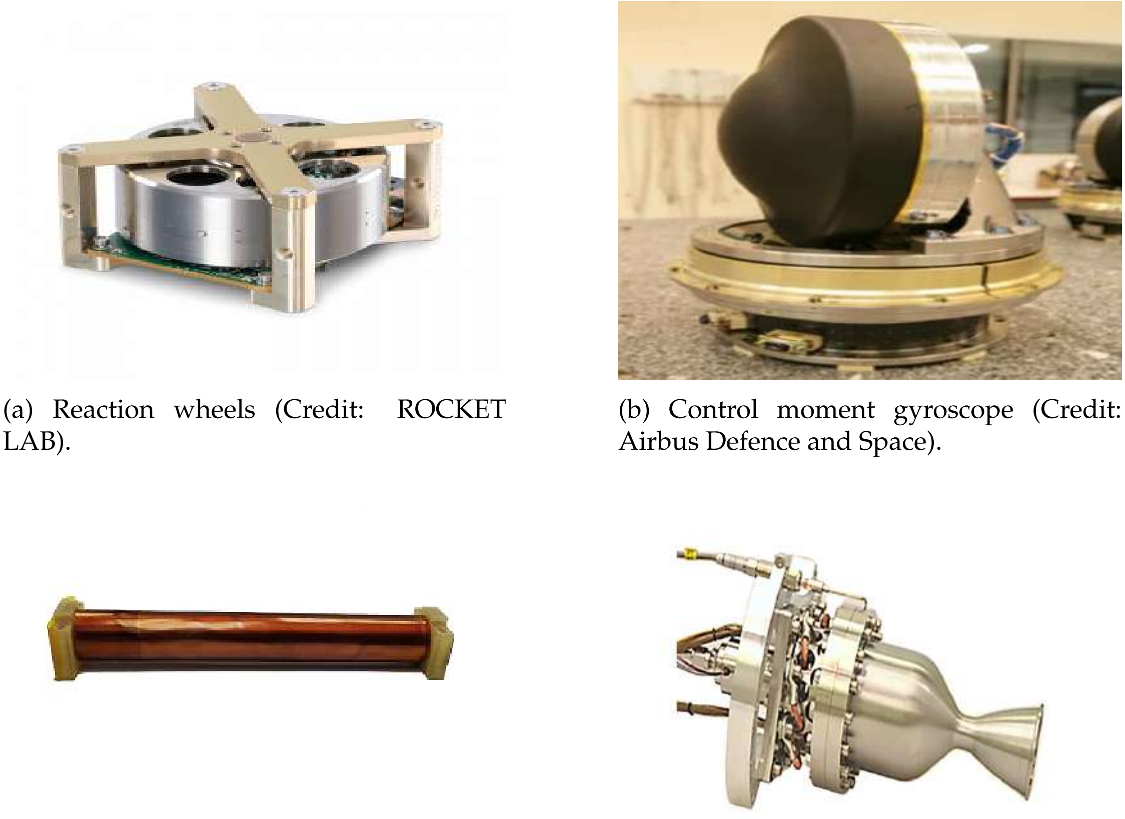
Figure 2.1: Attitude sensors.

(such as sliding mode control (SMC)) and even data-driven or intelligent control (e.g. reinforcement learning (RL)).

- *Actuation*: the control torques are then supplied to the attitude actuators (see Figure 2.2), stabilizing the satellite's attitude relative to the reference.

2.2.1 | Attitude representation

Typically, many reference frames are used to address the spacecraft orientation dynamics. To represent the orientation of a satellite about the Earth, for instance, the first reference frame (let us say $\mathcal{F}_1$) will be Earth-based, and the second (let us say $\mathcal{F}_2$) will be spacecraft body-based. The spacecraft's orientation relative to the Earth is then de-



(a) Reaction wheels (Credit: ROCKET LAB).

(b) Control moment gyroscope (Credit: Airbus Defence and Space).

(c) Magneto-torquer (Credit: NewSpace Systems).

(d) Thruster (Credit: Bradford Space).

Figure 2.2: Attitude actuators.

fined by the orientation of reference frame $\mathcal{F}_2$ relative to $\mathcal{F}_1$. This orientation is known as the *attitude* in aerospace nomenclature [4].

Approaches for establishing the accurate orthogonal matrix that transforms vectors from a space-fixed reference frame to a frame fixed in the spacecraft body are employed for spacecraft attitude determination, as outlined by [2]. Various parameterizations exist to describe the orientation of one frame about another, starting with the Direction cosine matrix (DCM), which serves as the foundational element for any attitude representation.

- *Euler angles*: ϕ, θ, ψ represent the *roll*, *pitch*, and *yaw*. Each angle depicts the rotation of an axis in $\mathcal{F}_2$ about its corresponding axis in $\mathcal{F}_1$. This form is only appropriate for small rotations due to singularity (i.e., gimbal lock) at higher rotations [5].

- *Quaternions*: specifically, the *unit* quaternion q, which is a quadruple ($q = [q_0, q_1, q_2, q_3]$)

used to express rotations without the risk of singularity. Additionally, it must adhere to a single constraint, the norm constraint: $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$.

- *Modified Rodrigues parameters (MRPs)*: The most recent attitude representation, adopting MRPs, minimizes the calculation overhead due to their 3-element structure. Like quaternions, the switching method between one MRP and its shadow prevents the singularity when the norm slightly surpasses 1.

The references [4, 2, 5] contain extensive information about these representations and their mathematical basis.

2.2.2 | Attitude model

Modelling a spacecraft's attitude involves solving differential equations to determine its dynamics. A complete description requires two sets of equations:

- *kinetics*: depicts the evolution of the body frame's angular rates ω_i in response to the torques applied to the body.
- *kinematics*: which defines the evolution of the attitude (expressed, for example, as unit quaternion q) in terms of the angular rates.

Define $\boldsymbol{\omega}$ as the 3-components vector of body frame angular rates relative to the inertial frame, $[I]$ as the inertia matrix typically expressed in the body frame. Then the famous *Euler rotational equation of motion* is given as:

$$[I]\dot{\boldsymbol{\omega}} = -[\tilde{\boldsymbol{\omega}}][I]\boldsymbol{\omega} + \boldsymbol{L_c} \tag{2.1}$$

where $\boldsymbol{L_c}$ denotes the vector of external torques, and the tilde operator is defined as:

$$[\tilde{\boldsymbol{\omega}}] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{2.2}$$

As the unit quaternion is commonly used in industry to express attitudes, we shall select it for this dissertation. Eq. (2.3) provides the four coupled kinematic differential equations for unit quaternion [5]:

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} \tag{2.3}$$

Using the quaternion form, Eqs. (2.1) and (2.3) fully characterize the spacecraft's attitude, typically relative to an inertial frame.

2.2.3 | Attitude control

Consider a spacecraft that must adopt a new attitude, such as orienting the telescope to a specific location on the earth's surface (imaging mode) or rotating the body so the solar panel faces the sun for battery charging. The desired manoeuvre will be executed by deriving the control effort from the trajectory description and inverse dynamics. These motions are referred to as open-loop manoeuvres without position or velocity feedback. Unreliable system models, unmodeled dynamics, and external influences will naturally cause the spacecraft to deviate from its intended trajectory. To guarantee the manoeuvre's stability, a feedback control law is necessary. This *feedback* term is linked to and reflects the ADS task described above and is accountable for providing the attitude state. The navigation system then compares this information to where the spacecraft should be at any given time. This comparison results in state (pointing) errors based on the attitude representation employed ($\delta\omega, \delta q, \delta\theta, ...$). Then, these errors are used to modify the control signal so that the spacecraft follows the desired trajectory with minimal deviation [5].

Crafting spacecraft attitude control laws demands a grasp of rigid body kinematics, kinetics, and control techniques. For instance, the appropriate choice of attitude coordinates holds significance in determining the efficacy of the resultant control law. Opting for any set within the Euler angle family might prove inefficient for executing substantial, arbitrary rotations owing to their limited nonsingular rotation range. In such cases, alternative representations like quaternions and MRPs are preferred [5]. Numerous control laws have been discussed in academic [6, 7] and industrial [8, 9] publications. In the following sections, we will provide an overview of the ACS error-handling strategies that have been developed.

2.3 | Fault diagnostic in ACS

In recent decades, numerous space applications have been created to provide civilian and military services, including meteorological forecasting, remote sensing, and space-based scientific facilities. Microsatellites are highly effective in achieving these objectives due to their low cost during design and production. Exogenous disturbances, such as magnetic perturbation, gravity gradient, and atmospheric drag, significantly impact attitude stability when microsatellites are designed to operate in low earth orbit (LEO).

Space debris constitutes an additional hazard in this environment. During the in-orbit operations of spacecraft, faults/failures may arise due to the aforementioned environmental conditions and the propensity of some physical components to fail, notably mechanical ones [10].

Despite the quality improvement of components in the space industry, the frequency of newly-emerging errors rises as system complexity increases. There are three types of faults, as mentioned earlier: (i) system faults, (ii) actuator faults, and (iii) sensor faults. These faults are crucial to the satellite's attitude control for the reasons listed below [10]:

- Poorer attitude determination in the event of sensor failure. Therefore, divergence of guidance will occur, resulting in the occasional activation of satellite-safe mode to ensure payload safety.
- Poorer controllability with actuator defects; thus, the performance of the attitude control loop (rise time, precision, etc.) suffers.
- Another possible, albeit uncommon, scenario involves software (process) errors such as division by zero or abnormal normalization of specific parameters (e.g., quaternion).

This research will focus on faulty sensors and actuators. Figure 2.3 depicts different sensor defects, while Figure 2.4 depicts actuator problems [11].

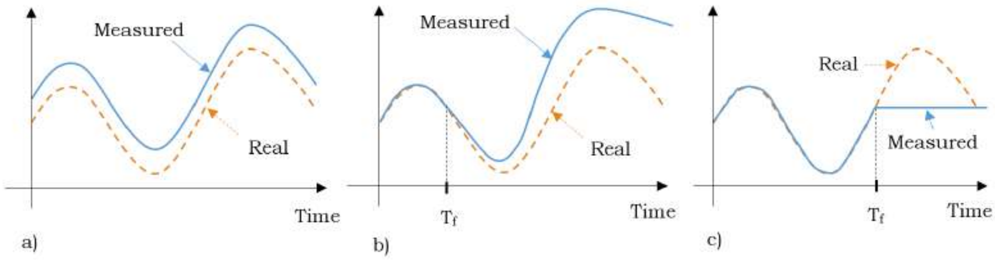


Figure 2.3: (a) bias fault; (b) drift fault; (c) frozen sensor.

It is now practical to provide real-world examples of component failures to illustrate the impact of those failures on spacecraft health and functionality. (i) SOHO, the SOlar and Heliospheric Observatory (Figure 2.5(a)), (ii) The Kepler space telescope (Figure 2.5(b)), and (iii) the Japanese X-ray observatory Hitomi (Figure 2.5(c)).

A series of gyroscope calibration problems halted SOHO's mission between June 25 and October 24, 1998. The spacecraft has resumed its regular sun-pointing mode following rigorous recovery efforts undertaken by National Aeronautics and Space Administration (NASA) and European Space Agency (ESA) in partnership (mission recovery

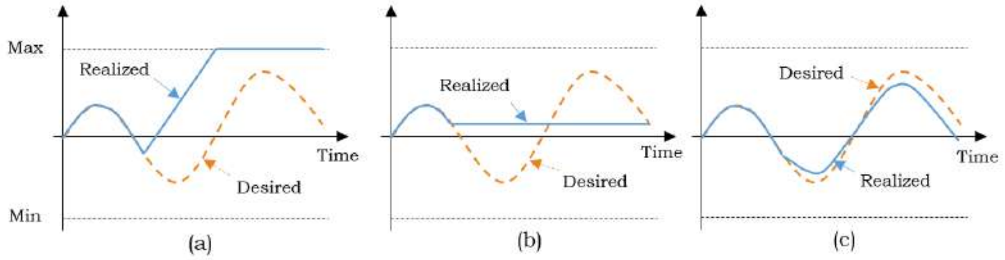


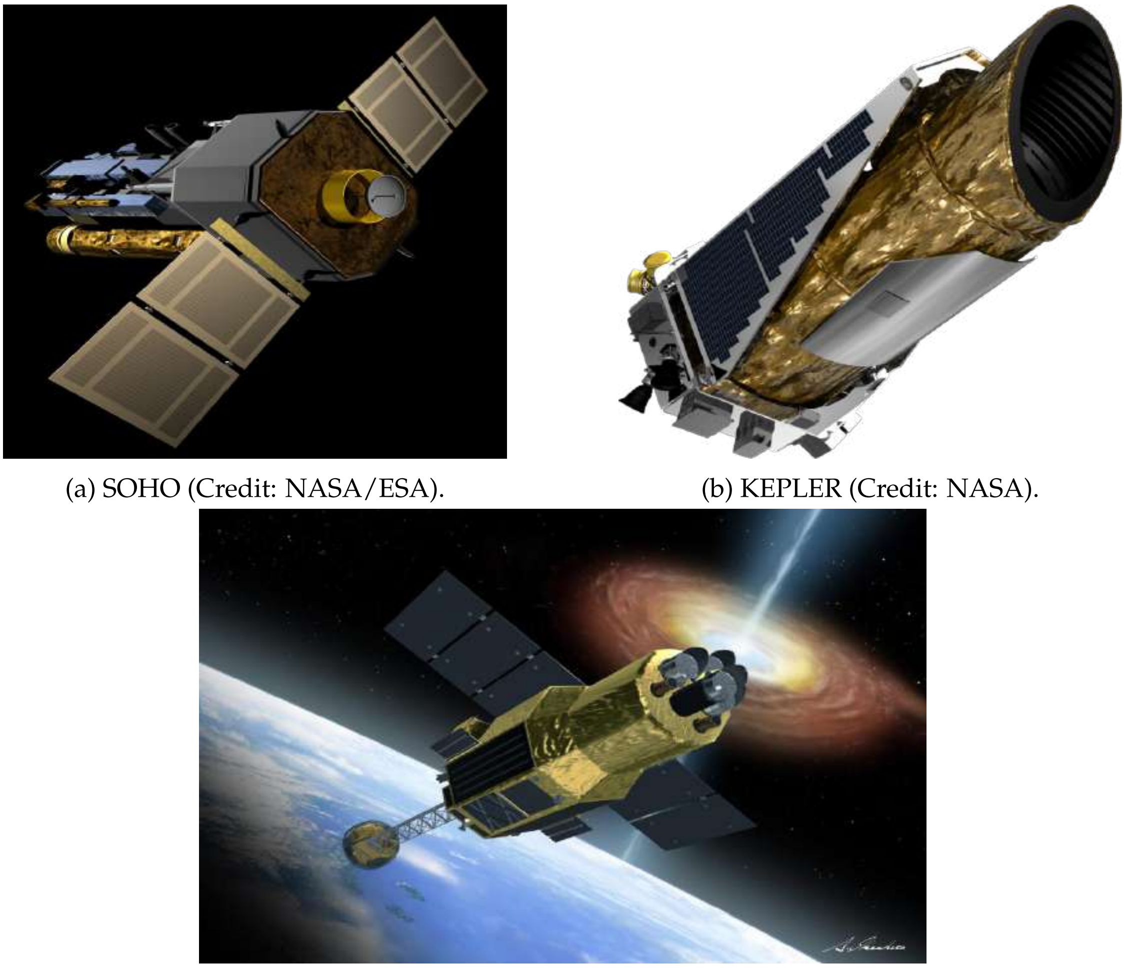
Figure 2.4: (a) hard-over fault; (b) locked-in-place fault; (c) loss of effectiveness.

timeline is illustrated in Figure 2.6). On joint advice, the operation engineers implemented a gyroless mode on February 1, 1999 [12]. Even though the near loss of SOHO was primarily caused by improper ground crew response, precise FDD could intervene and improve the judgment of ground crew. Examples of misconduct like when the crew turned off the gyro-B (deemed defective) which triggered the 7^{th} ESR (emergency sun re-acquisition) switch.

Wheel # 2 of Kepler's four reaction wheels failed in July 2012. On May 11, 2013, a second wheel (# 4) failed, endangering the mission's continuance as it requires three wheels for planet hunting. Since May, Kepler has been unable to collect scientific data because of its inability to point with sufficient precision. The engineers returned to a "rest" state to conserve fuel after determining that wheel # 2 could not deliver the requisite accuracy for scientific missions. NASA reported on August 15, 2013, that Kepler would no longer use the transit method for planet imaging after trying to remedy problems with 2 out of 4 reaction wheels had failed [13].

The repercussions for the next mission were more severe (complete mission loss). Hitomi was launched on February 17, 2016, and declared lost on April 1 by the U.S. Joint Space Operations Center (JSpOC) due to body fragmentation into ten parts. Japan Aerospace Exploration Agency (JAXA) examinations confirm the complete attitude control loss that occurred on March 25 due to incorrect Inertial Reference Unit (IRU) measurements. The Inertial Measurement Unit (IMU) registered a Z-axis rotation of 21,7 degrees per hour when the spacecraft was, in fact, stable [14]. To offset the nonexistent spin, the ACS transmitted excessive control values to the reaction wheels (RWs) in the initial phase and the thrusters later. On March 26, only 39 days after launch, the entire spacecraft disintegrated due to excessive rotational speed.

The missions mentioned above were in jeopardy due to subsystem problems, highlighting the need for a fault evaluation system known as fault detection and isolation (FDI). Occasionally, this module is known as FDD. These FDI methods are divided into three primary subsystems. Initially, the malfunction must be discovered using the



(a) SOHO (Credit: NASA/ESA).

(b) KEPLER (Credit: NASA).

(c) HITOMI (Credit: JAXA).

Figure 2.5: Space missions with complete/partial failures.

fault detection system. After identifying the fault, the following step involves isolating the defective component. Further, the identification module determines the magnitude of the fault. As depicted in Figure 2.7, one of the fundamental concepts behind FDI is the generation of residual before a threshold-based decision function is triggered to activate the reconfiguration mechanism.

In [15], FDD was classified depending on the availability of spacecraft physical models. This categorization is depicted in Figure 2.8. FDI is primarily separated into two broad types depending on whether the designer has access to a physical model: (i) model-based and (ii) model-free or data-driven techniques.

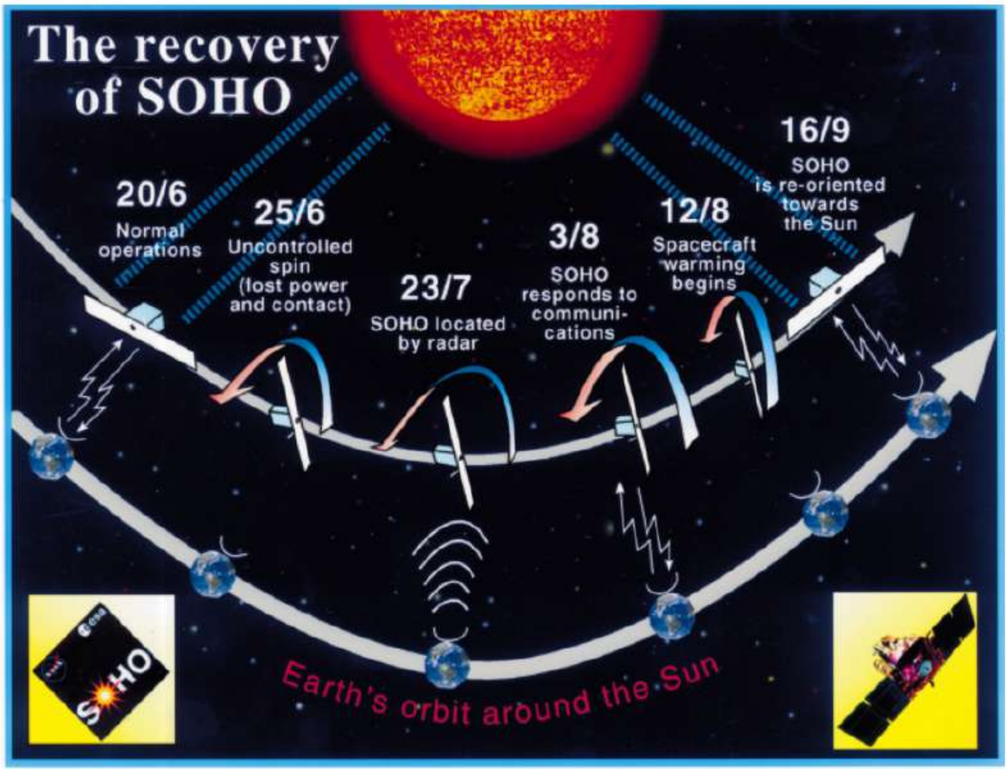

The recovery of SOHO
20/6
Normal operations
25/6
Uncontrolled spin (lost power and contact)
23/7
SOHO located by radar
3/8
SOHO responds to communi-cations
12/8
Spacecraft warming begins
16/9
SOHO is re-oriented towards the Sun
Earth's orbit around the Sun
SOHO

Figure 2.6: SOHO recovery timeline [12].

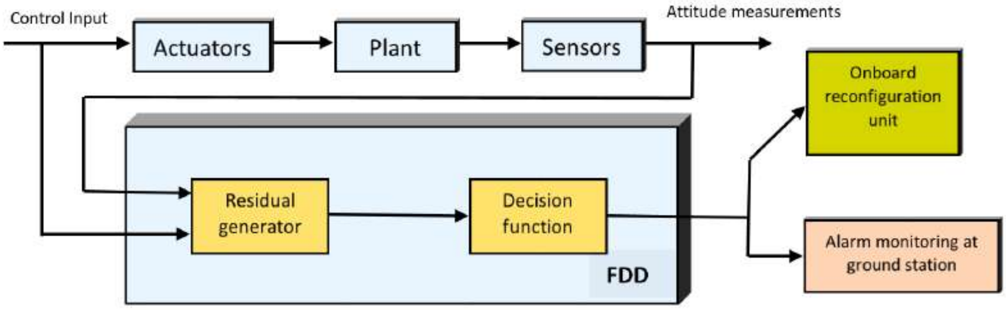


Figure 2.7: FDI principle (example: satellite attitude FTC) [10].

2.3.1 | Model-based FDI

2.3.1.1 | Brief literature review

In the FDD and FTC fields, numerous researchers have focused on model-based approaches where the FDD algorithm is designed upon the definition of the spacecraft model. A non-exhaustive list of model-based attitude control FDD research is shown in Table 2.1.

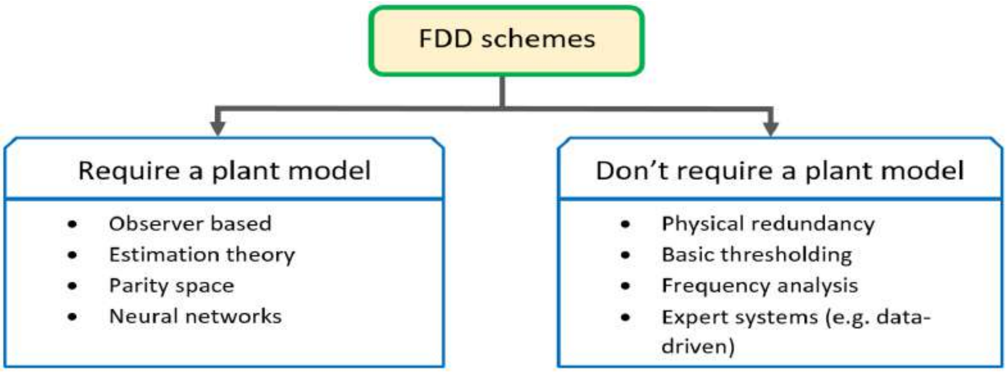


Figure 2.8: FDD classification based on the necessity of a model [15].

Table 2.1: Examples of model-based FDD employed for ACS.

FDD technique		Works based on the technique
Observer based	Sliding mode	[7, 16, 17, 18]
	$\mathcal{H}_\infty, \mathcal{H}_2$	[19, 20]
	Adaptive	[21]
Kalman filter/Extended KF		[22, 23, 24, 25, 26]
Parity space		[27]

This direction has seen a range of significant contributions. To begin, [28] introduced a model-based FDD approach, using housekeeping telemetry data. They obtained nominal behaviour through simulations and detected deviations in the behaviour of satellite components by comparing expected and observed behaviours. [21] proposed using adaptive observers to estimate actuator and sensor faults. Their approach incorporated a nonlinear geometric strategy to handle the influence of aerodynamic disturbance torques on fault estimation. To estimate time-varying actuator faults, [7] presented the utilization of a sliding mode observer (SMO). This observer's stability was established using the Lyapunov theory. SMOs are recognized for their robustness to external disturbances and parameter variations, making them a favoured choice in spacecraft ACS fault-tolerant methods. [22] addressed sensor and actuator faults in satellite ACS by employing two extended Kalman filter (EKF) for attitude estimation. The information derived from these EKFs was used to ensure FDI within the system. [25] developed a scheme for accommodating gyroscope faults, particularly in the orthogonal installation of 3-axis gyroscopes and one gyroscope installed at a slantwise orientation. They assessed measurement consistency between the gyroscopes and the star sensor through a bank of dedicated Kalman filters. [26] tackled FDI for agile spacecraft, focusing on detecting and isolating faults in thrusters, RWs, and RW tachometers. Their FDI scheme involved using a Generalized Likelihood Ratio (GLR) technique applied to the residual signal. [19] introduced a nonlinear observer within an $\mathcal{H}_\infty$ framework for satellite formation flight. In this scheme, each satellite within the constellation could diagnose faults in its neighbouring satellites. [29] presented an EKF-based approach to predict the Remaining Useful Life (RUL) of RW motors in satellites. This method utilized two EKF algorithms for the prediction of RW motor RUL.

[30] conducted a quantitative classification of model-based FDD approaches, considering factors such as suitability for FTC design and computational complexity. They suggested that simultaneous or multiple parameter estimation, including multi-observers, EKF, and 2-stage KF methods, is more appropriate for FTC applications. However, these methods may be less suitable in cases where computational resources are limited. These contributions represent the extensive research in model-based FDD and its application in the context of Fault-Tolerant Control for various aerospace systems. In the following section, several model-based FDI methods applied in spacecraft ACS will be presented, including Sliding Mode Observer (SMO), two parallel Kalman Filters (KF), and Model-based Multiple Model Adaptive Estimation (MMAE).

2.3.1.2 | Examples of spacecraft model-based FDI

The kinetics equation must be modified to adapt the model to specific emerging fault scenarios. In [31], for instance, multiple actuator problems are addressed. Including the fault's information requires modifying Eq. (2.1). Consequently, the kinetics equation is rewritten as:

$$[I]\dot{\boldsymbol{\omega}} = -[\tilde{\boldsymbol{\omega}}][I]\boldsymbol{\omega} + \boldsymbol{L}_c + Db\boldsymbol{u} + D\boldsymbol{\alpha} \tag{2.4}$$

where $D \in R^{3\times n}$ represents the distribution matrix of n RWs. $b, \boldsymbol{u}$, and α are utilized to describe actuator failure, as shown in Eq. (2.5):

$$u_i^F = b_i u_i + \alpha_i \tag{2.5}$$

where u_i^F, b_i, u_i, and α_i represent the delivered torque, effectiveness factor, desired torque, and bias fault for the ith reaction wheel, respectively. Then, the fault detection strategy can be built utilizing the augmented system. For instance, a SMO is derived, and an adaptive fault detection threshold l is defined. The observer's efficiency is demonstrated in both the healthy ($b = I, \alpha = 0$) and defective ($b \neq I, \alpha \neq 0$) scenarios. When the error signal (ω_e) exceeds l, at least one actuator has an unidentified fault.

Another common estimation method is the Kalman filter (KF). The latter is mostly used for optimal state estimation. At the same time, it can also be employed for fault diagnosis, as the fault can be viewed as an additional state in the dynamics or kinematics equations. Based on the work in [22], the modification of dynamical equations will be described in detail below. First, the gyroscope (see Figure 2.1(b)) and star trackers (SST) (see Figure 2.1(a)) measurement models are given by:

$$\begin{aligned} \omega_m &= \omega + b + \nu_g + f_g \\ \dot{b} &= \nu_b \\ q_m &= q + \nu_q + f_q \end{aligned} \tag{2.6}$$

where ω_m and q_m denote the sensed quantities for angular velocity and quaternion, ω and q represent the actual quantities. b represents the gyro bias; ν_g, ν_b, and ν_q are three uncorrelated zero-mean Gaussian white noises; f_g and f_q represent the sensor fault signals. As in Eq. (2.4), the kinetic equation is also modified to model actuator defects.

As indicated in Figure 2.9, the fault diagnosis scheme can now be built as a bank of EKFs [22]. In EKFs design, both models (*dynamics* and *kinematics*) are required. Let q_e represent the quaternion deviation as in:

$$q = \hat{q} \otimes q_e \tag{2.7}$$

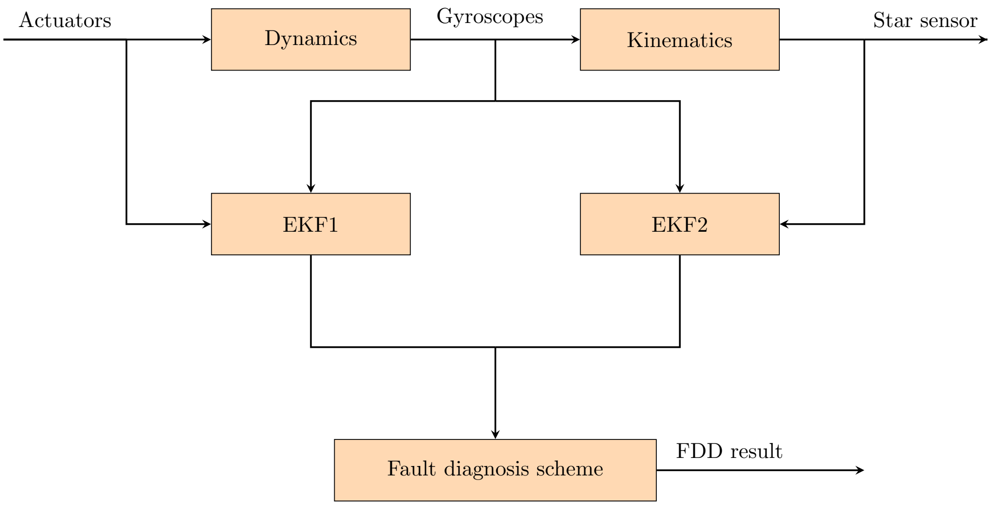


Figure 2.9: FDD using analytical redundancy and EKF [22].

The error kinematics become:

$$\dot{q_e} = \frac{1}{2} q_e \otimes \hat{\omega} - \frac{1}{2} \hat{\omega} \otimes q_e + \frac{1}{2} q_e \otimes \Delta\omega \tag{2.8}$$

where

$$\hat{\omega} = \omega_m - b$$
$$\Delta\omega = \begin{bmatrix} 0 & \omega - \hat{\omega} \end{bmatrix}^T = \begin{bmatrix} 0 & -\Delta b - \nu_g \end{bmatrix}^T$$
$$\Delta b = b - \tilde{b}$$

$\tilde{b}$ estimates the gyroscope's biases. Any quaternion can be expressed to separate the scalar and vector components, so q_e can be written as: $q_e = \begin{bmatrix} q_{e0} & q_{ev} \end{bmatrix}^T$. By simplifying Eq. (2.8), we obtain:

$$\begin{aligned} \dot{q}_{ev} &= -[\tilde{\omega}]q_{ev} - \frac{1}{2}(\Delta b + \nu) \\ \dot{q}_{e0} &= 0 \end{aligned} \tag{2.9}$$

For the filter state equation, the derivations above yield

$$\begin{bmatrix} \dot{q}_e v \\ \Delta\dot{b} \end{bmatrix} = \begin{bmatrix} -[\tilde{\omega}] & -\frac{1}{2}I_3 \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \begin{bmatrix} q_e v \\ \Delta b \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}I_3 & 0_{3\times3} \\ 0_{3\times3} & I_3 \end{bmatrix} \begin{bmatrix} \nu_g \\ \nu_b \end{bmatrix} \tag{2.10}$$

Eq. (2.10) can be expressed in the following compact form using the state vector $x_1 = \begin{bmatrix} q_{ev} & \Delta b \end{bmatrix}^T$:

$$\begin{aligned} \dot{x}_1(t) &= F_1 x_1(t) + G_1 \epsilon_1(t) \\ y_1(t) &= H_1 x_1(t) + \nu_q \end{aligned} \tag{2.11}$$

where $H_1 = \begin{bmatrix} I_3 & 0_{3\times3} \end{bmatrix}$.

The discrete system matrices (precisely, Φ_1 and Γ_1 in Eq. (2.12) below) can therefore be defined as functions of the continuous system matrices (F_1 and G_1) as well as the sampling period T using some discretization technique. The discrete state equation is therefore given by (using the definitions: $\Phi_1(k, k-1) = f(F_1, G_1, T)$ and $\Gamma_1(k) = g(F_1, G_1, T)$):

$$\begin{aligned} x_1(k) &= \Phi_1(k, k-1)x_1(k-1) + \Gamma_1(k-1)\epsilon_1(k-1) \\ y_1(k) &= H_1(k)x_1(k) + \nu_q(k) \end{aligned} \tag{2.12}$$

The KF can be created using prediction and update as usual (the author directs the reader to [22] for a thorough mathematical explanation).

Similarly, the *nonlinear* dynamical equation and state vector $x_2 = \begin{bmatrix} \omega & T_c \end{bmatrix}^T$ can be used to derive the second filter:

$$\begin{bmatrix} \dot{\omega} \\ \dot{T}_c \end{bmatrix} = \begin{bmatrix} [I]^{-1}T_c - [I]^{-1}[\tilde{\omega}][I]\omega \\ -\frac{1}{\tau}T_c \end{bmatrix} + \begin{bmatrix} 0_{3\times3} \\ I_3 \end{bmatrix} \nu_\omega \tag{2.13}$$

Table 2.2: Diagnostic logic for faults

No.	Diagnostic logic	Fault type
1	$\|r_1\| < \lambda_1, \|r_2\| < \lambda_2$	Fault-free
2	$\|r_1\| \geq \lambda_1, \|r_2\| < \lambda_2$	SST fault
3	$\|r_1\| < \lambda_1, \|r_2\| \geq \lambda_2$	Actuators fault
4	$\|r_1\| \geq \lambda_1, \|r_2\| \geq \lambda_2$	Gyroscopes fault

Providing Φ_2 and Γ_2 functions of (F_2, G_2 and T) again discretizes this equation, where

$$\begin{aligned} F_2 &= \left.\frac{\partial \begin{bmatrix} [I]^{-1}T_c - [I]^{-1}[\tilde{\omega}][I]\omega \\ -\frac{1}{\tau}T_c \end{bmatrix}}{\partial x}\right|_{x=\hat{x}} \\ &= \begin{bmatrix} \left.\frac{\partial([I]^{-1}T_c - [I]^{-1}[\tilde{\omega}][I]\omega)}{\partial x}\right|_{\omega=\hat{\omega}} & \left.\frac{\partial(-\frac{1}{\tau}T_c)}{\partial x}\right|_{\omega=\hat{\omega}} \end{bmatrix}^T \\ G_2 &= \begin{bmatrix} 0_{3\times 3} \\ I_3 \end{bmatrix} \end{aligned} \tag{2.14}$$

Furthermore, the second filter's discrete state equation is provided by:

$$\begin{aligned} x_2(k) &= \Phi_2(k, k-1)x_2(k-1) + \Gamma_2(k-1)\epsilon_2(k-1) \\ y_2(k) &= H_2(k)x_2(k) + \nu_g(k) \end{aligned} \tag{2.15}$$

where $H_2 = H_1$.

Two residual signals have been defined in this model-based FDD as:

$$\begin{aligned} r_1(k) &= q_{ev}(k) \\ r_2(k) &= \Delta\omega(k) = \omega(k) - \hat{\omega}(k) \end{aligned} \tag{2.16}$$

Therefore, the fault diagnosis (see Table 2.2) can be established by defining two suitable thresholds λ_1 and λ_2 coherent with the measurement noise boundaries.

A well-known method in aerospace FDD is known as multiple model adaptive estimation (MMAE) and uses banks of filters (such as KF or EKF) [11]. This method requires the definition of faults in advance, and each fault is treated by a single filter that delivers the state estimate, covariance, and residual. Figure 2.10 shows the MMAE's architectural layout.

Due to its reactivity in the face of parameter fluctuations, MMAE outperforms approaches without multiple models, enabling quicker fault isolation. By giving equal weight to each KF output, this technique also makes it possible to create a reliable state

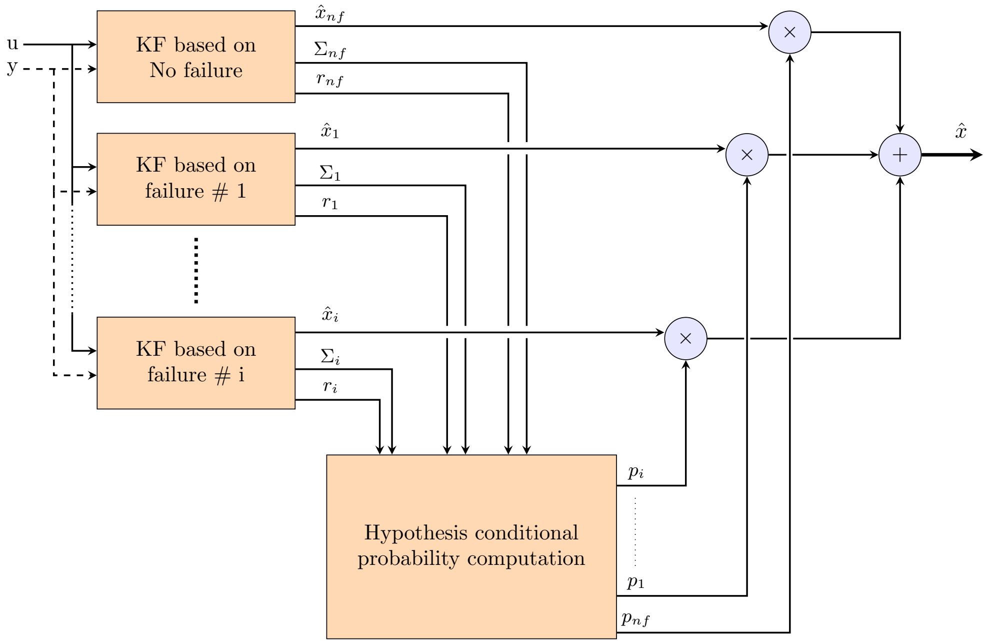


Figure 2.10: Classical MMAE architecture [11].

estimate even in the event of an actuator or sensor failure. On the other hand, MMAE primary drawback is its reliance on the pre-configured fault concept. Additionally, due to the computational load, the number of errors that can be detected is constrained [11].

There might not be a high-fidelity spacecraft dynamics model at the space mission's design stage. This unavailability arises with complicated satellite flexible modes. In other words, if model-based design methodologies were used, the FDD system may not be sufficient to provide a satellite ACS with high efficiency [10]. New FDD strategies must be researched to address that shortcoming. These methods will be detailed in the following subsection.

2.3.2 | Data-driven FDI

2.3.2.1 | Brief literature review

In the context of FDD, data-driven techniques have gained prominence as an alternative to model-based approaches, and numerous articles on this subject have been published in recent years. A significant drawback of implementing Artificial Intelligence (AI) methods in satellite software, as demonstrated in [10], is the increased computational cost compared to parameter and state estimation strategies. This highlights a fundamental challenge associated with adopting AI approaches in satellite systems.

Conversely, model-based techniques rely on a solid foundation of physical knowledge, encompassing the dynamic model and the expected fault model. These considerations continue to drive the ongoing discourse on enhancing these methods. This can involve expanding the capabilities of model-based approaches to accommodate a wide range of fault scenarios or optimizing AI techniques to reduce computational overhead through faster and more efficient algorithms.

Arthur Samuel provides a concise definition of Machine Learning (ML) in [32]: *"ML is a system science technique that can learn from sample input data and extract structural information for developing a model."*. In the context of FDD for satellites, ML algorithms use the datasets created upon the harnessed Telemetry (TM) data. However, the amount of TM data a spacecraft can transmit is limited, as installing sensors for every parameter is not cost-effective and space-efficient. Furthermore, even for the selected TM parameters, the volume of data samples that can be transmitted within a given time frame is constrained by the limitations of communication bitrates.

Figure 2.11 shows ML categorization, where we can divide it into:

- Which learning type is used to obtain the models?
- What task we choose to obtain the desired output.

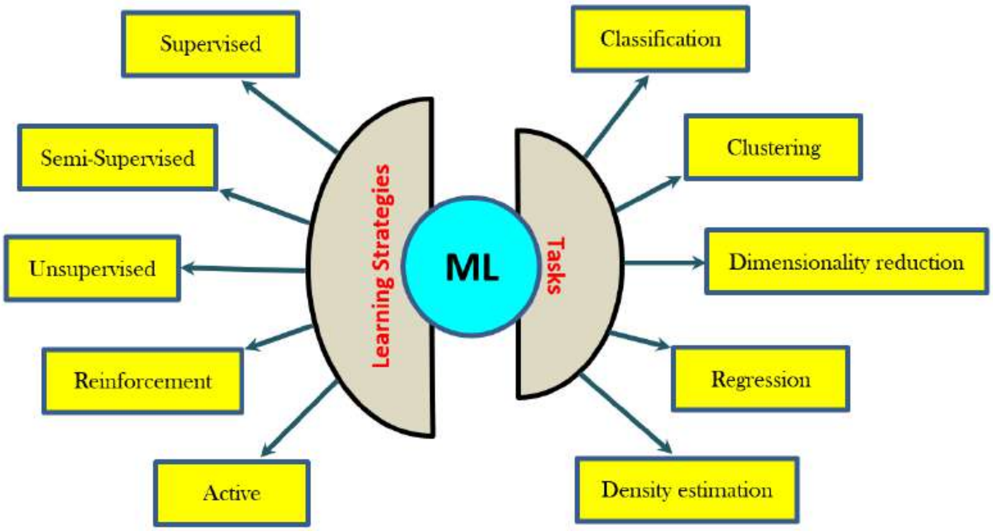


Figure 2.11: Machine learning categorization [10].

In the study conducted by [33], Kernel Principal Component Analysis (KPCA) was applied to a training dataset to investigate the correlations among various parameters related to reaction wheels. The proposed methodology enables fault detection by identifying abnormal variations in testing data, and notably, it doesn't require prior knowledge of the fault model. In their work, [34] examined telemetry data from Egypt's Egyptsat-1 satellite utilizing Support Vector Machine (SVM), focusing on extracting nominal behaviour patterns of satellite components. Following this, they implemented a fault diagnosis scheme utilizing Fault Tree Analysis (FTA) to identify the most probable cause of the 2010 satellite failure. Another approach to data feature extraction for FDD was presented by [35], which employed Supervised Local Linear Embedding (SLLE). This approach was applied to real satellite telemetry data, specifically the Luojia1-01 satellite, and significantly enhanced FDD capabilities. In the work by [36], a novel model-free framework for detecting and isolating faults in RWs was proposed. The framework considered factors such as gyroscopic effects, measurement noise, and aerodynamic disturbances. This research aimed to improve the reliability of RW fault detection and isolation. The mixed learning approach introduced in this context leverages a combination of machine learning techniques, including SVM, Random Forest (RF), Naïve Bayes, and Partial Least Square (PLS), to carry out FDI tasks. During the initial phases of assessing the reliability of spacecraft components, it is common to conduct long-term tests on critical hardware like moment wheel assemblies (MWAs). In the investigation conducted by [37], a data-driven method for monitoring the health of MWAs was introduced. This approach utilizes acoustic parameters and implements a clustering fusion methodology. The primary aim is to scrutinize feature vectors comprising parameters like root mean square (RMS), kurtosis, and sharpness.

In addition to the aforementioned data-driven approaches, neural network (NN) methods offer a robust solution for executing FDI strategies. In the work by [38], a novel FDI approach for detecting faults in RWs bearings is introduced, employing a convolutional neural network (CNN). This approach utilizes time-domain signal data from the vibrations of the rotating object to create input feature maps. The CNN's image recognition capabilities are then harnessed to diagnose bearing faults. An additional investigation by [39] addresses the issue of the swift attenuation of fault effects in closed-loop control systems. Their proposed Fault Detection and Isolation (FDI) strategy hinges on deep neural networks (DNN). Utilizing the sliding window technique, a DNN is employed to discern fault characteristics effectively across diverse data modes.

In a different context, [40] addressed the autonomous FDI of Nanosatellites. Their approach leverages deep learning techniques to detect and identify faults in reaction wheels. They develop a long short-term memory (LSTM) model using training data, and the model processes residuals to provide fault labels to the satellite's CPU. It's worth noting that the validation of most of these approaches is typically based on simulations. Moreover, these approaches frequently assume that faults and disturbances are theoretically boundeIn, in some cassimulation models es, are simplified, concentrating on a restricted range of scenarios, such as minor attitude rotation angles (as seen [7]).

A noteworthy obstacle in the advancement of FDI for spacecraft, particularly within academic circles, lies in the limited accessibility to high-fidelity simulators employed by major space organizations like NASA, ESA, and *Centre National d'Etudes Spatiales (CNES)*. As highlighted by [41], hierarchical fault detection, isolation, and recovery (FDIR) systems are indispensable for ensuring the health management of spacecraft, particularly during critical phases such as orbit insertion for deep-space probes. The FDI methodologies discussed thus far predominantly pertain to the first and second levels of hierarchical FDIR, concentrating on Component-Level FDD. The remaining levels of hierarchical FDIR deal with high-critical faults, particularly in situations with slow data transmission (e.g., large spacecraft-earth distances), and they involve sophisticated techniques commonly used in industry, which may not be within the scope of this research. For further details on the spacecraft's hierarchical FDIR architecture, please refer to Figure 2 in [41] and Figure 7.1 in [42].

It's important to note that certain FDD stages need to be conducted at ground control centres in some Micro and Nano-satellite configurations, especially the latter, which have constraints related to size, computational resources, and energy. There are two prominent strategies for this purpose:

- In many micro-satellite configurations, attitude determination relies on gyro-stellar estimation. For instance, references such as [43, 44] provide insights into this method. However, this approach faces challenges when multiple gyroscopes fail simultaneously and the star tracker is blinded. This situation can significantly increase attitude-pointing errors, triggering a switch to safe mode and potentially interrupting the mission. Ground crew members are then responsible for performing FDI tasks to identify which gyroscope is faulty and initiate the design of a gyro-less configuration, which may involve charges like fine-tuning of Kalman filter gains.

- In the context of interplanetary missions, where communication delays are substantial (e.g., New Horizons experienced a delay of approximately 4.5 hours when close to Pluto), ground-based FDD tasks do not immediately impact mission safety. Consequently, onboard FDIR schemes tend to be quite complex. Conversely, as pointed out by [42], in non-critical missions characterized by high availability rates (commonly observed in Micro and Nano-satellites operating in Low Earth Orbits), ground intervention takes place after the transition to safe mode. The ground crew then conducts a thorough analysis of telemetry data to pinpoint the source of the fault and take necessary actions.

For example, the following section summarizes our data-driven method used for gyroscope FDD as in [45].

2.3.2.2 | Example: Self-adaptive dynamical classification applied for gyro FDI

In this section, we will elucidate our concept for gyro FDD, which centres on monitoring the evolution of Euclidean distances of residuals. Hence, any deviation from the anticipated normal behaviour usually signals a sensor fault. Additionally, statistical analysis of the Euclidean distance evolution is utilized to enhance detection robustness and reduce the chances of inaccurate diagnoses. To evaluate the efficacy of our approach, we incorporate different drift speeds into the satellite attitude control simulator to simulate faults. The results obtained are subsequently compared with alternative machine learning (ML) methods, highlighting the superior performance of our scheme concerning both missed alarm rates and incorrect detection rates. Notably, our approach operates without prior knowledge about the faults in the attitude sensors, making it entirely data-driven. The suggested method comprises three key phases: ***i***) the creation of a feature space, ***ii***) the calculation of drift indicators, and ***iii***) the monitoring and interpretation of drift utilizing a self-adjusting mechanism.

The feature space is formulated using two residuals: gyro-based and SST-based residuals. Both assess deviations concerning the reference rates, denoted as ω_{ref}, as described in Eqs. (2.17) through (2.20).

$$Gyr_{res} = \omega_{ref} - \omega_{gyr} \tag{2.17}$$

$$\delta q = q^{*}_{k-1} \otimes q_k \tag{2.18}$$

where δq is the quaternion error and q_{k-1}, q_k are two successive attitude quaternions delivered by the SST. The asterisk (*) and $\otimes$ symbols denote the quaternion conjugate and multiplication, respectively.

$$\omega_{SST} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_{SST} = 2 \begin{bmatrix} \delta q_2 \\ \delta q_3 \\ \delta q_4 \end{bmatrix} / T_s \tag{2.19}$$

where T_s is the system sampling period.

$$SST_{res} = \omega_{ref} - \omega_{SST} \tag{2.20}$$

Subsequently, we employ a variability-based self-adaptive dynamical classification (VSADC) technique, a dynamic clustering tool designed to handle evolving data. This unsupervised method possesses auto-adaptive capabilities, making it suitable for classification in a broad spectrum of dynamic systems. In the realm of ACS for three-axis stabilization, the typical category of residuals displays concentrated clusters in the proximity of the origin within the feature space. Within this category, the centre typically resides close to (0,0), and the covariance matrix is non-null, attributed to systematic noise. The assumed Gaussian distribution of data noise aligns with contemporary gyro-stellar attitude estimators utilizing Kalman filters, a prevalent technique on satellites such as the Myriad family from the CNES [46].

When new observations, denoted as X_{new}, are introduced, the learning rules are activated, creating and adjusting data prototypes and classes. Furthermore, historical data smoothing, as discussed earlier, serves to reduce noise transmission in the detection channel. To accomplish this, we have integrated sliding windows as a filtering technique [45]. Consequently, adapting prototypes using variability-based self-adaptive dynamical classification (VSADC) entails a recursive update of the centre and covariance matrix within a sliding window of user-defined width.

The occurrence of a fault is signalled by the dissimilarity exceeding a predetermined threshold between the nominal class C_n and the evolving class C_e. To quantify this dissimilarity, we evaluate the distance, as specified in Eq (2.21), between the gravity centres μ_n and μ_e. The drift indicator corresponds to this distance and is continuously updated for each new feature vector (X_{new}) arrival.

$$d_E = \sqrt{(\mu_{n,r_{gyr}} - \mu_{e,r_{gyr}})^2 + (\mu_{n,r_{SST}} - \mu_{e,r_{SST}})^2} \tag{2.21}$$

Where d_E indicates the Euclidean metric. In addition to the earlier discussed Euclidean metric, which measures the gap between the gravity centres, we also consider the variability of this distance, denoted by its standard deviation (σ). Including this statistical feature contributes to a more robust performance in fault identification. To manage this, we introduce a threshold, σ_{lim}, which will be defined later.

Some dynamic classification methods may use a predefined constant value for σ_{lim} (e.g., 3 times the standard deviation of the nominal class distribution [47]). In contrast, our approach dynamically adjusts this threshold based on the occupation areas in the feature space, thereby incorporating self-adaptive characteristics. Nevertheless, this threshold must balance minimizing false alarms and detecting faults, mainly when the drift is gradual. The behaviour of this variance can be understood in two ways:

- **Increase in the Drift Region**: In normal conditions, the variance of residuals remains bounded ($\sigma \leq \sigma_{max\_nom}, \forall \sigma \in \sigma_{nom}$), where σ_{nom} represents the set of standard deviations (nominal case). When a fault initially occurs, σ begins to increase until it surpasses σ_{max_nom}. Therefore, it is wise to set the threshold σ_{lim1} to σ_{max_nom} to help detect faults efficiently and minimise false alarms.

- **Stagnation in Bias-Like Faults**: where it is more efficient to set the new threshold, σ_{lim2}, as the mean of the standard deviations of the recent sliding windows. This approach effectively detects new drifts faster than σ_{lim1}, which may not guarantee the detection of a new fault if the σ of related data is smaller than σ_{lim1}.

Figure 2.12 illustrates the evolution of σ and its impact on the self-adaptation process in our approach. VSADC is outlined in Algorithm 1.

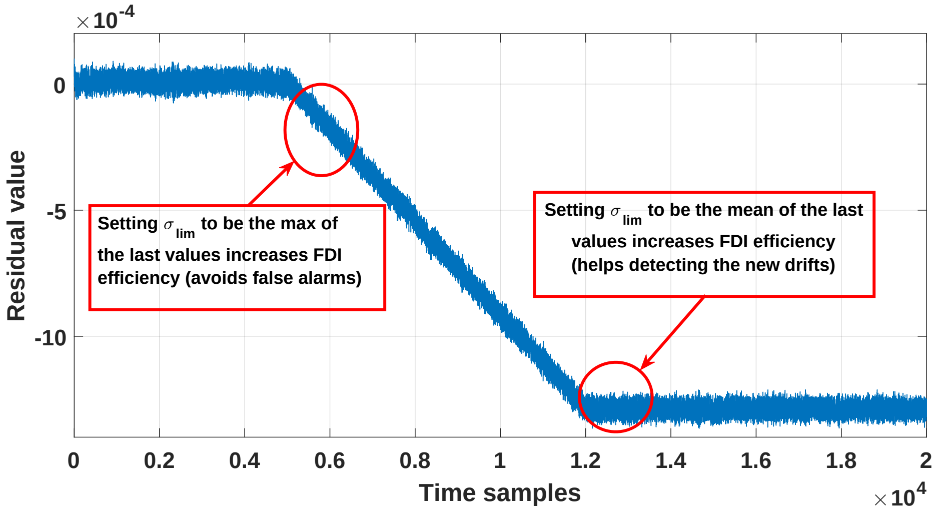


Figure 2.12: Visual interpretation of σ_{lim} selection. [45].

After setting the adequate environment for VSADC testing against famous ML classifiers, the accuracy and labeling comparison for gyro drift then bias faults are depicted in Figures 2.13 and 2.14, respectively.

VSADC outperforms other methods in terms of:

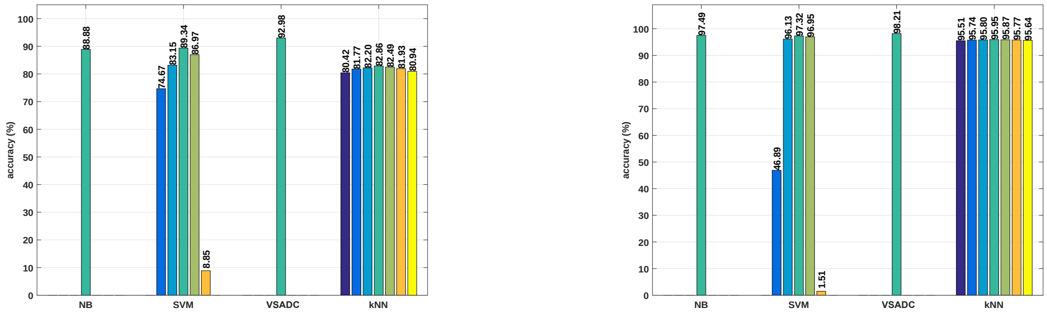
- fast detection,
- accuracy,
- comparing it to Naïve Bayes, it has lower detection noise.

The superior performance of VSADC is attributed to the dynamic adaptation of the standard deviation. This technique effectively mitigates the shattering effect in prediction. In contrast, other methods encounter challenges related to class overlapping during mode transitions (from healthy to fault, fault type 1 to fault type 2, etc.). When the gyro initiates drifting, the gravity centre of residuals shifts accordingly. VSADC addresses

Algorithm 1 Data-driven gyro FDI (VSADC)

Inputs:
Configuration: sliding window width (SW);
k = onboard computation step ;
T_s = sampling rate (e.g. @ 4 Hz)
thr$_{dis}$ = distance threshold;
Outputs:
Gyro state of health (predict):
0: healthy ; 1: drift ; 2: bias.

```
for $axis = 1 : 3$ do
                                          ▷ compute SST-based and GYRO-based residuals
    $\text{res}_{gyr} = \omega_{ref}(k, axis) - \omega_{gyr}(k, axis)$
    $\delta q = q^{*}_{k-1} \otimes q_k$
    $\omega_{sst} = \frac{2*\delta q}{T_s}$
    $\text{res}_{sst} = \omega_{ref}(k, axis) - \omega_{sst}(k, axis)$
    $\text{batch}_{gyr}(k)$ = $\text{res}_{gyr}$
    $\text{batch}_{sst}(k)$ = $\text{res}_{sst}$
    if $k \leq SW$ then
        $\text{nominalbatch}_{gyr}(k)$ = $\text{res}_{gyr}$
        $\text{nominalbatch}_{sst}(k)$ = $\text{res}_{sst}$
        $\mu_{X1,nom}$ = mean($\text{nominalbatch}_{sst}$)
        $\mu_{X2,nom}$ = mean($\text{nominalbatch}_{gyr}$)
    else                   ▷ 1st feature is SST residual; 2nd feature is Gyro residual
        winX1 = $\text{batch}_{sst}(k - SW + 1 : k)$
        winX2 = $\text{batch}_{gyr}(k - SW + 1 : k)$
        $\mu_{X1}$ = mean(winX1)
        $\mu_{X2}$ = mean(winX2)
                                   ▷ compute euclidean distance for evolving prototype
        $d_E = \sqrt{(\mu_{X1} - \mu_{X1,nom})^2 + (\mu_{X2} - \mu_{X2,nom})^2}$
        $\text{batch}_{distance}(k) = d_E$
        $\text{batch}_{\sigma-distance}(k) = \sigma(\text{batch}_{distance}(k - SW + 1 : k))$
                                   ▷ compute $\sigma_{lim}$ for self-adaptation purposes
        $\sigma_{lim1}$ = max($\text{batch}_{\sigma-distance}(k - SW + 1 : k)$)
        $\sigma_{lim2}$ = mean($\text{batch}_{\sigma-distance}(k - SW + 1 : k)$)
        if $d_E \geq \text{thr}_{dis}$ then
            $\sigma = \sigma(\text{batch}_{distance}(k - SW + 1 : k))$
            if $\sigma \geq \sigma_{lim1}$ then          ▷ drift case (1st occurrence)
                predict(k) = 1
            else if $\sigma \geq \sigma_{lim2}$ then     ▷ drift case (continuous)
                predict(k) = 1
            else                                         ▷ bias case
                predict(k) = 2
            end if
        else                                             ▷ nominal case
            predict(k) = 0
        end if
    end if
end for
```



(a) Slow drift.

(b) Fast drift.

Figure 2.13: Accuracy comparison between VSADC and ML classifiers [45].

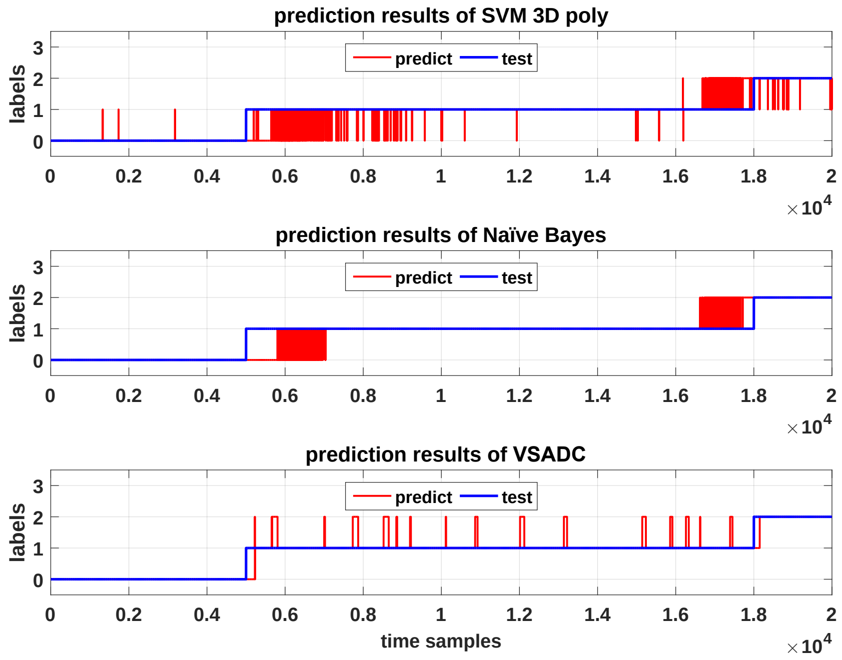


Figure 2.14: Labeling comparison between VSADC, SVM and NB [45].

this evolution by incorporating the variance of data, a feature not present in other methods. Consider NB as an example; its prior probability significantly influences biased classification, particularly in transition zones.

2.4 | Fault tolerant control in ACS

Fault-tolerant control (FTC) is a methodology utilized across various engineering fields to ensure system reliability and performance in the presence of faults or failures. These fields include aerospace control and estimation [48, 10], power systems and electrical engineering [49, 50], automotive engineering [51, 52], and industrial automation [53, 54], to name a few.

As Section 2.3 outlines, elementary faults in the Attitude Control System (ACS) can lead to satellite mission disruptions, necessitating a transition to safe mode or causing other catastrophic damages when adaptive control in response to faults is not implemented. An alternative control concept known as Fault-Tolerant Control (FTC) must be employed to ensure mission continuity with acceptable performance in the presence of faults. FTC systems are designed to achieve satisfactory control performance even when system components experience faults or failures.

FTC can be categorized into two main strategies. The first, referred to as ***passive*** FTC (PFTC), focuses on a priori control design to attenuate the impact of faults through control robustness. PFTC relies on pre-modelled faults, which can result in poor performance when un-modelled faults occur. Additionally, PFTC often necessitates hardware redundancy to achieve its goals, which may not always be feasible, especially in microsatellite configurations.

To address the limitations of PFTC, an alternative strategy known as ***active*** FTC (AFTC) has been developed. AFTC incorporates a Fault Detection and Isolation (FDI) module that locates faulty components and identifies the characteristics of the faults, such as magnitude and severity. Based on the FDI system information, the controller needs to be reconfigured to mitigate the effects of the fault on system performance [42, 55, 56]. A schematic illustrating the concept of AFTC applied to control the satellite attitude is presented in Figure 2.15.

The following factors indicate why it is very challenging to design FTC suitable for application in satellite ACS [42]:

- Fault should be promptly detected to avoid its effect propagation throughout the system. Fault criticality is also a key feature defining the FDD performance,
- Fault must also be well isolated,
- The Presence of noises and uncertainties is inevitable. Therefore, the system should be robust,
- The detection rate must be high enough to cover a wide range of faults,

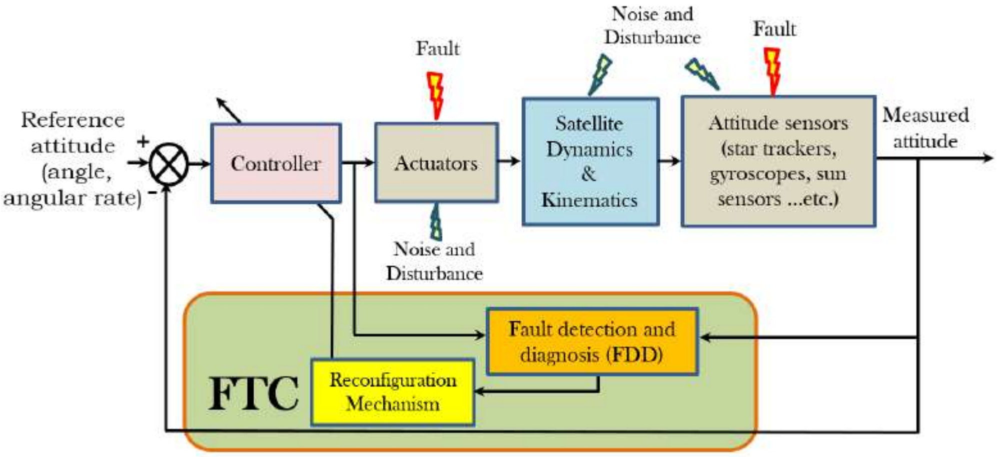

Reference attitude (angle, angular rate)
+
-
Controller
Fault
Actuators
Noise and Disturbance
Noise and Disturbance
Satellite Dynamics & Kinematics
Fault
Attitude sensors (star trackers, gyroscopes, sun sensors ...etc.)
Measured attitude
Fault detection and diagnosis (FDD)
FTC
Reconfiguration Mechanism

Figure 2.15: Active FTC scheme for satellite ACS [10].

- The rate of false alarms should be minimized,
- The fundamental FTC strategy hinges on hardware redundancy, but its implementation is constrained by weight and cost considerations. For instance, in an agile micro-satellite requiring three-axis stabilization, achieving redundancy for angular velocity measurements necessitates the installation of three additional gyroscopes!,
- The robustness of feedback control against noise and uncertainty may inadvertently conceal the faulty behaviour of certain components, thereby diminishing fault detectability.

Furthermore, performing FTC tasks, including fault detection and reconfiguration, often requires autonomous operation in faulty scenarios, especially in Low Earth Orbit (LEO) missions with limited ground station visibility, typically as short as 45 minutes per day. In contrast, for interplanetary or deep space missions, establishing a communication link between the spacecraft and the ground station takes considerable time, posing a significant challenge for rapid response to faults.

Additionally, FTC strategies generally demand more processing power compared to traditional Proportional-Derivative (PD) or Proportional-Integral-Derivative (PID) controllers. This places a considerable burden on satellite computing resources, making the implementation of FTC strategies challenging in terms of computational requirements [10].

As illustrated in Figure 2.16, [55] provides a classification of FTC schemes based on their active or passive nature, to which we introduce artificial intelligence-based approaches as an active form of FTC. These schemes find applicability in general control engineering, including aerospace control systems.

Conversely, AI or data-driven methods represent a potent tool for addressing FTC issues, including FDD. They primarily rely on historical data acquired from sensor and actuator subsystems. Moreover, promptly identifying specific fault types, such as sensor drift, can pose challenges without a thorough examination of historical sensor data. This underscores the importance of employing data-driven methods to address such challenges [47].

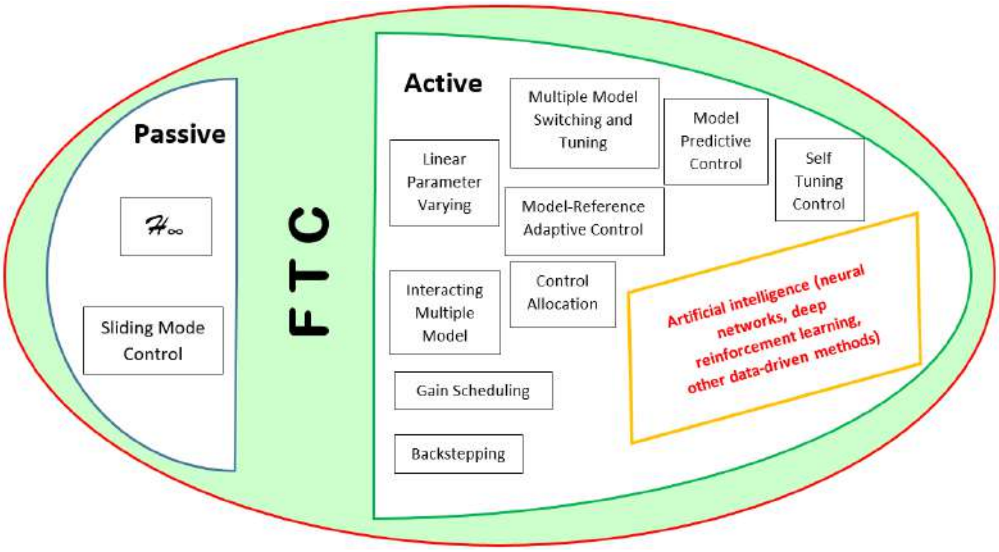

FTC
Passive
$\mathcal{H}_\infty$
Sliding Mode Control
Active
Linear Parameter Varying
Multiple Model Switching and Tuning
Model Predictive Control
Self Tuning Control
Model-Reference Adaptive Control
Interacting Multiple Model
Control Allocation
Gain Scheduling
Backstepping
Artificial intelligence (neural networks, deep reinforcement learning, other data-driven methods)

Figure 2.16: Passive and active FTC classification (adapted from [55]).

We will now explore various FTC schemes presented in recent academic literature over the past years. We will not delve into fault detection and isolation aspects, as those were extensively covered in the previous section.

In the investigation conducted by [57], the task of achieving attitude stabilization for rigid spacecraft under the influence of sensor and actuator faults was addressed. The proposed Integral Sliding Mode Controller (ISMC) was designed to robustly mitigate external disturbances and counteract the effects of both actuator and sensor faults on the system. Another study by [58] delved into a control switching strategy grounded in an adaptive gain fault observer output. Their approach utilized an ISMC-dual-layer gain adaptation controller, leading to improved transient performance in the presence of faults when compared to other gain adaptation methods. [59] addressed the issue of active FTC for satellite ACS dealing with simultaneous actuator and sensor faults. They designed an $\mathcal{H}_\infty$ FTC that demonstrated the capability to precisely stabilize attitude angles and angular velocities in the presence of faults. In [60], a second-order sliding surface is established, demonstrating almost global asymptotic stability at equilibrium. Using this surface, an FTC is developed to address actuator faults and external disturbances in attitude control systems. [61] proposed a modified performance function (MPF) to design a robust adaptive controller, ensuring stable attitude tracking with bounded errors. The proposed controller maintains stability despite external disturbances, thruster faults, and input saturation. In their work, [62] introduced a nonsmooth robust adaptive FTC law capable of operating under zero-momentum and input saturation conditions. This control method necessitates the operational RWs to produce sufficient control torque, ensuring fault tolerance. Meanwhile, the study by [63] revolves around a concurrent-learning controller that autonomously updates itself based on previous control signal values to generate the current control signal. This controller is specifically crafted with an event-triggered policy to overcome data transmission constraints arising from bandwidth limitations in the actuators. Furthermore, a robust fault observer is devised to assist the concurrent learning controller in compensating for actuator faults. [64] introduced two innovative double-loop fault-tolerant controllers for addressing the stabilization of flexible spacecraft's attitude and the mitigation of appendages' vibrations. The inner loop employs fault-tolerant finite-time terminal sliding mode (TSM), while the outer loop utilizes MPC. The stability of the closed-loop systems is rigorously proven using the Lyapunov theorem, even in the presence of actuator faults and external disturbances.

In conjunction with the previously mentioned control strategies, approaches based on Machine Learning (ML) and Artificial Intelligence (AI) exhibit promising capabilities for Attitude FTC. As an illustration, [65] introduced a fault-tolerant learning control approach tailored for Autonomous Helicopter Vehicles (AHVs) when faced with challenges such as actuator faults, external disturbances, and parametric uncertainties. A fixed-time neural disturbance observer is employed to compensate for fault diagnosis errors and external disturbances. The approach also includes an online learning-enhancement mechanism that uses historical data to improve learning accuracy and tracking performance in dealing with parametric uncertainties. The study by [66] proposed a hierarchical FTC framework for over-actuated hypersonic reentry vehicles. The framework incorporates deep learning techniques while it achieves fault tolerance in both the control allocation and control layers. The proposed design also includes an LSTM neural network as a fault diagnosis unit. [67] presented a data-driven approach for estimating actuator faults in aircraft, which, when integrated with an adaptive controller, demonstrates superior performance in handling actuator failures and disturbances compared to existing fault-tolerant controllers. In this scheme, a deep neural network has been trained on a dataset generated from simulations of aircraft with intentionally injected faults. [68] introduced an FTC approach for spacecraft attitude stabilization. It employs a neuro-adaptive estimator to handle disturbances and presents an event-based control strategy that conserves on-board resources while ensuring system stability. An innovative FTC scheme for hypersonic reentry vehicles (HRV) in the presence of complex uncertainties was proposed in [69]. The combination of a command filter and a neural network simplifies controller design, enhances control efficiency, and ensures system stability and convergence. Moreover, an adaptive update law for NN weights was introduced to enhance the neural network's approximation accuracy, leveraging convex optimization techniques. [70] tackled the problem of spacecraft attitude takeover control using cellular satellites with limited communication capacity. The proposed method employs a hysteresis quantizer, NN approximation (where a Radial Basis Function NN aids in understanding and predicting spacecraft behaviour), and distributed control allocation to address communication limitations, unknown spacecraft dynamics, and control allocation issues. In their work, [71] presented an innovative attitude-tracking control strategy utilizing reinforcement learning for combined spacecraft takeover manoeuvres. It effectively deals with unknown and complex system dynamics by employing Q-learning and policy-iteration techniques. It also introduces a Q-learning procedure based on online policy iteration. This method allows Bellman's optimality equation solution to be used by measurement data, resulting in a model-free control strategy. To robustly track the attitude trajectory during spacecraft manoeuvring, [72] proposed a control strategy employing a quasi-sliding-mode methodology. The aforementioned control law was integrated with a three-layer back-propagation NN to enhance energy efficiency and reduce actuators' power consumption. This NN aids in auto-tuning the control gains.

Another critical aspect that demands attention when seeking effective FTC schemes pertains to optimization. When faults affect the actuators or sensors of spacecraft ACS, it becomes imperative to optimize the attitude control signals, considering various factors, including:

- The remaining fuel reserves, for instance, hydrazine, in the event of faulty angular rate sensors, as they are primarily employed during orbit manoeuvres,

- Ensuring that the spacecraft's actuators, such as RWs, operate without reaching saturation,

- Taking into account the electric budget when additional load is imposed on electrical actuators, like magneto-torquers and Control Moment Gyroscopes (CMG),

- In specific missions, such as Mars/Moon landers or hypersonic vehicles, optimizing the spacecraft's reentry trajectory is essential [73, 74].

2.5 | Summary

In this chapter, FDD/FTC methods that can be applied in satellite attitude determination and control have been thoroughly reviewed. These methods were classified mainly based on their nature *w.r.t* physical knowledge (i.e., model-based and model-free or data-driven). At this juncture, it is pertinent to underscore certain limitations of data-driven strategies. These methods grapple with challenges stemming from the absence of actual failure data required for constructing classification-based models. Additionally, data-driven approaches necessitate access to representative data for each fault behaviour, which can be challenging. Often, real data acquired from spacecraft are not accessible for academic research purposes, affecting both the quantitative and qualitative aspects of training data. During the lifespan of a spacecraft, creating a comprehensive database of failure scenarios can be infeasible due to factors such as component reliability, security and safety constraints, or exorbitant costs associated with obtaining such data.

Furthermore, data-driven approaches are essentially black-box models, meaning we cannot interpret their output in a manner that would allow for certification or an explanation of the underlying causes. This limitation is particularly critical for systems like satellite attitude control, where a deep understanding of the system's behaviour is crucial. An alternative approach to circumvent these significant drawbacks is the adoption of model-based methods. The latter offers a physical interpretation and enables them to verify their output formally because they rely on the application of physical principles governing the system's dynamics and behaviour. These techniques, widely used in aerospace engineering and extensively documented in the literature, continue to be the prevailing contemporary tools for designing onboard FTC/FDD.

Alternatively, to address the limitations of both model-based and data-driven methods, we can create a hybrid model combining the strengths of both approaches. This concept has been explored in several studies [75, 76, 77]. In this hybrid model, a physical model is initially constructed based on the available information on dynamics and system configurations. Subsequently, this model is continuously enriched and strengthened over time by incorporating incoming data related to the system's operational conditions and environmental factors. The primary contributions of this dissertation centre around this concept, as will be elucidated in the ensuing chapters.

3

Attitude FTC using data-driven FDI

3.1 | Introduction

As previously demonstrated, maintaining precise satellite orientation significantly enhances the quality of the end product, whether it pertains to Earth surface imaging or various other commercial or military applications. Achieving this precision relies on the effective interplay of two key concepts: attitude determination and attitude control.

In the realm of attitude determination, two primary categories emerge static approaches and estimation techniques. Static approaches involve processing observations in both frames (reference and body) to compute the attitude matrix, typically relying on at least two vectors (as exemplified by the TRIAD algorithm [78]). These methods were developed to address the well-known Wahba's problem, encompassing scenarios with any vector observations [79]. Notable techniques within this category include Shuster's QUEST [78] and Davenport's q method [80].

However, static approaches often neglect the presence of noise inherent in sensor measurements. To address this limitation, state estimation techniques incorporate stochastic variables into the mathematical framework [2]. Methods referred to as "attitude estimation" leverage a spacecraft's dynamic model within a filter that processes data from multiple measurements taken over time. Modern filtering techniques, particularly Kalman filters, play a pivotal role in attitude estimation by filtering noisy observations and estimating attitude and gyro biases [2].

The control supervision scheme outlined in this chapter is inspired from [81] and encompasses two essential subsystems: data-driven FDD and fault-tolerant reconfiguration. When compared to traditional ACS, these subsystems work together to yield superior results. Our approach represents an enhanced version compared to the one presented in Section 2.3.2.2, where we augment the data-driven system with a gradient-sensitive measure, thereby improving the accuracy of gyro FDD. Subsequently, the reconfiguration system adjusts the filter and controller gains using a discontinuity-avoidance technique to provide more suitable control signals.

Chapter 3 is organized as follows. It begins with an in-depth overview of the utilization of Kalman filters as sensor fusion techniques for attitude estimation. It then delves into the details of our proposed approach, shedding light on both FDI and FTC subsystems. The chapter concludes by presenting and discussing the results obtained when applying our technique to stabilize the satellite's attitude in the presence of gyro faults.

3.2 | Attitude determination based on Kalman filter

Since its pioneering application in the NASA Apollo program, the KF has evolved into one of the most influential and widely recognized methodologies for state estimation. It operates as a model-based technique, providing an optimal estimation of the state, expressed in terms of mean and variance, which recursively diminishes with each time step. The KF proves particularly well-suited for Attitude Determination (AD) tasks in microsatellites (μsat) due to its relatively low computational requirements. Typically, the Kalman Filter is designed for sensor fusion in systems that incorporate two types of sensors: the SST and gyroscopes. In this context, the filter's primary objective is to estimate the spacecraft's attitude, often represented as a unit quaternion and the gyro drift. The estimated gyro drift is then subtracted from the raw gyro measurements to yield a precise estimate of angular velocity. This filter configuration is commonly referred to as Gyro stellar estimator (GSE) [46, 43].

3.2.1 | Mathematical foundation

The open literature widely describes sensor fusion of gyroscopes and star trackers using optimal state estimation algorithms. GSE has proven its optimality for attitude determination tasks and is extensively implemented on flying spacecraft [46, 44, 82]. As outlined in [46, 83], the filter state is given by quaternion of attitude and gyro drift. The gyro model and linear filter equations will be given in what follows.

3.2.1.1 | Gyro measurement model

A widely used gyro measurement model is given by [82] as follows:

$$\begin{aligned} \tilde{\omega} &= \omega + d + \nu_g \\ \dot{b} &= \nu_b \end{aligned} \tag{3.1}$$

where $\tilde{\omega}$ and ω d denote the measured and true rate. d is the gyro drift. ν_g and ν_b are two independent zero-mean white Gaussian noises. Please note that a drift model is broadly assumed to be unavailable; this point will be further discussed in the following paragraph.

3.2.1.2 | Prediction

In the GSE, the filter state is chosen to be a 6x1 vector, replacing the quaternion with an error rotation vector. During the filter propagation step, the transition equation uses approximation techniques for the -considered small- error rotation vector. The prediction is then given by Eq. (3.2).

$$\begin{aligned} \hat{q}_{k/k-1} &= \hat{q}_{k-1/k-1} \otimes \delta q_k \\ \hat{d}_{k/k-1} &= \hat{d}_{k-1/k-1} \end{aligned} \tag{3.2}$$

with:

$$\delta q_k = \begin{bmatrix} 1 - \frac{\|\theta_k\|^2}{2} \\ \frac{\theta_k}{2} \end{bmatrix}$$

$$\theta_k = (\omega_{gyr} - \hat{d}_{k-1/k-1}) \times T_s$$

and: θ_k is the attitude error angle, ω_{gyr} is the gyros measurement, T_s is the filter time step. Note that the evolution model is usually unavailable, so the predicted drift is the last estimated one.

3.2.1.3 | Innovation

As stated in [84], one can define the error angle to be twice the quaternion vector part. Thus, the innovation calculation as described by [46] yields

$$innovation = \theta_k - \theta_{k/k-1} = 2 \times \overrightarrow{\hat{q}_{k/k-1} \otimes q_{SST}} \tag{3.3}$$

where q_{SST} is the quaternion delivered by the Satellite Star Tracker (SST). In Eq (3.3), the ($\overrightarrow{...}$) mark denotes the last three components (vector part) of quaternion multiplication result.

3.2.1.4 | Update

A constant Kalman gain is chosen to relax the computation based on the spacecraft's mechanical structure and sensor characteristics. This gain is carefully computed during the early design phase. In our case, the correction term is given by Eq (3.4).

$$\begin{bmatrix} \theta\_cor_k \\ d\_cor_k \end{bmatrix} = \mathbb{K} \times innovation \tag{3.4}$$

where $\mathbb{K}$ is a 6×3 matrix of pre-computed optimal Kalman gain. The formulation of quaternion error in [84] is beneficial and helps us to get the following:

$$q\_cor_k = \frac{1}{2} \begin{bmatrix} \sqrt{4 - \|\theta\_cor_k\|^2} \\ \theta\_cor_{k,X} \\ \theta\_cor_{k,Y} \\ \theta\_cor_{k,Z} \end{bmatrix} \tag{3.5}$$

Finally, the state update is given by Eq (3.6).

$$\begin{aligned} \hat{q}_{k/k} &= \hat{q}_{k/k-1} \otimes q\_cor_k \\ \hat{d}_{k/k} &= \hat{d}_{k/k-1} - d\_cor_k \end{aligned} \tag{3.6}$$

Subsequently, we will demonstrate the filter's performance in both nominal and faulty scenarios. This endeavour aims to provide insights into the impact of faults on ACS behaviour and rationalize integrating alternative methods alongside the Kalman Filter (KF) to mitigate excessive degradation of overall performance.

3.2.2 | Case study of KF performance (μsat configuration)

To assess the performance of GSE, we conduct tests using authentic satellite data (sourced from the Algerian space mission ALSAT-2B, as illustrated in Figure 1.1). Figure 3.1 presents this dataset, showcasing both the attitude quaternion and the angular rates provided by the SST and gyros, respectively. Upon examining the data presented in Figure 3.1, it becomes evident that no discernible faults affect the sensors, with the observed fluctuations attributed solely to inherent measurement noise. The GSE expounded upon in Section 3.2.1, has yielded the quaternion estimation outcomes, as depicted in Figure 3.2. Figure 3.2 illustrates the performance of the GSE in the scenario involving sensors in good health. In this case, the filtering process effectively mitigates the impact of sensor noise. As elaborated in the preceding chapter (refer to Section 2.2), these estimates of attitude, when combined with the reference trajectory programmed by the mission planning team, are used in computing pointing errors. Subsequently, these pointing errors are fed into a control scheme to calculate the torques necessary for achieving attitude stabilization.

As elucidated in [10], faults are inevitable when dealing with ACS. Figure 3.3 provides an overview of the fault rates associated with various components of AOCS.

Figure 3.3 reveals a noteworthy observation: gyroscopes contribute to 17 % of component failures, a relatively substantial portion. As a result, as we explore in Section 3.5,

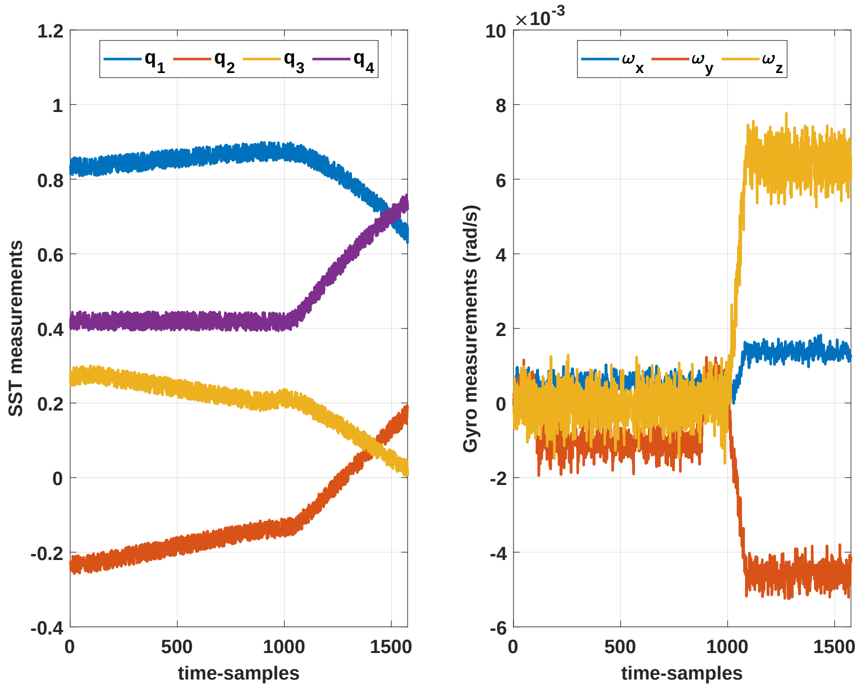


Figure 3.1: Gyro-stellar input measurements.

the influence of Gyro faults on system performance is considerable and warrants thorough examination. In the same section, we will also examine how a Gyro bias can impact the Kalman filter performance and how our approach proves advantageous in such scenarios.

3.3 | Proposed data-driven method for gyro diagnosis

A fixed Kalman gain is adopted for drift estimation in compact spacecraft to reduce computational complexity. Nevertheless, abrupt drift in a gyroscope can give rise to convergence challenges for the estimator in this setup. Furthermore, employing higher gains can exacerbate noise propagation. Typically, opting for a drift gain satisfying the condition $drift\_gain_{Kalman} \ll attitude\_gain_{Kalman}$ strikes a favourable balance. The data-driven approach introduced in this chapter is structured into three sequential steps, as outlined below.

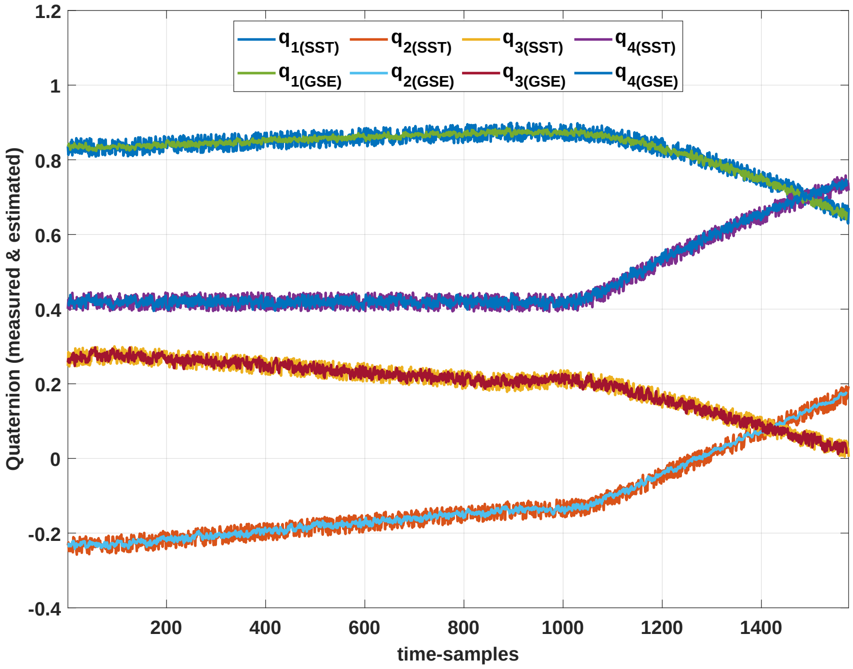


Figure 3.2: GSE output (healthy sensors).

3.3.1 | Residual definition

Angular rate residuals are defined as the disparity between reference values and the gyros output, expressed in Eq (3.7).

$$r = \omega_{ref} - \omega_{gyr} \tag{3.7}$$

The estimation of faults directly from these residuals, as computed using Eq (3.7), is susceptible to the influence of sensor intrinsic noise and controller efforts. Consequently, it is imperative to employ an effective smoothing and filtering technique to enhance the performance of FDI.

3.3.2 | Residual pre-processing

When considering the residuals derived from Eq (3.7) in their unprocessed state, they can be processed further to extract information characterized by reduced variance. The suggested approach involves data filtering through the real-time mean value calculation. This mean value corresponds to the average of the data within a moving window.

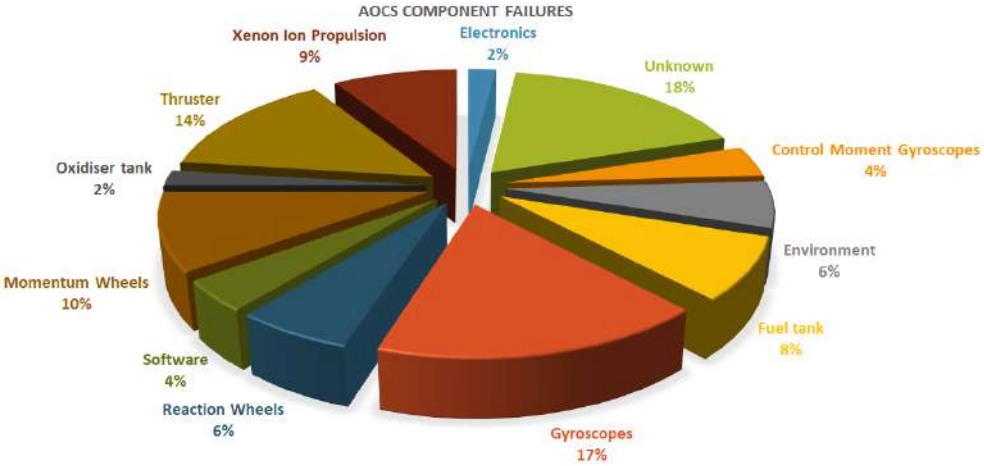


Figure 3.3: Failure rates of AOCS parts [85].

Theorem. *Let us explore the resemblance between two aspects:*

- *The historical pattern of residuals, influenced by the interplay of closed-loop dynamics and sensor noise,*

- *A prototypical under-damped second-order system.*

Consequently, detecting a sudden bias-like fault can be accomplished by employing a sliding window approach on the unprocessed residuals, as calculated using Eq (3.7)*. The optimal width for this window, which ensures rapid fault detection with minimal variance and overshoot, corresponds to the pseudo-period of the system's step response.*

Proof. Drawing an analogy to second-order under-damped linear time-invariant (LTI) systems, the closed-loop system, which is affected by noisy measurements and experiences a sudden fault, will exhibit residuals that follow the typical step response shape depicted in Figure 3.4. The step response of an LTI system with characteristics such as a damping ratio ζ, undamped natural frequency ω_n, and damped natural frequency $\omega_d = \omega_n\sqrt{1-\zeta^2}$ is given by [86]:

$$y_1(t) = 1 - \frac{\omega_n \exp^{-\alpha t}}{\omega_d} \sin(\omega_d t + \phi) \tag{3.8}$$

Mathematically, the average value of a function f over an interval $[a, b]$ is given by [87]:

$$av_{[a,b]}(f) = \frac{1}{b-a}\int_a^b f(x)\,dx \tag{3.9}$$

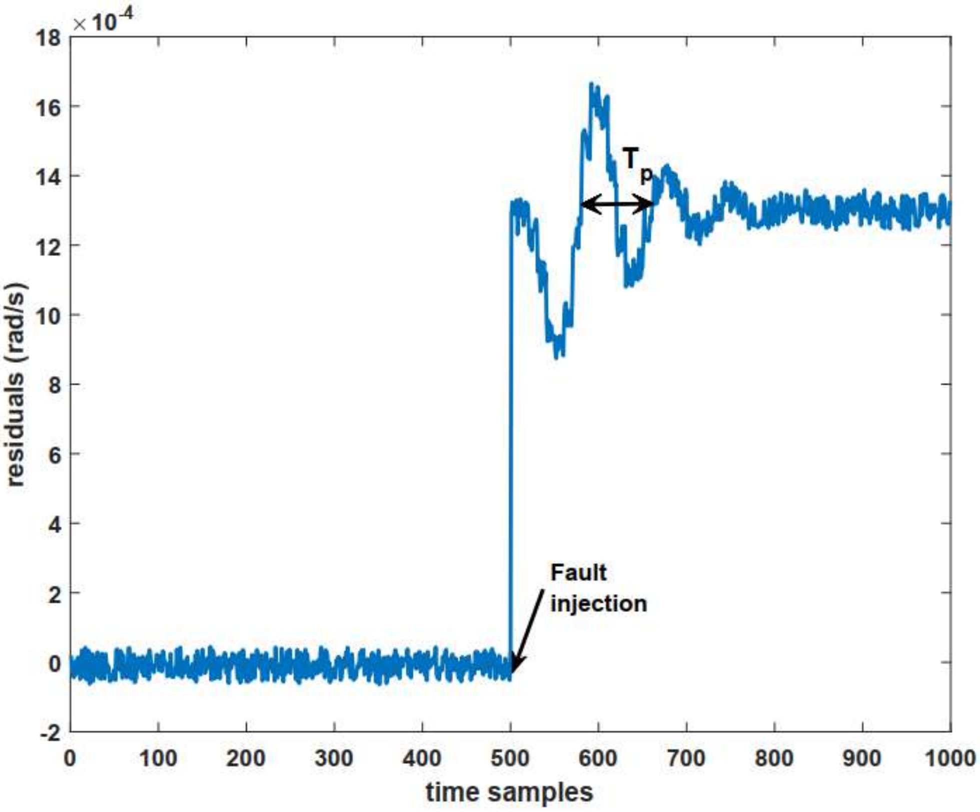


Figure 3.4: Raw residuals conditioned by abrupt fault injection.

Applying the expression above on sine function yields

$$\int_a^b \sin(x)\,dx = -\cos(b) + \cos(a) \tag{3.10}$$

It is of note that the exact restoration of an abrupt fault value necessitates that the mean value of the step response is null. Consequently, through substitution, Eq (3.10) infers the following:

$$\cos(a) = \cos(b) \tag{3.11}$$

In light of the periodicity inherent in the cosine function, Eq (3.11) remains valid for all $b = a + (n \times T_p)$. However, the expeditious restoration is assured when $n = 1$. □

Moreover, the pseudo-period, denoted as T_p can be ascertained directly from the closed-loop transfer function, as illustrated in Figure 3.5 and detailed in Eqs (3.12) to (3.15).

$$y = (K_p + K_d s)(\frac{1}{Is^2})(x - y) \tag{3.12}$$

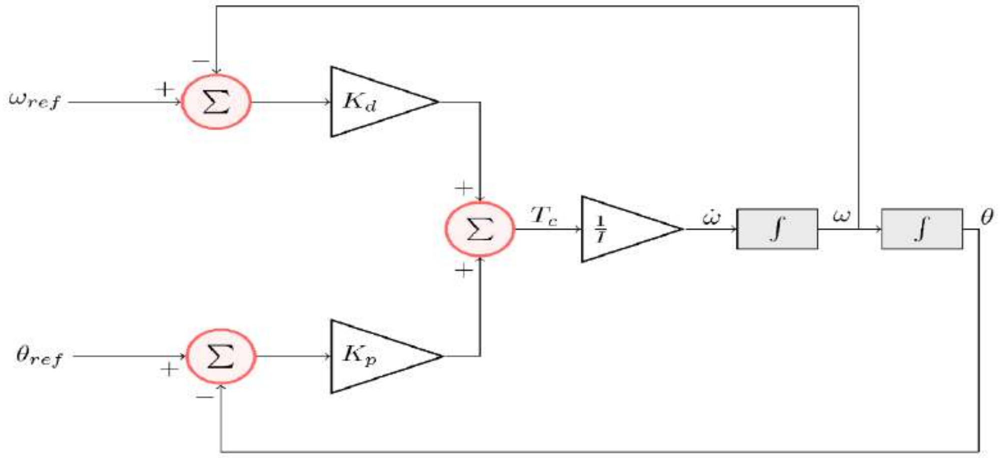


Figure 3.5: Closed-loop dynamics for attitude stabilization.

$$TF = \frac{y}{x} = \frac{\frac{K_d}{I}s + \frac{K_p}{I}}{s^2 + \frac{K_d}{I}s + \frac{K_p}{I}} \tag{3.13}$$

$$den(TF) = s^2 + 2\zeta\omega_n s + \omega_n^2 \tag{3.14}$$

$$T_p = \frac{2\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{3.15}$$

Conducting a simulation with the following numerical values: $K_p = 0.6; K_d = 0.9; I_{xx} = 14.5$, yields the outcomes depicted in Figure 3.6. This figure shows that the magnitude of the injected fault equates to $5 \times 10^{-3} rad/s$. The most favourable outcomes are obtained when employing a sliding window width equal to $T_p = 31.25 sec$ (as calculated using the aforementioned equations). For smaller window sizes, the processed residuals exhibit heightened sensitivity to the steep slopes in the raw values, amplifying the variance. Furthermore, the utilization of larger windows results in a delay in the identification process, primarily because residual changes are detected later. It is important to highlight that, in alignment with the theorem outlined earlier, the most effective reduction in variance occurs at T_p and its multiples (i.e. nT_p, with $n = 1, 2, \ldots$). Nevertheless, opting for a window size of T_p ensures prompt detection.

3.3.3 | variability-gradient based self-adaptive and dynamical classification

In this research endeavour, we have conceived the Variability Gradient-Based Self Adaptive and Dynamical Classification (VGSADC) as a model-free algorithm [81]. The core

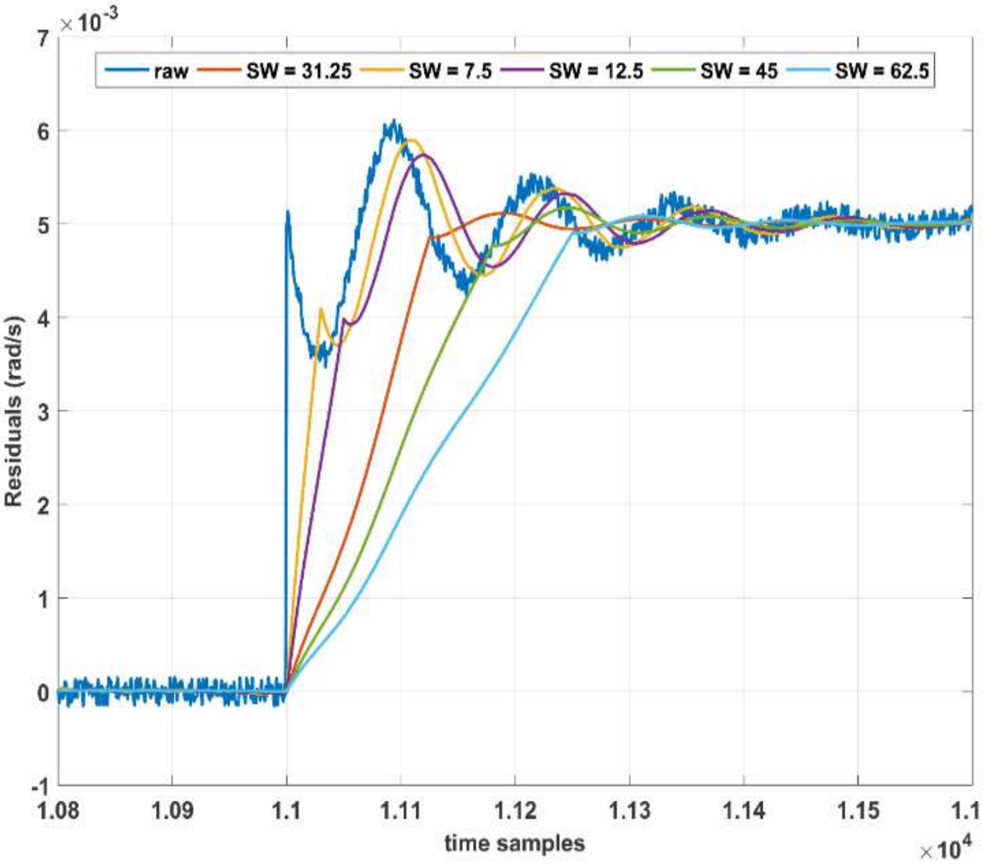


Figure 3.6: Residual preprocessing comparison.

objective of VGSADC is detecting and identifying gyroscope faults based on meticulously processed residuals. Within this algorithmic framework, we incorporate a gradient-sensitive indicator in conjunction with the optimal sliding window (as detailed in the preceding subsection) to augment the performance of fault identification.

In our approach, the term "dynamical" implies that the selection of data for FDI is continually updated in order to minimize variance as soon as any abrupt changes in residuals cease to manifest. Specifically, the estimated fault transition shifts from historical minimum and maximum values to the mean of raw residuals whenever the gradient or standard deviation of the data falls below predefined thresholds. Furthermore, using the gradient-sensitive indicator facilitates the classification of faults into two distinct categories: high- and low-gradient. Consequently, the FTC system is engaged in reconfiguration in accordance with the prevailing fault class. The key procedural steps of VGSADC can thus be succinctly summarized (refer to Figure 3.7):

- ***Detection***: the detection of a faulty condition occurs promptly when the standard deviation of the residual data, which is of the same length as the sliding window, surpasses the predefined threshold.,
- ***Classification***: in order to enhance the accuracy of fault identification, VGSADC assigns a gradient indicator to the data. This indicator aids in promptly identifying faults with minimal delay and estimation errors,
- ***Identification***: following the categorization of faults into high or low-gradient classes, a more precise estimation of the fault can be achieved.

The pseudocode outlining the steps of FDI is illustrated in Algorithm 2. The comprehensive procedural details of the VGSADC algorithm can be delineated as follows:

- Calculate the width of the sliding window using Eq (3.15),
- Initialize gradient and residual batches for all axes,
- Set the fault flags ($FFlag$) for all axes to **False**,
- At the current time-step, compute the residual value (according to Eq (3.7)) and calculate the residual gradient, updating the corresponding batches,
- For axes where $FFlag$ is **False**, if the standard deviation of the batched residuals ($std(batch_{res})$) is greater than or equal to σ_{lim}, then set FFlag to **True** and record the minimum and maximum values of the batches,
- For axes with $FFlag$ set to **True**, conduct a two-level check as follows:
  - Examine whether the fault is in a steady state or not (by checking if $std(batch_{res})$ is greater than or equal to σ_{lim}),
  - Assess whether the fault exhibits a high gradient or not (by comparing the maximum and minimum gradients to a certain threshold $\pm\nabla_{lim}$).
- Based on the results of the preceding checks, assign fault values for each case, utilizing either the maximum/minimum values or the mean of the current residual batch for identification and classification purposes.

Algorithm 2 Data-driven gyro FDI (VGSADC)

```
Input: controller gains ($K_p$,$K_d$);
satellite inertia ($I$); $\sigma_{lim}$; $\nabla_{lim}$
Configuration: sliding window width SW = $f(K_p, K_d, I)$;
k = onboard computation step ;
ts = sampling rate (e.g. @ 4 Hz)
Initialization: Initialize local_batch for all axis
$FFlag \leftarrow [False, False, False]$
for $axis = 1 : 3$ do
    $r(k, axis) = \omega_{ref}(k, axis) - \omega_{gyr}(k, axis)$
    $grad(axis) = (r(k, axis) - r(k-1, axis))/ts$
    $batch_{res}(k, axis) = r(k, axis)$
    $batch_{grad}(k, axis) = grad(axis)$
    if $size(batch_{res}(1 : k, axis)) \geq SW$ then
        if $\neg FFlag(axis) \wedge \sigma(batch_{res}(k - SW : k, axis)) \geq \sigma_{lim}$ then
            Set Fault detection flag
            $FFlag(axis) \leftarrow True$
            $gradmax(axis) = \max(batch_{grad}(k - SW : k, axis))$
            $gradmin(axis) = \min(batch_{grad}(k - SW : k, axis))$
            $resmax(axis) = \max(batch_{res}(k - SW : k, axis))$
            $resmin(axis) = \min(batch_{res}(k - SW : k, axis))$
        end if
        if $FFlag(axis)$ then
            if $\sigma(batch_{res}(k - SW : k, axis)) \geq \sigma_{lim}$ then
                if $gradmax(axis) \geq \nabla_{lim} \vee gradmin(axis) \leq -\nabla_{lim}$ then
                    fault with high gradient (fast detection)
                    if $abs(gradmax(axis)) \geq abs(gradmin(axis)$ then
                        $fault(axis) = resmax(axis)$
                    else
                        $fault(axis) = resmin(axis)$
                    end if
                else
                    fault with low gradient (mean of raw residuals)
                    $fault(axis) = E(batch_{res}(k - SW : k, axis))$
                end if
            else
                steady state fault
                $fault(axis) = E(batch_{res}(k - SW : k, axis))$
            end if
        else
            $fault(axis) = 0$
        end if
    end if
end for
```

3.4 | Reconfiguration and Fault tolerant control

To control the attitude of ALSAT-2B and other satellites within the Myriad satellite family [46, 44], ACS designers implement a PID control scheme. It's noteworthy that, in practice, the integral term is often excluded to prevent excessive actuator saturation, particularly in the presence of external perturbation torques.

Moreover, when confronted with substantial initial pointing errors, actuator saturation becomes a recurring issue attributable to the linear Proportional-Derivative (PD) control law. To address this challenge, a switching approach is outlined in [8, 88], which alternates between the linear PD control law and a non-linear ***speed bias*** control. This transition is triggered when the pointing error surpasses a predefined threshold, with careful consideration given to minimizing control discontinuity. Equation 3.16 delineates the specifics of this switching strategy [81].

$$\begin{cases} T_{c,i} = -(\delta\omega_i + \omega_d sign(\delta\theta_i)), if|\delta\theta_i| > \theta_L \\ T_{c,i} = -(K_\theta \delta\theta_i + K_\omega \delta\omega_i), if|\delta\theta_i| \leq \theta_L \end{cases} \tag{3.16}$$

where i denotes the axis number $i \in 1, 2, 3$. Parameters θ_L and ω_d are chosen to guarantee control performance and convergence. The convergence during the speed bias regime implies that T_c equals zero. K_θ and K_ω are then chosen to ensure control continuity at the switching (when $\delta\theta_i = \theta_L$). For example, the numerical values for the DEMETER spacecraft are: $\theta_L = 0.3°$, $\omega_d = 0.015°/s$, $K_\theta = 0.1$ and $K_\omega = 2$ [8]. In our proposed approach, the reconfiguration process is initiated promptly upon detecting and identifying a faulty gyroscope output by the VGSADC algorithm. This reconfiguration encompasses adjustments to both the (GSE) and the controller.

3.4.1 | GSE reconfiguration

When a sudden change occurs in the gyroscope output, it leads to an increased disparity between the predicted and measured states, as evident in Eqs. (3.2) and Eq. (3.3). Consequently, the Kalman filter faces the challenge of divergence, primarily due to the elevated innovation values. To mitigate this divergence and enhance convergence, the estimator is provided with a new estimated initial condition derived from the drift estimates delivered by the FDI. This step replaces the data-based model with an inadequately defined drift model, as depicted in Eq. (3.1). The resulting system adopts a hybrid approach that leverages both model-based and data-driven techniques [81].

It's worth noting that the updated gains in the GSE are contingent on the spacecraft's configuration. As elucidated in [89], smaller gains in the Kalman filter serve to attenuate noise transmission. Conversely, larger gains enhance tracking performance, making it swifter. The algorithm adapts the GSE by updating the drift to facilitate rapid tracking. Subsequently, during abrupt (fast) changes, the algorithm increases the attitude gain, placing greater confidence in SST readings. In our approach, system designers must store these gains in a Coarse reconfiguration database (CRD). The term ***coarse*** implies that the AOCS team should subsequently upload more precise values. The online reconfiguration in our approach aims to minimize satellite downtime while ensuring acceptable performance.

However, it's important to acknowledge two primary limitations of the proposed method:

- The data-driven filtering scheme, employing a sliding window, introduces a certain level of detection delay. This delay, however, does not excessively impact attitude control performance due to gain adaptation. Additionally, the fault magnitude is filtered to reduce transients, which contributes to improved fault identification,

- The proposed approach necessitates the definition of the CRD beforehand, which in turn requires additional time during the design phase.

3.4.2 | Fault tolerant control

Motivated by the switching control law described in Eq. (3.16), our FTC framework hinges upon a strategic elimination of the derivative gain (K_d) subsequent to the issuance of a reconfiguration directive by the supervisory system. This deliberate action aims to avert substantial control discontinuities brought about by the transient fluctuations in gyro measurements. It is imperative to emphasize that the gain reconfiguration remains effective until the restoration of accurate drift estimation by the GSE. Nonetheless, ensuring the concurrent adjustment of the proportional gain (K_p) is imperative to uphold the seamless continuity of control. Herein, let us denote $[\delta\theta_S, \delta\omega_S]$ as the angle and rate errors at the point of switching, leading to the Eq. (3.17) below:

$$\begin{cases} T_{S,i} = -K_p \delta\theta_{S,i} - K_d \delta\omega_{S,i} \\ T_{S,i} = -K_{p,S} \delta\theta_{S,i} \end{cases} \tag{3.17}$$

In this equation, $T_{S,i}$ signifies the commanded torque, and $K_{p,S}$ represents the updated proportional gain computed as follows:

$$\begin{cases} (K_{p,S} - K_p)\delta\theta_{S,i} = K_d \delta\omega_{S,i} \\ K_{p,S} = K_p + K_d \dfrac{\delta\omega_{S,i}}{\delta\theta_{S,i}} \end{cases} \tag{3.18}$$

It is noteworthy that the gains in the aforementioned equations are represented as vectors of dimensions 3 × 1, although, for brevity and clarity, the specific subscripts corresponding to these vectors have been omitted.

Upon the convergence of the GSE with respect to drift estimation, signifying the cessation of abrupt changes, our FTC scheme orchestrates the reconfiguration of the controller by instituting a new derivative gain. The proportional gain is reinstated to its prior value. In this context, let $[\delta\theta_{SB}, \delta\omega_{SB}]$ symbolize the angle and rate errors at the switching-back, and we maintain the same strategy of continuous control:

$$\begin{cases} T_{SB,i} = -K_{p,S}\delta\theta_{SB,i} \\ T_{SB,i} = -K_p \delta\theta_{SB,i} - K_{d,SB}\delta\omega_{SB,i} \\ (K_{p,S} - K_p)\delta\theta_{SB,i} = K_{d,SB}\delta\omega_{SB,i} \\ K_{d,SB} = (K_{p,S} - K_p)\dfrac{\delta\theta_{SB,i}}{\delta\omega_{SB,i}} \end{cases} \tag{3.19}$$

Here, $K_{d,SB}$ denotes the new derivative gain implemented commencing from the switch-back. This orchestration serves to maintain the stability and effectiveness of the control system under dynamic conditions.

Our hybrid supervision system aims to reconfigure the GSE to optimize filter convergence. Also, it reduces control signal discontinuity by updating the controller gains. The proposed scheme involves a two-part process:

- data-driven FDI subsystem for gyro fault detection and coarse identification,
- AOCS reconfiguration, namely adjustment of GSE and controller.

The hybrid supervision system is depicted in Figure 3.7.

3.5 | Numerical simulation results

The MATLAB environment (see Appendix. A) has been employed to validate our approach using ALSAT-2A in-orbit telemetry data. This telemetry dataset encompasses information related to gyroscope drift values, which indicate the system's health and

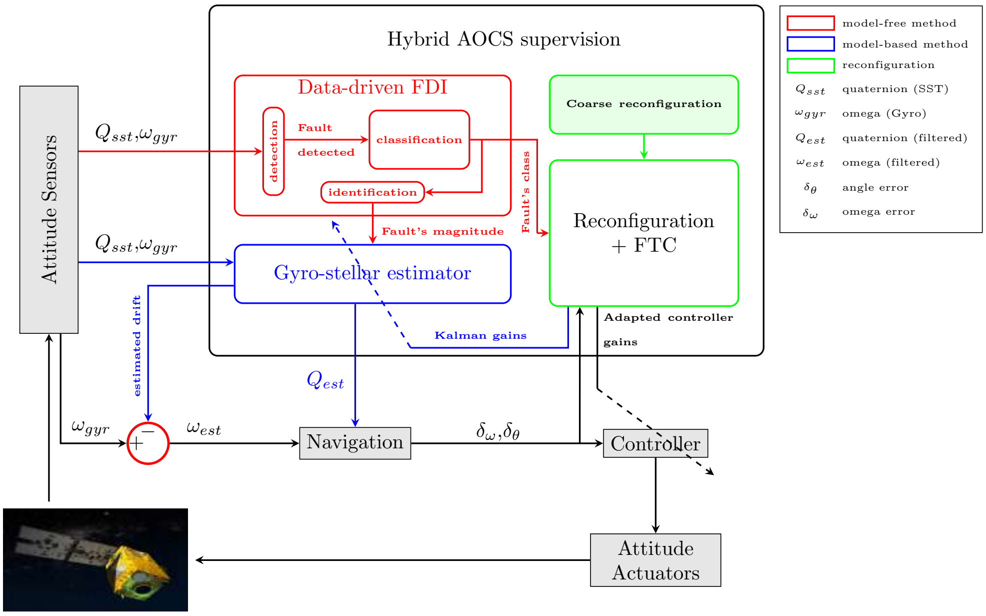


Figure 3.7: Global scheme for AOCS supervision.

ageing. Additionally, it includes the AOCS reference trajectory, specifically represented by quaternion and angular rate data, which is planned and uploaded by the control centre.

In the context of fault simulation, a noteworthy scenario is the sudden alteration of the X-axis gyroscope drift. This change is simulated to transition from its initial values at the beginning of the mission to values characteristic of ten years later, as visually depicted in Figure 3.8. The reference trajectory, capturing various mission profiles such as heliocentric pointing and imaging manoeuvres, is elucidated in Figure 3.9.

Further details regarding the AOCS configuration and CRD data can be found in Table 3.1. It's essential to note that while the CRD controller parameters are presented in Table 3.1, our proposed FTC scheme is designed to facilitate automatic gain switching. Consequently, it should be underscored that in real-life scenarios, backup security measures should always be taken into account. In other words, it is prudent to assign CRD controller parameters for takeover in the event of an adaptation anomaly, ensure the system's continued reliability and safety.

3.5.1 | Fault diagnosis results

In this section, we elucidate the fault diagnosis process, which is managed locally by the VGSADC algorithm and overseen globally by the hybrid supervision system. The

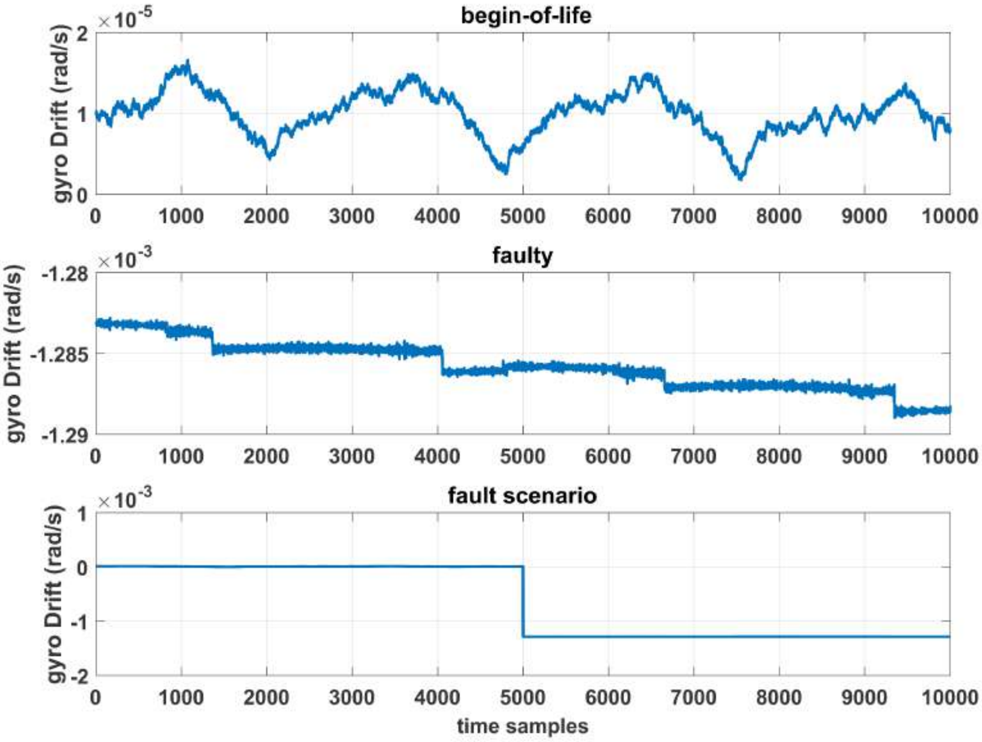


Figure 3.8: Gyro fault scenario.

Table 3.1: AOCS and CRD parameters.

Gains	AOCS		CRD	
	Controller	GSE	Controller	GSE
$K_p(Nm/rad)$	0.2		0.7	
$K_d(Nms/rad)$	0.7		0.2	
$Drift(s^{-1})$		10^{-3}		10^{-3}
$Attitude$		0.66		0.95

former, the VGSADC algorithm, is responsible for conducting the initial estimation of gyro drift, while the latter, the hybrid supervision system, takes charge of adapting both the system's state and the gains of the GSE.

To effectively demonstrate the efficacy of our approach, we have selected three distinct GSE configurations, each serving a unique purpose:

- The first configuration represents the nominal GSE, characterized by the initial gains set at the beginning of the mission,

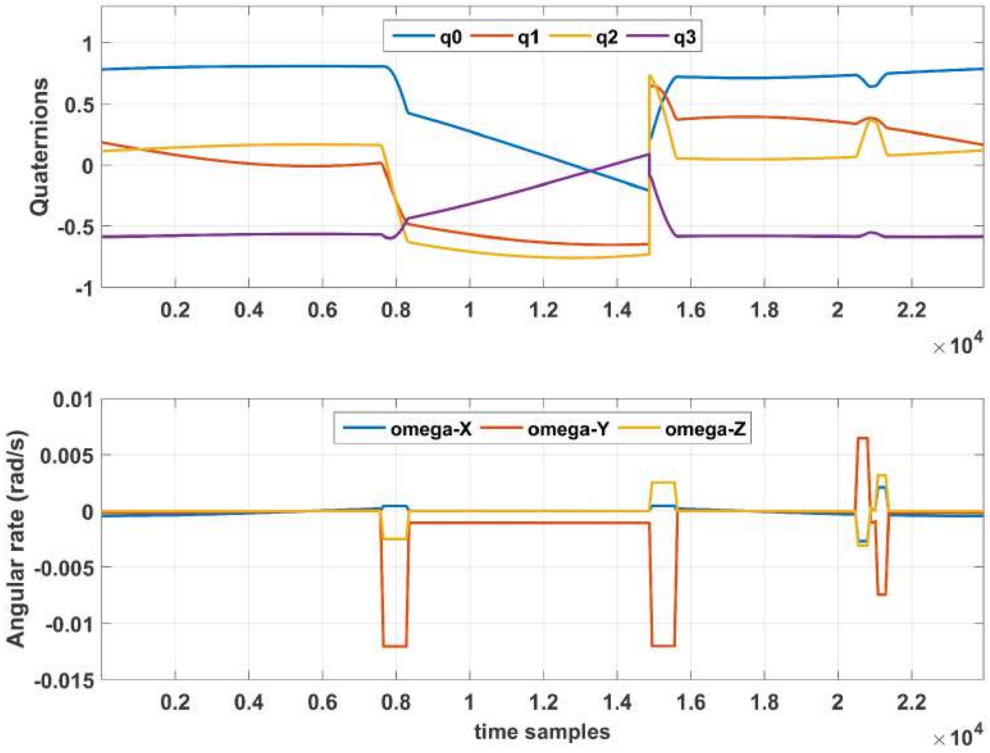


Figure 3.9: Reference trajectory.

- The second configuration involves the GSE with elevated drift gain, tailored to facilitate rapid change detection,
- The third configuration embodies our hybrid scheme, which combines local and global fault diagnosis strategies.

As described in the residual preprocessing section, determining an optimal sliding window width is imperative, and it is set equal to the pseudo-period of the closed-loop system dynamics. Upon substituting the pertinent input data, namely controller gains and satellite inertia, into Eqs. (3.13) through (3.15) , we arrive at the following expressions:

$$\omega_n = \sqrt{\frac{K_p}{I}} = \sqrt{\frac{0.2}{14.5}} = 0.1174 rad/s \tag{3.20}$$

$$\zeta = \frac{K_d}{2\omega_n I} = \frac{0.7}{2 \times 0.1174 \times 14.5} = 0.2055 \tag{3.21}$$

$$T_p = \frac{2\pi}{0.1174\sqrt{1 - 0.2055^2}} = 54.66 sec \tag{3.22}$$

The pseudo-period T_p is subsequently employed as the sliding window width within the VGSADC algorithm. As previously mentioned, the utilization of T_p serves to enhance the tracking of abrupt faults by mitigating oscillations in the estimation process and curtailing the delay in fault identification.

Upon confirming a fault occurrence, our proposed supervision system undertakes the adjustment of GSE gains, specifically by elevating the attitude gain from 0.66 to 0.95. This increase in attitude gain plays a pivotal role in diminishing angle errors, thereby enhancing overall control performance. Additionally, the provision of raw estimates of gyroscope faults serves the purpose of stabilizing the filter, particularly during phases of high gradient fault occurrences. The outcomes of drift estimation are graphically illustrated in Figure 3.10.

Evidently, employing our hybrid data-driven/Kalman system presents a favourable compromise between swiftly obtaining estimations with higher gains, albeit with some degree of noise, and the slower estimations achieved with lower gains. This judicious trade-off strikes a balance between accuracy and speed in the estimation process.

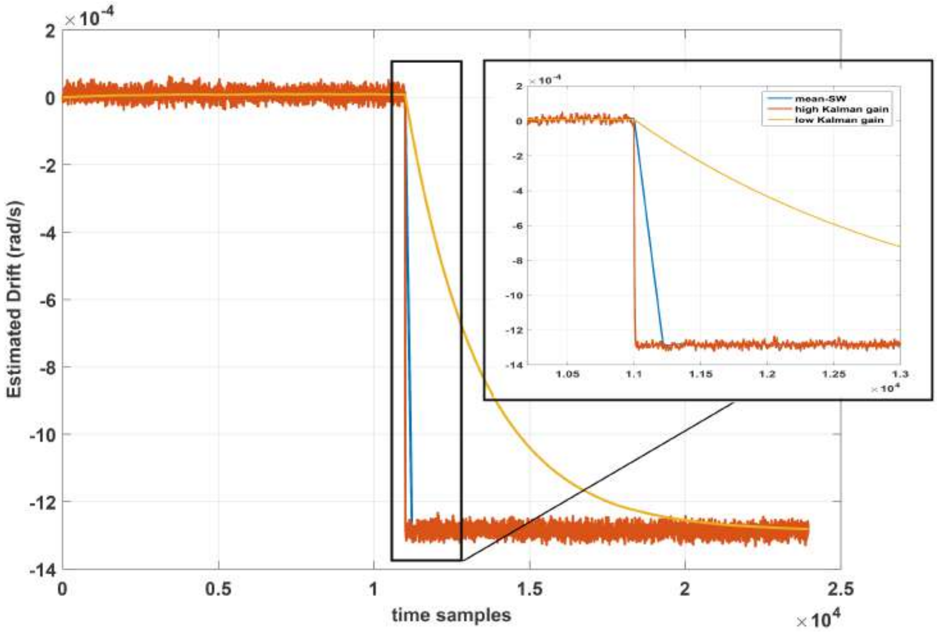


Figure 3.10: X-gyro fault estimation results (compared with KF).

Moreover, the accuracy of fault diagnosis is rigorously validated through a comparative analysis of results obtained via two distinct techniques:

- The first technique involves the use of a basic sliding window,

- The second technique employs our VGSADC (Gradient Enhanced) approach.

Figure 3.11 visually depicts the estimation errors generated by both techniques. It is readily apparent that the basic sliding-window technique exhibits a steeper slope in the evolution of fault estimation. This pronounced slope, indicative of a higher gradient, imposes an undesirable delay in the fault estimation process. Consequently, this approach requires a longer duration to achieve a complete fault estimation.

In contrast, the hybrid scheme, combining VGSADC with the reconfigured GSE, effectively mitigates this delay by reducing it by 130-time samples, bringing it down from 11170 to 11040. This substantial reduction in delay unequivocally underscores the superiority of our proposed method. The combined attributes of accuracy and speed in fault identification render the overall Fault tolerant control scheme autonomously capable of mitigating pointing errors, as detailed in the subsequent subsection.

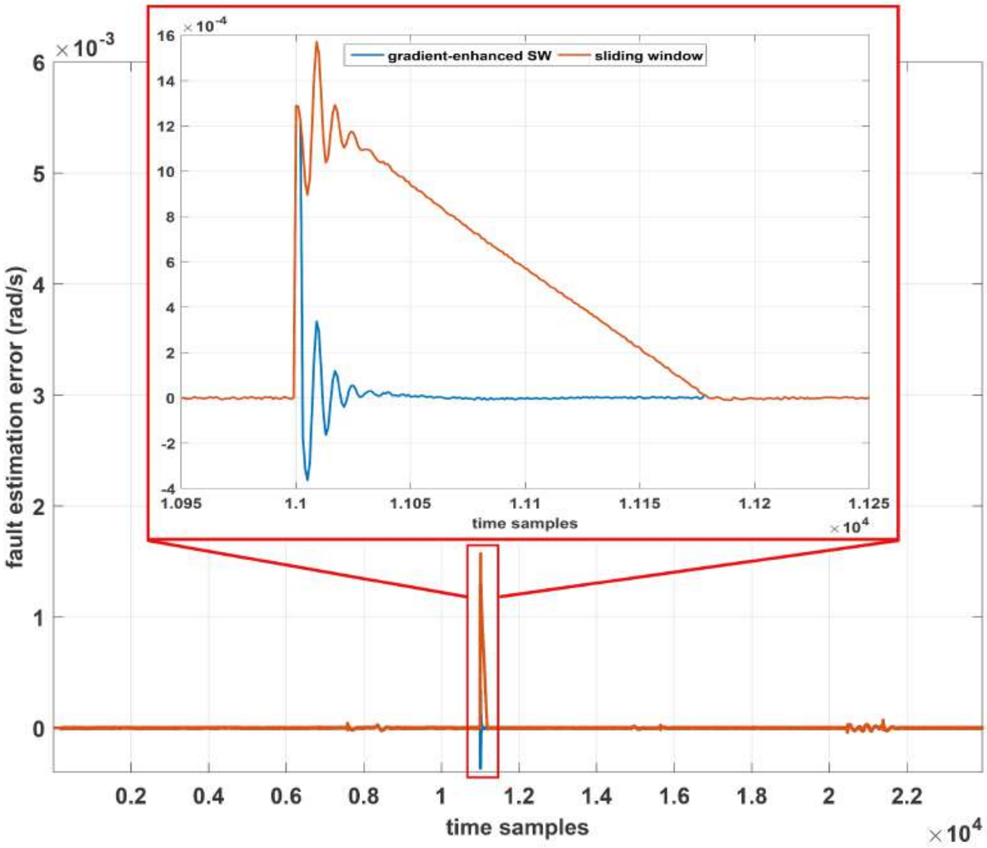


Figure 3.11: Comparison of estimation error (VGSADC vs basic-SW).

3.5.2 | Fault tolerant control results

Following fault detection and accommodation, the supervisory system adjusts the controller gains with the aim of minimizing control effort. Table 3.2 provides a comparison of controller gains before and after adaptation, while Figures 3.12 and 3.13 illustrate pointing errors and control signals, respectively. Typically, pointing accuracy is maintained at 0.1°, which is crucial for evaluating the AOCS performance of small satellites. To put this into perspective, at an altitude of 700 Km, a pointing error of 0.1° corresponds to an in/cross-track error of 1.22 Km on the Earth's surface. A narrower window of 0.01° is adopted for more stringent pointing requirements.

Figure 3.12 highlights that adapting the controller/GSE yields the most accurate pointing results. Additionally, incorporating Kalman gain adaptation provides a slight improvement in performance. The settling time, defined as the time required for the pointing error to fall within the selected accuracy window (0.01°), is a critical metric. A settling time limit of 100 s is set to facilitate comparison, beyond which other dynamic phases start to influence the comparison.

Table 3.2: Adapted controller gains.

Gains	**Value**	**Observation**
Before fault		
$K_p \quad (Nm/rad)$	0.2	nominal gain
$K_d \quad (Nms/rad)$	0.7	nominal gain
Fault initially detected		
$K_d \quad (Nms/rad)$	0.2	source: CRD
$K_{p,s} \quad (Nm/rad)$	[0.86; 0.85; 0.84; 0.86]	Adaptation sim [8;7;4;3]
Fault fully estimated		
$K_p \quad (Nm/rad)$	0.2	back to nominal gain
$K_{d,sb} \quad (Nms/rad)$	[0.61; 0.36; 0.64; 0.36]	Adaptation sim [8;7;4;3]

Table 3.3 presents settling times for all possible configurations, including those with and without GSE/controller adaptation. The table also addresses the impact of state (drift) feeding using our data-driven method. It's evident that GSE gain adaptation alone, without updating the state, fails to achieve the best performance. Our hybrid system achieves superior performance, integrating an augmented GSE through data-driven FDI and controller adaptation. Additionally, adapting the Kalman gain leads to slight improvements in the results, such as a 0.25 s improvement between ***sim8*** and ***sim4*** and a 1.25 s improvement between ***sim6*** and ***sim2***. The worst-case scenarios are

observed in simulations without state feeding.

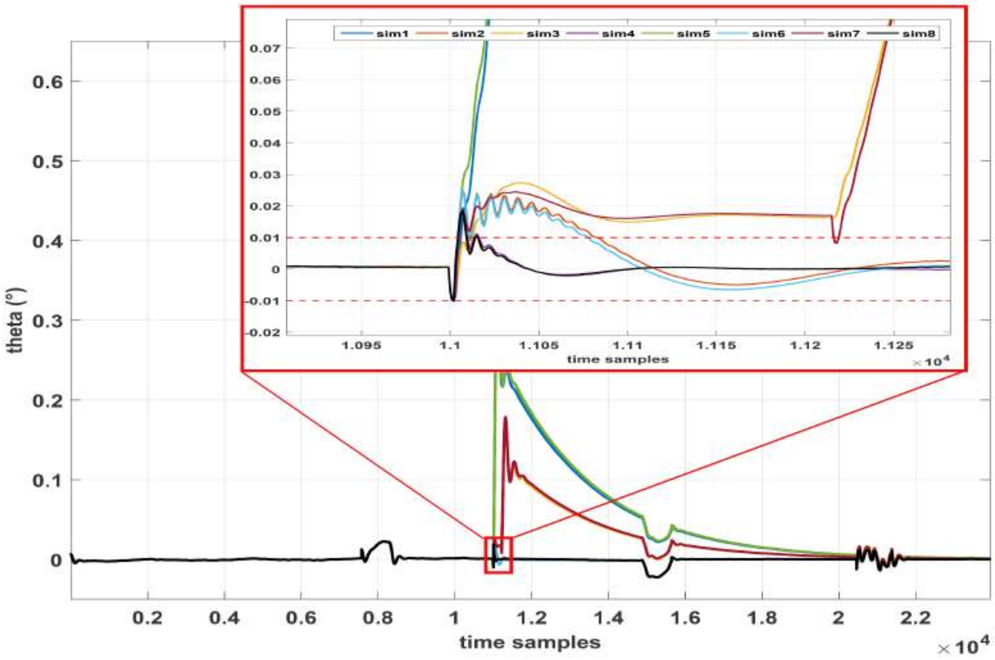


Figure 3.12: Pointing error results with our hybrid system.

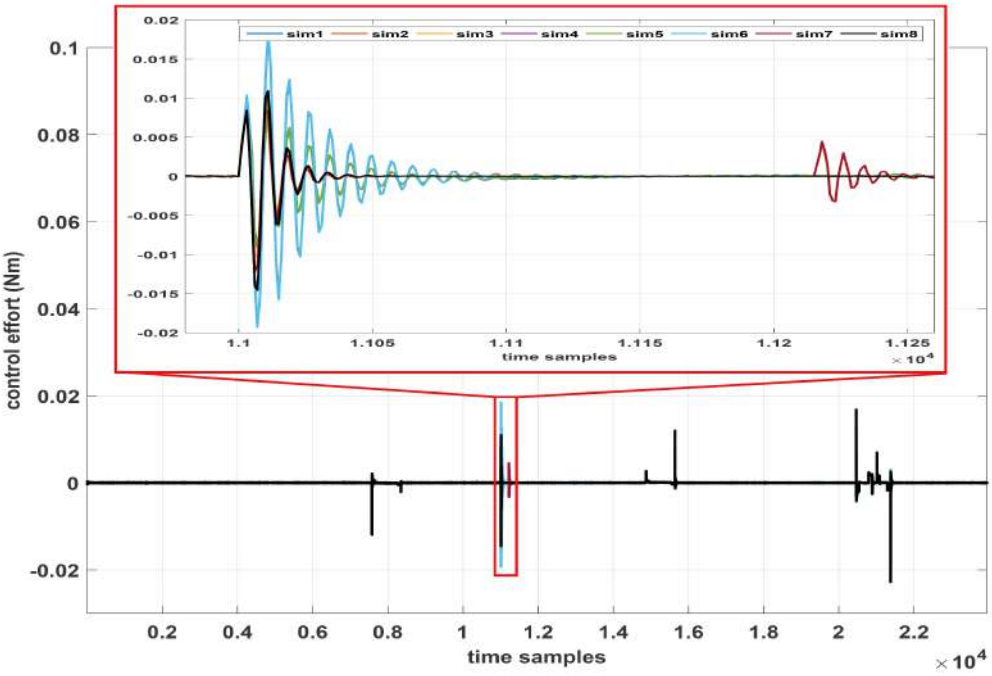


Figure 3.13: Control effort results with our hybrid system.

Table 3.3: Settling time results.

Configuration			Code	Settling time (s)
Adapted Kalman	Adapted controller	State-feeding	sim8	**4**
		No state-feeding	sim7	> st. limit
	Controller not-adapted	State-feeding	sim6	19.5
		No state-feeding	sim5	> st. limit
Kalman not-adapted	Adapted controller	State-feeding	sim4	4.25
		No state-feeding	sim3	> st. limit
	Controller not-adapted	State-feeding	sim2	20.75
		No state-feeding	sim1	> st. limit

3.6 | Summary

In this chapter, we have presented a hybrid supervised attitude determination and control system. This system comprises a data-driven fault detection and identification module followed by reconfiguration and fault-tolerant control. The former module is responsible for providing initial coarse fault estimates, while the latter initiates reconfiguration operations using both these fault estimates and pre-defined parameters from the coarse reconfiguration database.

Our approach considers previous knowledge about the system's dynamics, especially when bias faults occur in angular rate sensors. In such instances, the closed-loop response behaves as a second-order system, leading to sinusoidal residuals. Given that our proposed scheme relies on applying sliding windows (SW) to raw data, we need to analyze further how the optimal SW width can be determined. In this context, we have utilized the average value definition of an arbitrary function to demonstrate that the ideal SW width corresponds to the pseudo-period T_p of the system's step response.

Furthermore, our proposal's fault-tolerant control (FTC) component emulates real system behaviour by implementing a careful switching strategy to prevent control discontinuities. Moreover, the reconfiguration mechanism adapts the estimator gains initially stored in a predefined database.

In the simulation section, we introduced Gyro fault scenarios by simulating sensor readings at the beginning and end of their operational life. The proposed scheme was applied to the fault-augmented attitude control system (ACS), resulting in improved performance in the case of gyro faults. The achieved pointing error using our approach surpasses that of the beginning-of-life configuration with fixed estimator/controller gains and exhibits reduced noise compared to a Kalman filter with higher gains.

4

EGE-RL: improving the RL algorithm with application on attitude FTC

4.1 | Introduction

This chapter will introduce the RL paradigm, one of ACS's most modern intelligent control strategies. RL has primarily addressed decision-making problems in the discrete world, where states and actions are countable and can be tabulated. Like most industrial processes, this dissertation addresses a continuous control problem with infinite state and action spaces.

Numerous industrial processes can be modelled as Markov decision processes (MDPs), which vary from simple control problems to systems with a high-dimensional state space [90, 91]. The RL paradigm is one of the most active study fields for the MDP solution. The action space for most control problems, including attitude control, is continuous. In contrast, the fundamental RL paradigm focuses on discrete action spaces like games. Q_Learning is commonly employed for this objective [92]. Utilizing the so-called actor-critic structure, Deep deterministic policy gradient (DDPG) and Twin Delayed DDPG (TD3) were the pioneers in handling continuous action space in the RL paradigm [93, 94].

This chapter introduces a novel method aimed at enhancing Actor-Critic (AC) learning, with the primary goal of expediting the training process while striving to attain more optimal policies. This approach considers the similarity between the output generated by the RL agent and the control signal derived from a conventional (model-based) controller. Both the RL and classical controllers are provided with the same environment state. Subsequently, this similarity metric becomes integral to the fundamental Markov decision process (MDP) reward, facilitating a guided exploration throughout the continuous action space.

Chapter 4 is structured as follows. Initially, it provides a mathematical overview of MDPs and the RL paradigm. It then delves into the details of the proposed guided shaping based on reference control. The chapter concludes by presenting and discussing the outcomes obtained when applying our agent to control the satellite attitude within the MDP framework.

4.2 | Markov decision processes

MDPs are a mathematical framework widely employed in decision-making under uncertainty. They provide a rigorous foundation for modelling and solving problems that involve sequential decision processes with stochastic outcomes. MDPs are characterized by a set of key components:

- ***States ($\mathcal{S}$):*** MDPs comprise a finite or countably infinite set of states, denoted as S, which represent the possible situations or configurations of the system at discrete time steps,

- ***Actions ($\mathcal{A}$):*** A set of actions, denoted as $\mathcal{A}$, represents the available choices or decisions that an agent can make at each state. Actions are chosen based on a policy,

- ***Transition Probabilities ($\mathcal{P}$):*** The transition probabilities, represented by $\mathcal{P}$, describe the likelihood of transitioning from one state to another when a specific action is taken. These probabilities capture system stochastic behaviour. However, the MDP paradigm also covers the system's particular case of *deterministic* evolution. Alternatively, $\mathcal{P}$ captures the system dynamics generally modelled as differential equations in the control theory language.

- ***Rewards ($\mathcal{R}$):*** At each state, the agent receives a numerical reward, denoted as R, as a consequence of its action. The agent always looks at how to maximize its cumulative reward,

- ***Policy (π):*** A policy (π) maps states to actions and guides the agent's decision-making. It specifies the strategy that the agent follows to select actions at each state,

- ***Discount Factor (γ):*** The discount factor, γ ($0 \leq \gamma < 1$), determines the importance of future rewards relative to immediate rewards. It models the agent's preference for short-term versus long-term gains.

The primary objective in a given MDP is how an optimal policy can be found, denoted as π^*, that maximizes the expected cumulative reward, often referred to as the "return" or "value," over an infinite time horizon. The value of a state, denoted as $V(s)$, represents (when following the optimal policy starting from that state) the expected cumulative reward. Similarly, the value of a state-action pair, denoted as $Q(s, a)$, is the expected cumulative reward when the agent takes the action 'a' in state 's' and follows the optimal policy after that [95, 90, 91].

4.3 | Reinforcement learning

Reinforcement Learning (RL) stands as a robust algorithmic framework in which an agent, often referred to as the controller, acquires a suitable control strategy known as a "policy." This policy is designed to fulfil specific objectives through a series of interactions with its surrounding environment, typically denoted as the controlled system or plant [95]. In the realm of RL, agents can undergo training in either their real environment or a simulated representation thereof [96, 97]. The latter option is commonly employed in the context of spacecraft control problems.

Within each episode, defined as a sequence of steps starting from an initial state and concluding at a terminal state, the RL agent systematically gains an understanding of the environment's state. Subsequently, armed with this information and guided by the current policy, the agent picks an action, essentially the control signal, executed within the environment. Following the execution of this chosen action, the environment transitions to a new state and produces a scalar reward signal. The reward and the representation of the ensuing state are subsequently conveyed back to the agent. The conclusion of an episode occurs when the agent achieves its predefined task, such as maintaining the rotational velocity of a spacecraft below some predefined limit after launcher separation (stumbling manoeuvre), or when certain constraints, like retaining the pointing error ($\delta\theta$) inferior to some threshold, are violated (see [98, 99]).

Subsequent to the episode's conclusion, a new episode can be initiated by randomly configuring the initial state variables and conditions of the environment. For example, this may entail setting parameters such as an asteroid's rotational speed or the terrain's shape in a lander-powered descent experiment [100]. Leveraging the information after the execution of each episode, RL agents endeavour to compute a policy maximizing the expected cumulative reward or minimizing the expected cumulative cost. This process forms the core of RL-based control strategies in complex dynamic systems.

To facilitate the learning process, it is imperative to establish a well-crafted reward (or cost) function that encapsulates the desired performance objectives of the agent. This function effectively evaluates the agent's accomplishments following interactions with the environment, particularly in achieving its designated task. Additionally, selecting an appropriate RL method or algorithm becomes crucial to effectively process the information amassed during the training [96, 95].

As a conventional formalization of problems involving sequential decision-making, Markov Decision Processes (MDPs), as elucidated in the preceding section, provide a clear and structured framework for RL approaches. The agent's policy, denoted as $\pi(\cdot|s) : \mathcal{S} \rightarrow \Delta_{\mathcal{A}}$, represents a mapping from states to a probability distribution, denoted as $\Delta_{\mathcal{A}}$, encompassing the entire action space $\mathcal{A}$. This policy entirely defines the control strategy. It is worth noting that this definition accommodates deterministic policies as well, represented as $\pi(\cdot) : \mathcal{S} \rightarrow \mathcal{A}$.

4.3.1 | Value functions

Figure 4.1 illustrates the interaction between the agent and the environment, modelled as an MDP [95]. Within a specific episode and at each discrete time step t, the agent receives an observation or state from the environment $s_t \in \mathcal{S}$. Subsequently, the agent selects an action a_t sampled from the distribution characterized by $\pi(\cdot|s_t)$, denoted as $a_t \sim \pi(\cdot|s_t)$, with the symbol $\sim$ indicating that a_t is drawn from this distribution. Following the execution of this action, the agent receives a reward signal $r_{t+1} = \mathcal{R}(s_t, a_t)$, while the environment transitions to a new state $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$. As mentioned, during the learning process, the agent endeavours to maximize its (discounted) cumulative reward by updating the policy π. In general, two types of value functions are employed to evaluate the efficacy of a given policy π: the state-value function $V^{\pi}(s)$ and the action-value function $Q^{\pi}(s, a)$ [95].

For a specific policy π, the state-value function $V^{\pi}(s)$ can be defined as follows

$$V^{\pi}(s) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s] \tag{4.1}$$

where $\mathbb{E}_{\pi}[\cdot]$ is the expectation operator calculated when applying the policy π, and s_0 is the initial state. One can define the action-value function $Q^{\pi}(s, a)$ as follows

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s, a_0 = a] \tag{4.2}$$

$Q^{\pi}(s, a)$ is the expected (naturally discounted) cumulative reward after taking the action a in state s. Also, the agent follows the given policy π afterwards. When maximized, the

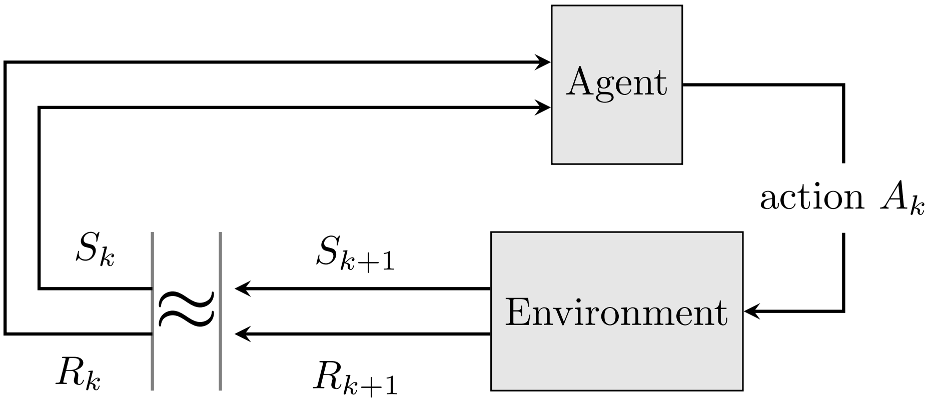


Figure 4.1: The interaction between agent and environment (Adopted from [95]).

value functions in Eqs. (4.1) and (4.2) are denoted $V^*(s)$ and $Q^*(s, a)$, which obtained by following the optimal policy π^*.

Value functions and policies can be conveniently represented in tabular form where the environment is simple. However, when dealing with state and/or action spaces that are extensive or continuous, the practicality of the tabular representation diminishes significantly [96, 95]. To tackle this challenge, Deep Reinforcement Learning (DRL) methods have emerged as a solution by merging the RL with Deep Learning [101]. More precisely, these methods enable RL-based solutions to address problems characterized by state and action spaces that are high-dimensional or continuous. This is achieved by employing Neural Networks (NNs) to approximate control policies and iteratively enhance their expected return through environmental interactions. Moreover, DRL-based approaches are gaining prominence in spacecraft control problems due to their resilience in the face of system uncertainties, adaptability to novel scenarios, and their ability to operate efficiently within constrained computational and memory resources, making them suitable for on-board execution (e.g., refer to [102]). Such suitability for attitude control made us adopt the RL paradigm in this research.

4.3.2 | Types of RL methods (Value function and policy gradient)

Reinforcement Learning (RL) approaches can be broadly classified into two primary categories: value function-based and policy gradient methods. Value function methods are primarily concerned with estimating value functions accurately. Notable examples within this category include Temporal Difference (TD) and Monte Carlo-based techniques, such as Q-learning and SARSA. Additionally, Deep-Q Learning and Deep-Q

Network (DQN) belong to this class, where value functions are approximated using neural networks [101]. It's worth noting that RL value function methods are generally suited for discrete action spaces since they require consideration of all possible actions [95].

However, numerous applications involve continuous control spaces, particularly spacecraft control problems (e.g., as discussed in [103, 104]). Hence, policy gradient learning methods become essential [96, 95]. These methods involve learning a parameterized policy$\pi_\theta(\cdot) := \pi(a|s, \theta)$, where $\theta \in \mathbb{R}^d$ represents the policy parameter vector. This policy assigns probabilities to actions, indicating the likelihood of taking action a in state s while θ represents the policy parameter vector. Policy gradient approaches make action selections without relying on a value function, but to optimize the policy parameter θ, value functions can be employed.

In broad terms, the core concept behind policy gradient methods revolves around running a series of episodes iteratively and collecting the environment' samples when a specific policy π_{θ_k} is applied (note that k denotes the episode index). Subsequently, using the gradient ascent, the policy parameters are updated as follows:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta V^{\pi_{\theta_k}} \tag{4.3}$$

Where α, θ_k denote the learning rate and policy parameters at iteration k, respectively. $V^{\pi_{\theta_k}}$ represents the state-value function corresponding to the application of the policy π_{θ_k}. This iterative process aims to improve the policy's performance over time.

In the realm of literature, various policy gradient methods exist, each offering a distinct approach for the gradient term computation as mentioned in Eq. (4.3). Notable among these methods are:

- The REINFORCE algorithm proposed by Williams [105],
- Actor-Critic methods, including the Advantage Actor-Critic (A2C) algorithm introduced by Mnih *et al.* [106], DDPG algorithm by [93], and TD3 by [94],
- The Proximal Policy Optimization (PPO) algorithm presented by [107].

In all these policy gradient methods, neural networks are widespread for formulating value functions and policies. For example, in actor–critic approaches, a neural network (actor) is employed to define parameterized policies, determining actions to be taken in coherence with the observed states of the environment. Simultaneously, a critic network computes the value function by considering both the state and the rewards accrued from interactions with the environment [95]. This integration of neural networks is pivotal in enabling RL algorithms to learn and adapt in complex and high-dimensional spaces effectively.

4.4 | Guided shaping based on reference control

While the literature extensively discusses the use of AI in attitude control, as elaborated in Chapter 2, this chapter exclusively focuses on the application of reinforcement learning (RL) in this domain. Despite the various instances where RL has been employed in attitude control [108, 71, 109, 110, 111], this research places greater emphasis on the integration of classical model-based methods with RL. The objective is to enhance RL's learning and fault-tolerant control performance. Therefore, the following section provides a concise overview of this topic.

Several investigations have centred on combining proportional-integral-derivative (PID) control with RL in process control, including studies by [112, 113, 114, 115]. In the context of attitude control, and to expedite the learning process in TD3, Zhang *et al.* proposed a strategy guided by PID [112], illustrated in Figure 4.2. Their approach hinges on selecting the action with the highest value between the policy-generated output and the PID controller's output under specific circumstances. However, this approach is hampered by the need for a more comprehensive exploration strategy encompassing the PID control vector.

To address the challenges above, this study introduces a reward-shaping technique. Our method capitalizes on the resemblance between PID outputs and those of the RL agent to expand the exploration area around the reference control. Our strategy endeavours to expedite actor-critic (AC) agent training through expert-guided exploration (EGE), where "expert" signifies any model-based reference controller. We quantify the similarity between the RL agent's output vector and the reference controller's output vector and add this measure to the existing reward. This approach is designed to steer policy space exploration towards more rewarding regions [116].

4.4.1 | Preliminaries

As previously mentioned, Policy Gradient approaches aim to optimize policies in pursuit of maximum rewards [117]. Among these approaches, the Deep Deterministic Policy Gradient (DDPG) technique stands out as the first to address Reinforcement Learning (RL) with continuous action spaces [93]. DDPG is founded on the fusion of Q-learning and policy gradient concepts. Inspired by the work of Sutton *et al.* [118],

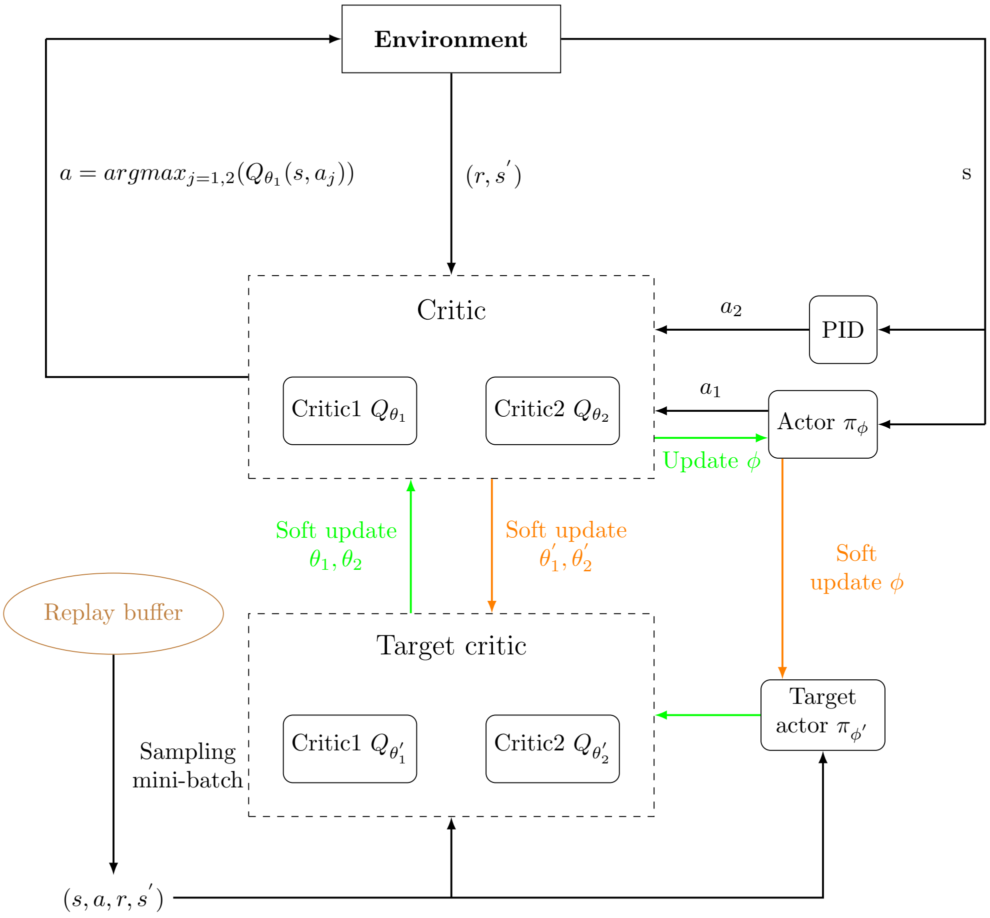


Figure 4.2: PID-Guide TD3 structure (Adopted and modified from [112]).

Silver *et al.* developed the deterministic policy gradient theorem, which is expressed as follows [117]:

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_{s\sim\rho^\mu}[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}] \tag{4.4}$$

Formulating a practical reward function in the context of RL is a daunting task, and if not approached carefully, it can lead to suboptimal solutions. For instance, Randløv and Alstrøm [119] described a scenario in which they employed an MDP to simulate riding a bicycle to reach a target location. They observed that by offering a positive reward whenever the distance to the target decreased, the agent learned a suboptimal policy of riding in small circles near the starting point. This example underscores the challenges associated with straightforwardly learning an optimal policy.

Moreover, RL has its roots in psychological, behavioural, and biological foundations, emphasizing the reinforcement of actions in proportion to the expected return. The concept of shaping, which involves reinforcing successive approximations of a desired behaviour, has been integral to RL research. It traces its origins back to the 1930s when behaviorist Skinner investigated its effects [120]. Since then, numerous researchers have explored the application of shaping in animal training to elicit specific behaviors [121, 122].

From a machine learning perspective, RL approaches must be capable of scaling to more complex and larger-scale problems while also accelerating the learning process. These requirements can be addressed through the use of shaping techniques, aiming to discover more optimal policies [119, 123, 124, 125]. However, implementing shaping strategies is more intricate than it may initially seem. [119] suggested that shaping in RL can be achieved by (***i***) establishing rewards that are related to the approximations between the agent's actions and intended behaviours, and (***ii***) creating a multi-stage architecture in which each stage is trained independently. Notably, this approach aligns with the recommendations of Shirobokov *et al.* [126]. In a similar vein, Shirobokov *et al.* hailed RL as a powerful model-free method capable of competing with traditional model-based controllers. They asserted that traditional controllers could serve as training aids for neural-like adaptive control systems [126].

Furthermore, Reinforcement Learning (RL) algorithms that heavily rely on the design of the reward function often encounter challenges related to convergence, including issues with learning speed and the risk of getting stuck in local minima. Ng *et al.* demonstrated that modifying the fundamental reward structure of the MDP can be beneficial for training agents and mitigating undesired behaviors [123].

As a result, building upon the previously mentioned approach of shaping, we will introduce in the subsequent section a straightforward method that encompasses the following elements:

- Guiding the RL agent to acquire a specific desired behaviour,
- Achieving this objective through a multi-level learning process, where the initial stage utilizes gradient ascent with reward shaping to reach an optimal policy, while subsequent stages proceed without reward shaping.

4.4.2 | Expert guided exploration (EGE)

Let's commence this section with a comprehensive review of policy gradient (PG) methods, as detailed in Section 4.3.2. PG algorithms encompass both policy evaluation and improvement, following the principles outlined in [95]. In the Actor-Critic (AC) framework, the critic's role is to estimate the action-value function $Q^{\mu}(s,a)$. Meanwhile, the actor is responsible for updating the policy, denoted as $\mu_{\theta}(s)$, based on the estimated $Q^{\mu}(s,a)$. In scenarios where the conventional approach of performing greedy maximization in continuous action spaces becomes computationally infeasible, a more favourable alternative is to adjust the policy in accordance with the gradient of Q. Specifically, the gradient, denoted as $\nabla_{\theta} Q^{\mu^k}(s,\mu_{\theta}(s))$, is utilized to modify the parameters represented by θ_{k+1}. Additionally, this method computes a mean or expectation (refer to Eq. (4.5)) concerning the state distribution $\rho^{\mu}(s)$ to rectify state dispersion, which can otherwise lead to a non-unique policy improvement direction [117]:

$$\theta_{k+1} = \theta_k + \alpha \mathbb{E}_{s\sim\rho^{\mu^k}}[\nabla_{\theta} Q^{\mu^k}(s,\mu_{\theta}(s))] \tag{4.5}$$

Applying the chain rule reveals that the gradient ascent is proportional to the product of two factors: the policy gradient concerning its parameters θ and the gradient of the critic's output concerning actions:

$$\theta_{k+1} = \theta_k + \alpha \mathbb{E}_{s\sim\rho^{\mu^k}}[\nabla_{\theta}\mu_{\theta}(s)\nabla_a Q^{\mu^k}(s,a)|_{a=\mu_{\theta}(s)}] \tag{4.6}$$

The expression within the expectation in Eqs. (4.5) and (4.6) encapsulates the intuitive principle that actions are reinforced based on the rewards obtained after their execution. In essence, the policy is updated to enhance the probability of performing action a in state s with this probability naturally influenced by $\nabla_a Q^{\mu^k}(s,a)|_{a=\mu_{\theta}(s)}$.

A standard training process for all policy gradient algorithms typically follows a pattern of starting with a higher exploration level and gradually decreasing it over time. Our strategy suggests a different approach by guiding this exploration using reward shaping, particularly at the beginning. Considering the definition of $Q^{\mu}(s,a)$ given by Eq. (4.2), we propose modifying the initial reward $r(s,a)$ in the MDP during the shaping phase. This modification aims to encourage exploration in regions where a potentially higher return is expected during the gradient ascent. We introduce an action-dependent function, denoted as $\Phi(s,a)$, into the reward function to achieve this. It accounts for the fact that the gradient term $\nabla_a Q^{\mu^k}(s,a)|_{a=\mu_{\theta}(s)}$ influences the actions taken at each step. It's important to note that $\Phi(s,a)$ must adhere to continuity requirements, and the gradient $\nabla_a \Phi(s,a)$ should guide the agent to follow an expert control method, represented by at least one policy within the infinite policy space. Notably, our approach relies on a metric that quantifies the similarity between the agent's actions and the reference control, which could be, for example, a PID control method.

For episodic tasks, Eq. (4.2) can be expressed as:

$$Q^{\mu}(s,a) = [\sum_{t=0}^{T} \gamma^t r(s_{t+1}, a_{t+1}) | s_0 = s, a_0 = a; \mu] \tag{4.7}$$

Where T denotes the duration of an episode. The reward function during the shaping phase is given by:

$$r_{shaping}(s,a) = r(s,a) + \Phi(s,a) \tag{4.8}$$

This representation allows us to rewrite Eq. (4.7) as:

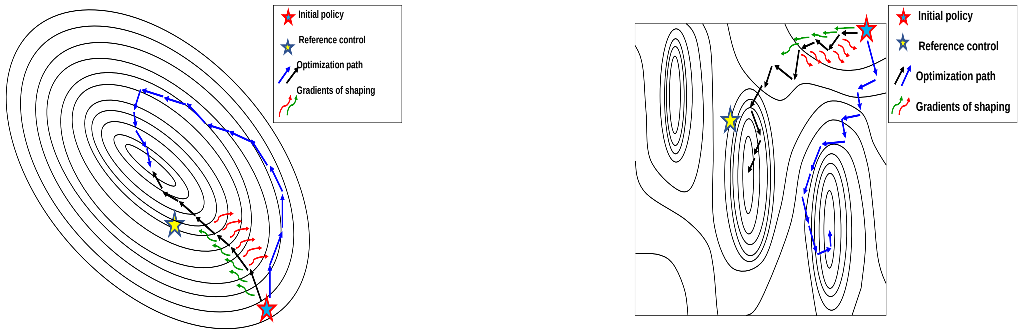
$$Q^{\mu}(s,a) = [\sum_{t=0}^{T} \gamma^t r(s_{t+1}, a_{t+1}) + \sum_{t=0}^{T} \gamma^t \Phi(s_{t+1}, a_{t+1}) | s_0 = s, a_0 = a; \mu] \tag{4.9}$$

In summary, our approach leverages reward shaping, driven by Eq. (4.8), to guide the agent exploration, especially during the initial stages of training. This approach incorporates an action-dependent function and aims to align the agent's actions with a reference control method while adhering to continuity requirements.

The critic estimates the action-value function $Q^{\mu}(s,a)$. Consequently, if we are able to compute the gradients $\nabla_{\theta}\mu_{\theta}(s)$ and $\nabla_a Q^{\mu^k}(s,a)|_{a=\mu_{\theta}(s)}$, we can derive the policy gradient and proceed with an update step, provided we can generate an experience-based trajectory dataset $(s_0, a_0, s_1, a_1, \cdots, s_{T-1}, a_{T-1}, s_T)$ through agent-environment interactions. Moreover, this expert guided exploration (EGE) technique guides the agent's behaviour in the initial episodes to encourage the selection of actions that closely align with those computed by the reference control. The detailed impact and mechanism of EGE will be elaborated upon in the following equation:

$$\begin{aligned}
\nabla_a Q^{\mu}_{shaping}(s,a)|_{a=\mu_{\theta}(s)} &= \nabla_a [\sum_{t=0}^{T} \gamma^t r(s_{t+1}, a_{t+1}) + \sum_{t=0}^{T} \gamma^t \Phi(s_{t+1}, a_{t+1}) | s_0 = s, a_0 = a; \mu] \\
&= \nabla_a \Bigg([\sum_{t=0}^{T} \gamma^t r(s_{t+1}, a_{t+1}) | s_0 = s, a_0 = a; \mu] \\
&+ [\sum_{t=0}^{T} \gamma^t \Phi(s_{t+1}, a_{t+1}) | s_0 = s, a_0 = a; \mu] \Bigg) \\
&= \nabla_a Q^{\mu}(s,a)|_{a=\mu_{\theta}(s)} \\
&+ \nabla_a [\sum_{t=0}^{T} \gamma^t \Phi(s_{t+1}, a_{t+1}) | s_0 = s, a_0 = a; \mu]
\end{aligned} \tag{4.10}$$

Using this approach, there's no need to explicitly compute the second term of Eq. (4.10) since it is implicitly determined by the critic. Instead, our intention is to showcase its impact on guiding the agent's exploration toward a region resembling an expert policy. This natural alignment accelerates the learning process. You can observe the role of EGE in guiding the agent in Figure 4.3(a). Assuming that the reference control inherently outperforms the initial policy, the gradients represented by the red and green arrows in Figure 4.3(a) guide the agent towards a more efficient optimization path. However, it's important to note that the process of shaping should be suspended during the early training phase to allow for further policy refinement using the fundamental PG algorithm. Extending the shaping of rewards over prolonged periods strives to make the resulting policy as similar as possible to the reference control. While this might seem advantageous, it hinders the use of RL's inherent benefits. Therefore, the hybridization in our approach is advantageous only within controlled shaping intervals. Beyond these intervals, two potential drawbacks may arise:



(a) Convex problem.

(b) Non-convex problem.

Figure 4.3: Effect of shaping gradient on policy optimization.

- the risk of finding suboptimal policies with short shaping intervals,
- in the case of longer shaping, the PG process aims for complete imitation of the reference control, aligning the trained policy as closely as possible with model-based control, thus negating the advantages of model-free RL.

In Eq. (4.4), $\nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}$ serves as a weighting factor that encourages a higher return. Similarly, the objective of $\nabla_a[\sum_{t=0}^{T}\gamma^t\Phi(s_{t+1},a_{t+1})|s_0 = s, a_0 = a; \mu]$ in Eq. (4.10) is to promote movement towards the reference policy, preventing overly lengthy exploration routes. Typically, the objective function is non-convex and encompasses multiple optima. Figure 4.3(b) illustrates that, as a result, the reference policy generally steers the agent towards an optimal solution equivalent to or superior to that achieved by the basic algorithm. Figure 4.4 provides an overview of the suggested RL framework, incorporating EGE.

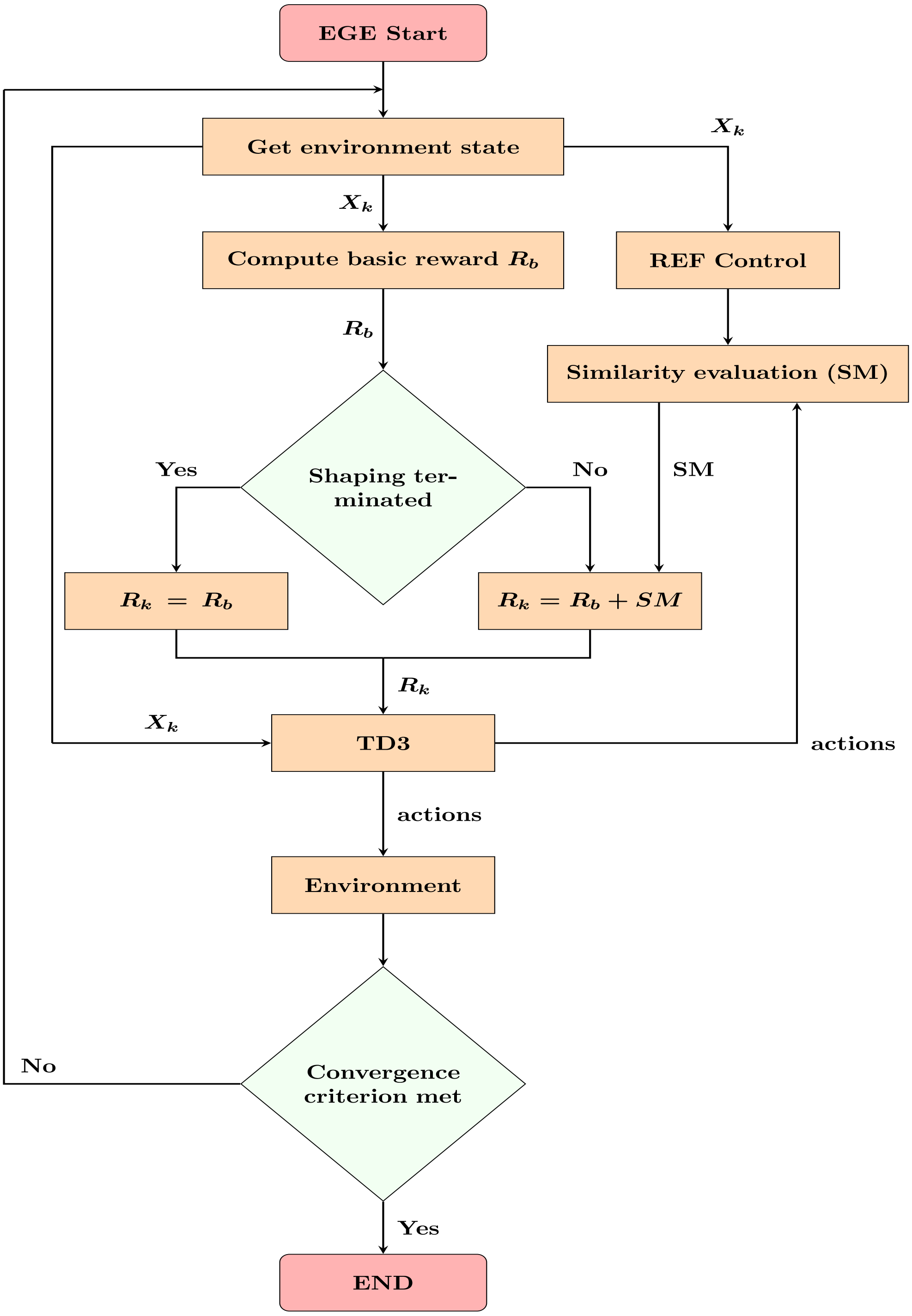


Figure 4.4: RL/EGE framework.

4.4.3 | Similarity metric (SM)

As previously demonstrated, early reward shaping relies on assessing the likeness between the output produced by the training agent and the reference control signal. Within the realm of 3D Euclidean space, the metric for similarity evaluates the proximity between two control vectors. Consequently, we can gauge this proximity through various functions. In this research, we employed two distinct types of similarity metrics to illustrate the flexibility in selecting an appropriate vector similarity evaluation method. However, it is important to note that the performance and computational complexity of EGE may vary depending on the specific similarity metric chosen, although this aspect falls outside the purview of our current study.

Henceforth, we will denote the outcomes generated by the RL agent and the reference control as two distinct 3D vectors, labelled V_1 and V_2. These vectors possess coordinates $[x_1, y_1, z_1]$ and $[x_2, y_2, z_2]$, respectively, which represent the projections of the attitude control efforts onto the axes of the satellite's body frame.

4.4.3.1 | Euclidean distance

The computation of the Euclidean distance (ED) between two vectors V_1 and V_2 can be expressed as follows:

$$ED = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{4.11}$$

Given that ED is inversely related to vector similarity, the Euclidean-based similarity metric (SM_E) is derived by taking the reciprocal of ED:

$$SM_E = \frac{Gain}{0.01 + ED} \tag{4.12}$$

Here, *Gain* is appropriately chosen to enhance the agent's sensitivity. It's worth noting that (refer to Eq. (4.12)) we add a small value, such as 0.01, to ED in the denominator to avoid division by zero.

4.4.3.2 | Gaussian function

The utilization of Gaussian functions for shaping the reward in the MDP has been discussed by [127]. In our work, the Gaussian function focuses on two key attributes: the angle between vectors and the 2-norm ratio. The angle, denoted as θ, between two vectors V_1 and V_2 can be expressed as:

$$\theta = \cos^{-1}\left(\frac{V_1 \cdot V_2}{\|V_1\| \|V_2\|}\right) \tag{4.13}$$

The use of the angle in reward shaping is also explored in a different context by [124]. In their study, Dayal *et al.* employed the angle between the agent's orientation and the target's coordinates [124]. Here, we consider the angle between two vectors in Euclidean space.

To calculate the 2-norm ratio, denoted as R_n, certain precautions are taken to avoid division by zero and limit the ratio within a manageable range. Initially, we establish upper and lower bounds for the ratio, represented as $\left[R_{min} = \frac{1}{R_{max}}, R_{max}\right]$ (e.g., $\left[\frac{1}{1000}, 1000\right]$). The calculation of R_n is as follows:

$$R_n = \begin{cases} R_{max}; \quad if \|V_1\| = 0 \quad or \quad \|V_2\| = 0 \\ clip(\frac{\|V_1\|}{\|V_2\|}, R_{min}, R_{max}); \quad otherwise \end{cases} \tag{4.14}$$

In Eq. (4.14), R_n is clipped to the predefined limits whenever it exceeds them, ensuring efficient computation and maintaining a safe margin from software tolerance thresholds. The maximum vector proximity occurs when θ approaches 0 and R_n approaches 1. Consequently, the Gaussian function, denoted as $g(x)$, such as:

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right) \tag{4.15}$$

is fully defined by selecting appropriate mean and variance parameters, typically $\mu_\theta = 0$ and $\mu_{R_n} = 1$. The Gaussian-based similarity metric, denoted as SM_G, is computed as follows:

$$SM_G = Gain \;\; g(\theta)g(R_n) \tag{4.16}$$

4.4.4 | Computational complexity considerations

Machine learning methods are known for their computational demands, and it is advisable to assess the computational load of each method and its impact on the system. In the case of EGE (Expert Guided Exploration), two computational aspects are introduced: (*i*) the inclusion of additional inputs into the agent's neural networks (NN), specifically, the torques from the reference controller and (*ii*) the integration of the software block responsible for computing the similarity metric. The latter, involving the similarity metric computation, is not computationally intensive and does not require special attention.

However, the expansion of inputs for a neural network necessitates careful consideration. Let's denote the number of parameters in the neural network as N_θ, the additional inputs as E_n, and the size of the first hidden layer as S_{h1}. The PG algorithm optimizes N_θ values during training. With the incorporation of EGE, there's a need to optimize an additional set of $E_n S_{h1}$ values, resulting in a total number of weights and biases to be optimized equally to $N_\theta + E_n S_{h1}$.

In the context of attitude control, where the decision-making vector involves three dimensions (representing 3-axis stabilization ACS), computational complexity has a relatively minimal impact on the system. However, conducting a thorough investigation is essential to ensure computational efficiency when dealing with larger dimensions of action spaces. As a summary, Table 4.1 illustrates a general comparison between EGE and the basic PG frameworks.

Table 4.1: EGE-enhanced and basic PG comparison.

Criteria	EGE-enhanced PG	Basic PG
Complexity	Complex if large control vector	Standard
Convergence	Fast if complex environment	Better if simple environment
Optimization	Better in minima avoidance	Standard

4.5 | Results and discussion

To assess the efficacy of the proposed approach, we employed an attitude control MDP in conjunction with an Actor-Critic (AC) agent. This was done to showcase the performance of EGE concerning both training speed and accumulated rewards. Furthermore, we conducted simulations involving FTC and parameter uncertainty to compare our method's performance with that of the standard TD3 agent.

The simulation experiments were carried out in the Matlab® and Simulink® environments. Within this setup, the MDP model, similarity function, and reference controller were constructed using Simulink, while hyperparameters were defined using Matlab code. The RL agent utilized in these experiments is part of the Reinforcement Learning Toolbox (Version 2.0). The default *ode4* solver was applied for all simulations.

4.5.1 | MDP setting

This section provides a detailed description of the MDP configuration for attitude control. This description aims to facilitate the replication of our results. The MDP is designed to simulate three-axis stabilization and employs quaternion representation to model the dynamics and kinematics of a typical spacecraft's attitude.

We introduce faults into the environment for policy validation purposes, specifically simulating gyroscopes and reaction wheel faults. This allows us to assess the policy's fault-tolerant performance. At the beginning of each episode, a random reference trajectory, denoted as $\{\omega_1, q_1, \omega_2, q_2, \cdots, \omega_T, q_T\}$, is generated. At each timestep i, this trajectory includes information about the 3-axis angular velocity (ω_i) and the attitude quaternion (q_i). The Markovian state, denoted as $\{\omega, q, \delta\omega, \delta q, Tc_{ref}\}$, encompasses various components, including angular velocity, attitude quaternion, angular velocity error, attitude quaternion error, and reference control torques. These components collectively serve as inputs to the TD3 agent.

For further clarity, Table 4.2 provides a comprehensive breakdown of the neural network architecture employed by the TD3 agent, while Table 4.3 presents a summary of the hyperparameter settings used in our experiments.

Table 4.2: NN architecture (Policy and value function).

Layer	Actor network		Critic network	
	# units	Activation	# units	Activation
input	17		20	
hidden 1	64	ReLu	64	ReLu
hidden 2	32	ReLu	32	ReLu
hidden 3			16	linear
output	3	tanh	1	linear

Table 4.3: Hyperparameter settings.

Hyperparameter	Value
Max episodes	2000
Max steps per episode	140
Sample time (sec)	0.25
Replay buffer size	2×10^6
Mini batch size	512
Actor learning rate	3×10^{-3}
Critic learning rate	3×10^{-4}
Discount factor	0.995
Target update	10
Target smooth factor	5×10^{-3}
Exploration model	OrnsteinUhlenbeck
Max Torque value (Nms)	0.012

We formulate a composite reward signal combining discrete and continuous components to assess the agent's performance. The continuous components are introduced to expedite convergence when the agent is in proximity to the target state, while the discrete components serve to deter the agent from entering undesirable states. Eq. (4.17) provides a detailed expression of the core reward function.

$$\begin{cases} r_1 = - \|q_v\| \\ r_2 = \begin{cases} \dfrac{1}{0.01 + \|q_v\|}; & if\|q_v\|_\infty \leq 0.09 \\ 0; & otherwise \end{cases} \\ r_3 = \dfrac{1}{0.01 + \|Tc\|} \\ r_4 = \begin{cases} -150; & if\|q_v\|_\infty > 0.18 \\ 0; & otherwise \end{cases} \\ r_b = \sum_{i=1}^{4} r_i \end{cases} \tag{4.17}$$

In this equation, the attitude quaternion error vector component, denoted as q_v, is a crucial factor to consider for improving pointing performance. As evident from Eq. (4.17), the term r_1 penalizes the agent (assigning a negative value) in proportion to the norm of q_v. Conversely, when the condition ($if\|q_v\|_\infty \leq 0.09$) is met, r_2 is a positive reward that encourages the agent to approach the target state where q_v is ideally zero. Therefore, this condition promotes actions that further reduce q_v. r_3 contributes to a reduction in energy consumption. It's worth noting that if the condition of the discrete term r_4 is met, the episode will terminate. Additionally, a small bias of 0.01 is added to the denominator of all fractional values to prevent division by zero.

4.5.2 | Similarity metric setting

In accordance with the definition of the similarity metric (SM), our approach assesses the proximity between the agent's control vectors and the reference control vectors. Within this study, we present two SM functions: one based on Euclidean distance and another based on Gaussian principles. The specific parameter configurations for these functions are provided in Table 4.4.

It's essential to note that the values listed in Table 4.4 are not arbitrarily chosen. For instance, the number of episodes during which we activate reward shaping is directly proportional to the training horizon, while adhering to the constraints outlined in Section 4.4.2. The Gaussian metric variances depend on physical characteristics and the

Table 4.4: Similarity metric configuration.

Euclidean			Gaussian (Gain = 20)			
Id	Gain	Shaping	Id	σ_θ	σ_{R_n}	Shaping
Euclid-1	0.45	500	Gauss-1	0.5	1	800
Euclid-2	0.45	800	Gauss-2	0.2	1	800
Euclid-3	0.45	1500	Gauss-3	1	1	800

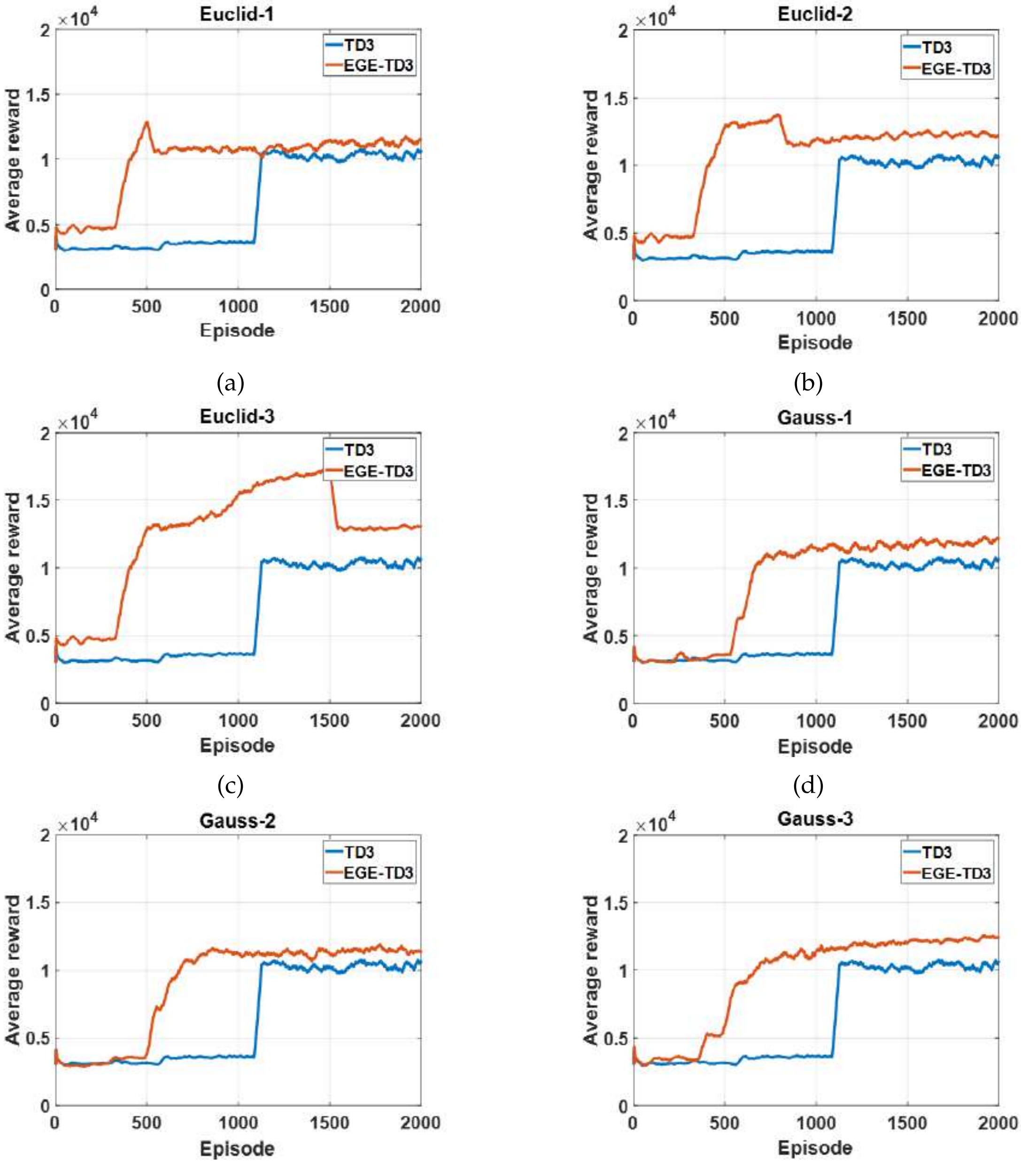
expert's evaluation of σ_θ and σ_{R_n}. Additionally, the gains are carefully set to ensure the agent's sensitivity is appropriate. For example, setting a gain that results in a shaping reward equivalent to 20% of the total reward is a prudent choice.

4.5.3 | Agent's learning behavior

In this subsection, we delve into the EGE-TD3 methodology and draw comparisons with the fundamental TD3 algorithm. TD3 was chosen for this comparison because it represents an enhanced version of DDPG, and both are state-of-the-art policy gradient algorithms. Figure 4.5 provides a comparative analysis of the learning curves for EGE-TD3 (Euclidean- and Gaussian-based) and TD3. These curves vividly demonstrate that EGE exhibits superior convergence performance in all configurations. Moreover, the final cumulative rewards achieved by agents utilizing EGE surpass those of the TD3 agent.

The straightforward definition of the Euclidean distance metric also results in better performance for agents utilizing this metric compared to those employing the Gaussian function. The impact of the EGE approach on the agent's action selection becomes evident when examining the shaping interval. For instance, when EGE was applied in the initial 1500 episodes (refer to Figure 4.5(c)), the final cumulative reward significantly exceeded that of the TD3 agent. This suggests that EGE empowered the agent to make more optimal decisions, highlighting the effectiveness of our training strategy. However, when the shaping interval is reduced (as seen in Figure 4.5(a)), there is no substantial difference in the final rewards between EGE-TD3 and TD3. In this case, the shorter shaping interval limited the EGE-TD3 agent's ability to discover a more optimal path compared to TD3.

Furthermore, it's noteworthy that the average rewards in the EGE-TD3 learning curves show an early increase compared to TD3. This early increase is a result of the action selection strategy at that stage, which encourages the imitation of the PID controller. This imitation allows for the accumulation of higher cumulative rewards during



(a)

(b)

(c)

(d)

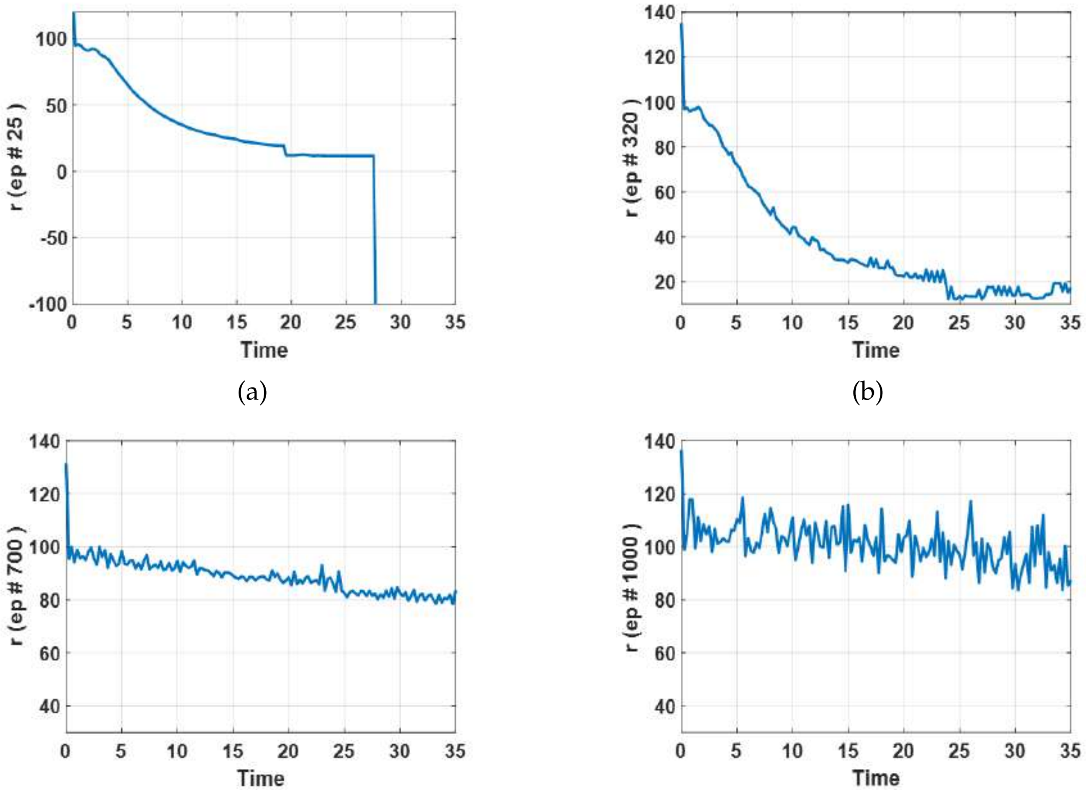
(e)

(f)

Figure 4.5: Comparison of average reward during training.

the same training episodes, a benefit not readily available to TD3 agents.

The EGE effect is elaborated further through Figure 4.6. In the initial phase, the term r_4 within Eq. (4.17) penalizes the agent leading to a premature episode termination (refer to Figure 4.6(a)). Subsequently, the agent generates actions that partially resemble those of the reference control (as depicted in Figure 4.6(b)). This resemblance corresponds to peaks in the reward value. Figure 4.6(c) illustrates that the reward exhibits numerous peaks of similarity, and Figure 4.6(d) demonstrates that these peaks in the later stages are significantly higher than those in the earlier phases.



(a)

(b)

(c)

(d)

Figure 4.6: Reward evolution during the shaping phase.

These observations suggest that as the training progresses, the agent discovers a trajectory along which it imitates the reference control more effectively. When combined with the policy gradient technique, this imitation can expedite the learning process, enabling the acquisition of more efficient policies.

4.5.4 | FTC performance

Furthermore, we have developed a testing environment to evaluate the performance of TD3 and EGE-TD3 policies. These tests encompass both fault-free and fault-injected scenarios. The primary goal of attitude control is twofold: tracking and regulation. Tracking involves maintaining a desired attitude based on an initial reference quaternion $q_{ref0} = [1, 0, 0, 0]$ and angular velocity $\omega_{ref} = [0, 0, 10^{-2}] \quad rad/s$. Regulation, on the other hand, aims to stabilize the required attitude around $q_{ref} = [1, 0, 0, 0]$ while maintaining zero angular velocity.

All initial conditions are defined as $q_0 = [0.7, 0, 0.7, 0]$ and $\omega_0 = [0, 10^{-2}, 0] \quad rad/s$. For FTC, we consider two types of faults: $f_{gyro} = [0, 0.02, 0] \quad rad/s$ representing gyroscope faults and $f_{actuator} = [0.005, 0, 0] \quad Nms$ representing actuator faults. The simulation lasts for 1200 seconds, with fault injection occurring 600 seconds after the simulation initiation. Notably, the actuator fault is terminated at $t = 1000$ seconds.

Table 4.5 provides a comprehensive breakdown of the criteria for comparing trained policies in nominal and faulty scenarios. Additionally, we conduct an additional robustness test for EGE, examining its performance in the presence of system parameter uncertainty (as detailed in Section 4.5.5).

Table 4.5: Comparison criteria of trained policies.

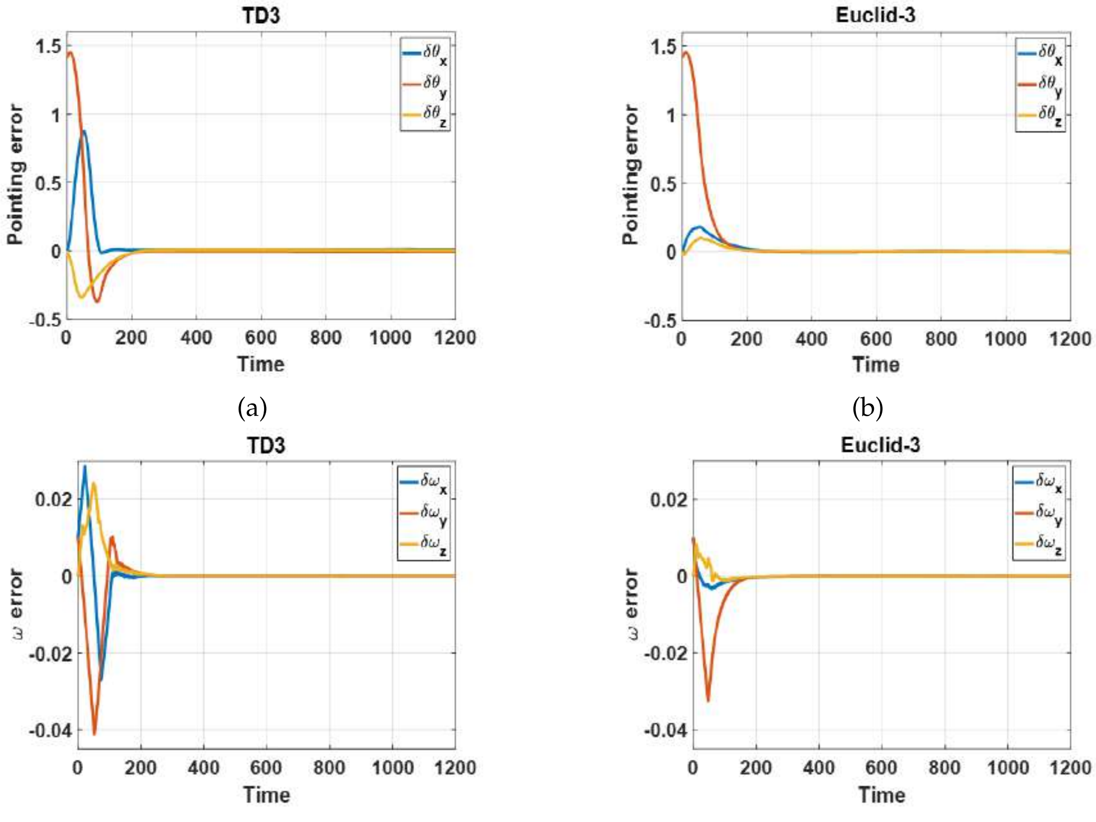
Criteria	Equation	Application
Pointing error	$\theta_{ref} - \theta$	All cases except actuator fault
Pointing error norm	$\sqrt{\delta\theta_x^2 + \delta\theta_y^2 + \delta\theta_z^2}$	Faults
Pointing error norm	$[Tc_x, Tc_y, Tc_z]$	Actuator fault
Torque norm	$\sqrt{Tc_x^2 + Tc_y^2 + Tc_z^2}$	Actuator fault
Angular velocity error	$\omega_{ref} - \omega$	All cases except actuator fault

4.5.4.1 | Fault-free tracking and regulation results

Figures 4.7 and 4.8 present the results regarding pointing and angular velocity errors. When comparing the performance of TD3 and Euclid-3-EGE agents, it's evident that both agents successfully stabilized the attitude with respect to the angle and velocity references. However, the EGE-TD3 agent displayed superior regulation and tracking performance, particularly along the X and Z axes. Specifically, the TD3 agent achieved its minimum pointing error ($\delta\theta$) at approximately 350 seconds, while EGE-TD3 reached this minimum at $t = 250$ seconds. Notably, our agent's performance in stabilizing the angular rate error ($\delta\omega$) is notably superior.

4.5.4.2 | Tracking and regulation results with gyro fault

To evaluate the performance of FTC in the first fault case, we introduced a gyro bias on the Y-axis for attitude tracking and regulation (refer to Figures 4.9 and 4.11). Notably, the EGE-TD3 agent effectively mitigated the influence of the Y-axis fault, minimizing



(a)

(b)

(c)

(d)

Figure 4.7: Fault-free tracking performance.

its impact on the orthogonal axes (X and Z). Furthermore, the EGE-TD3 agent notably excelled in reducing the average pointing error, as illustrated in Figures 4.10 and 4.12. Additionally, it's worth mentioning that the EGE-TD3 agent achieved post-fault stabilization in a significantly shorter time (160 seconds) compared to the TD3 agent (250 seconds). These results suggest that guided exploration enhanced the agent's resilience against sensor errors, a task in which model-based control may excel under specific circumstances [10, 31, 128]. Conversely, the TD3 agent prioritizes re-stabilizing the attitude with less emphasis on minimizing the required time and the fault's impact on other axes, which aligns with the fundamental reward function (refer to Eq. (4.17)).

4.5.4.3 | Tracking and regulation results with actuator fault

In the second fault scenario, a bias fault was introduced on the X-axis of the actuation system. Figures 4.13(a), 4.13(b), 4.14(a) and 4.14(b) illustrate the pointing error results, demonstrating the superior performance of the EGE-TD3 agent over the TD3 agent.

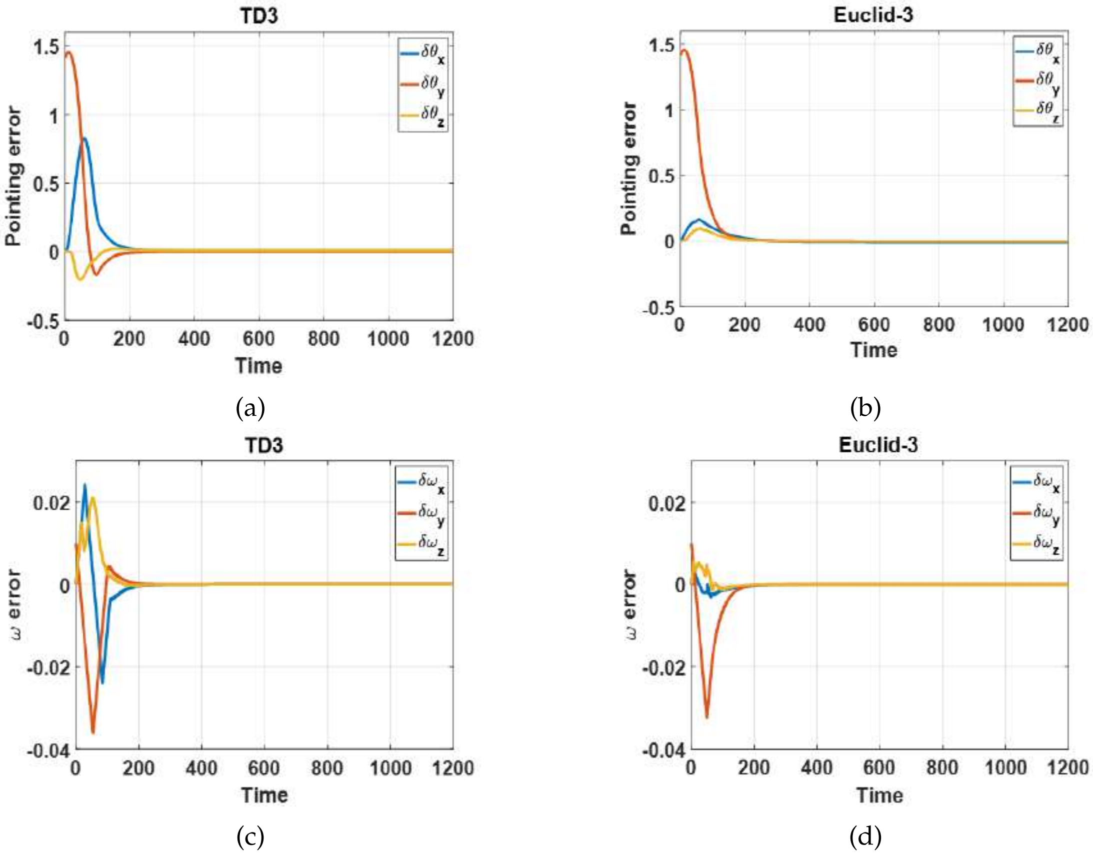
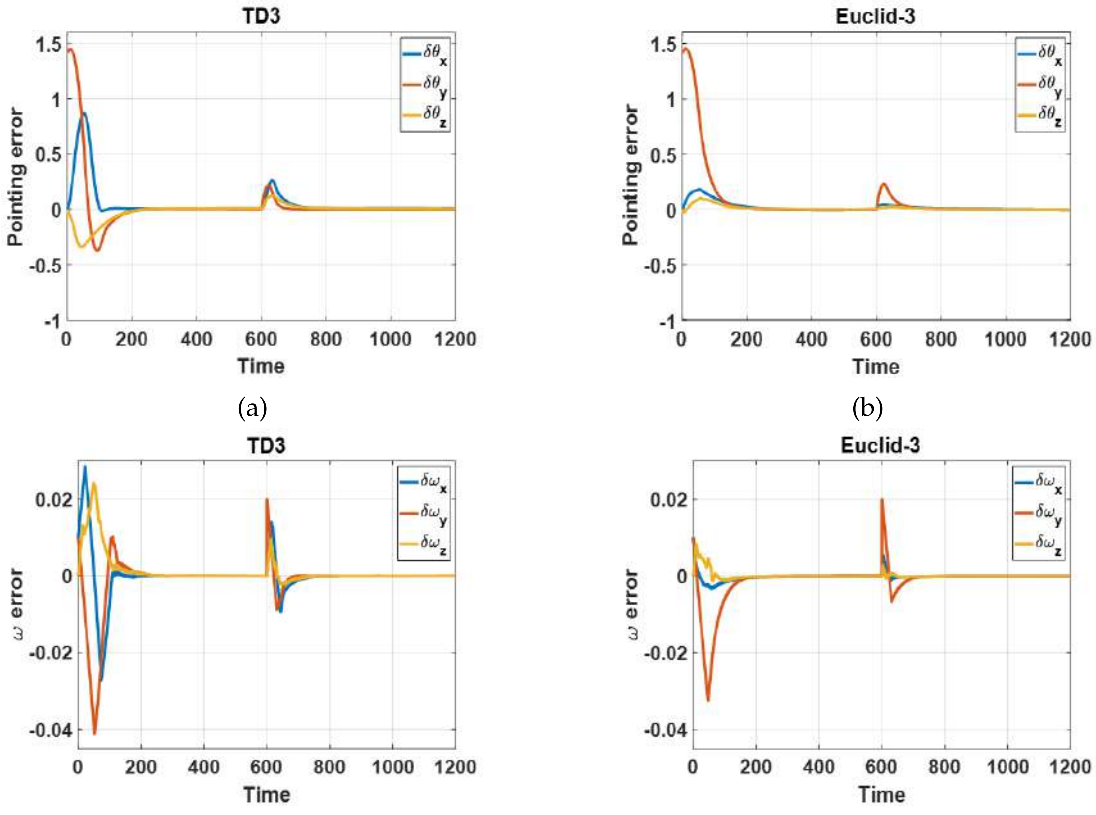


Figure 4.8: Fault-free regulation performance.

Torques were used as a comparison criterion in this case. Figures 4.13(c), 4.13(d), 4.14(c) and 4.14(d) display the applied torques by both agents. These results highlight that the impact of the fault is significantly more pronounced with the TD3 agent. While attitude stabilization remains comparable in this fault condition, the EGE-TD3 agent markedly reduces the energy consumption of the actuation system. Additionally, the EGE-TD3 agent demonstrates superior actuator fault decoupling performance in this scenario. Examining the torques applied by both agents in Figures 4.13 and 4.14, it becomes evident that EGE enables the development of a strategy for rectifying actuator failure without affecting other axes. In contrast, the TD3 agent does not effectively decouple the fault, leading to suboptimal results. Consequently, our EGE agent ensures fault-tolerant control performance with minimal risk of actuator saturation.

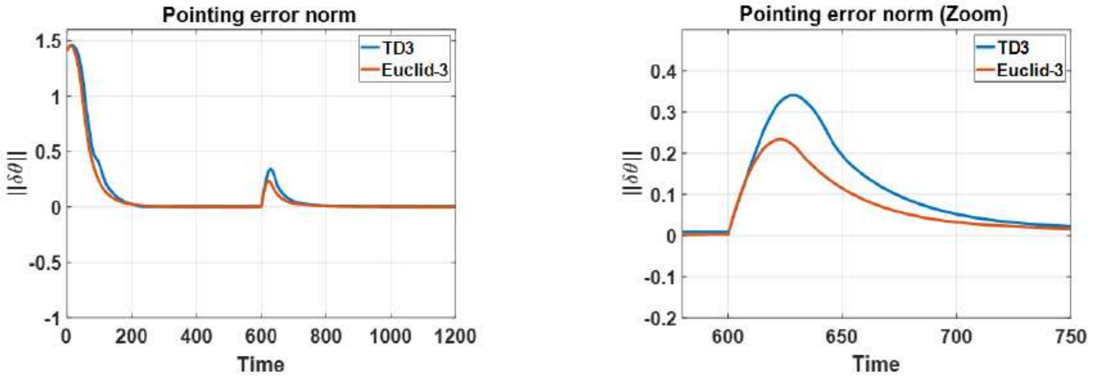
Analyzing the actuator fault results, particularly in terms of control effort, it is clear that EGE excels in addressing the optimization concept. On the same training horizon, it is evident that the standard TD3 algorithm struggles to navigate local minima optimally. These findings suggest that the fundamental TD3 agent generates suboptimal



(a) (b)

(c) (d)

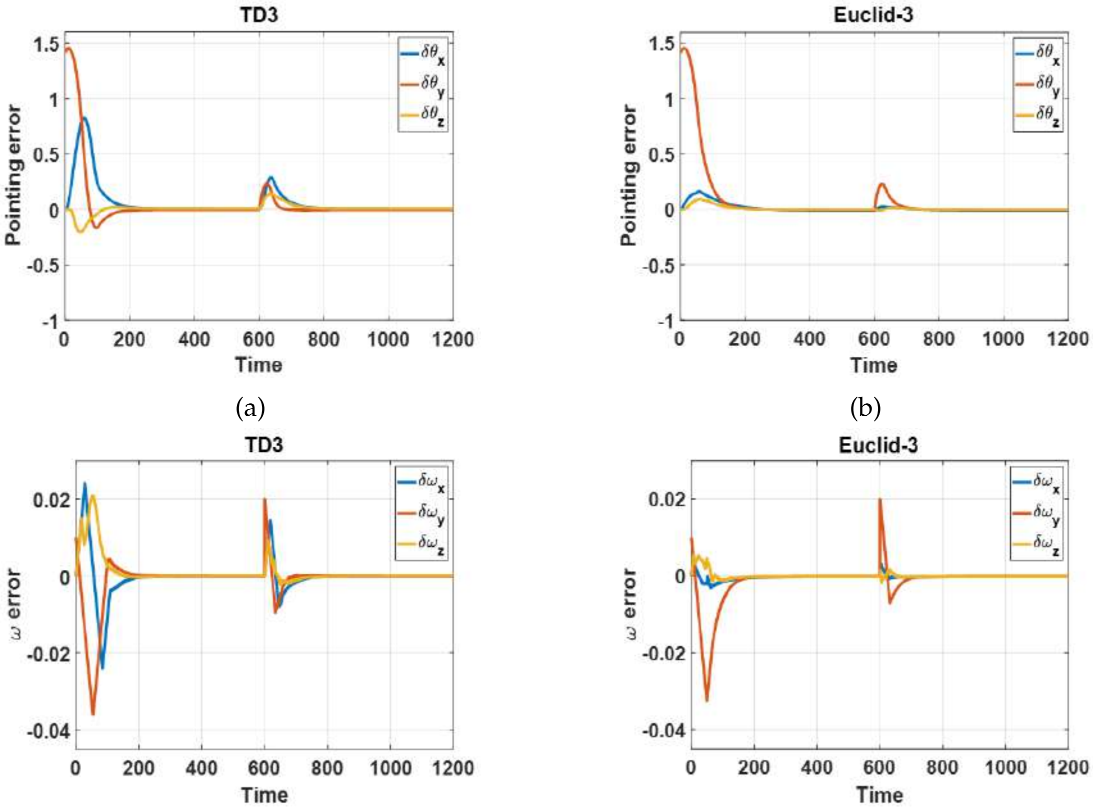
Figure 4.9: Tracking performance in gyro fault case.



(a) (b)

Figure 4.10: $||\delta\theta||$ in tracking case with gyro fault.

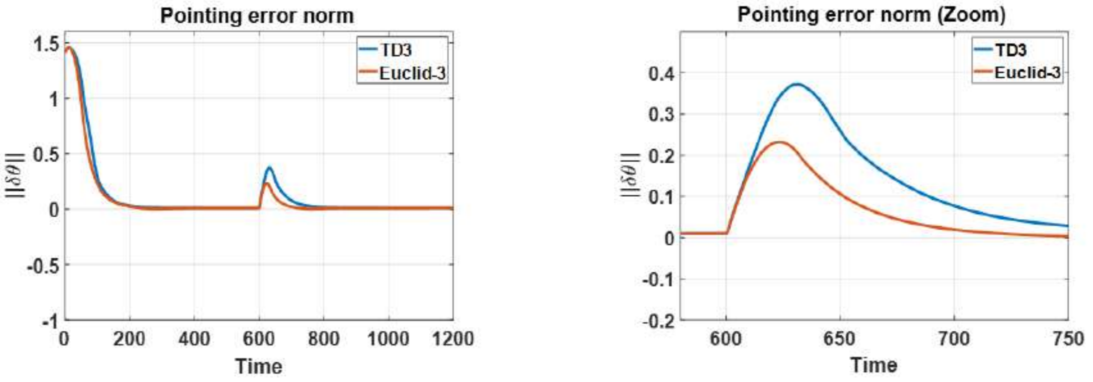
control signals regarding energy consumption. Therefore, assuming that the TD3 algorithm may reach a local minimum where only pointing performance is prioritized is reasonable. In contrast, our approach trains an agent capable of simultaneously opti-



(a) (b)

(c) (d)

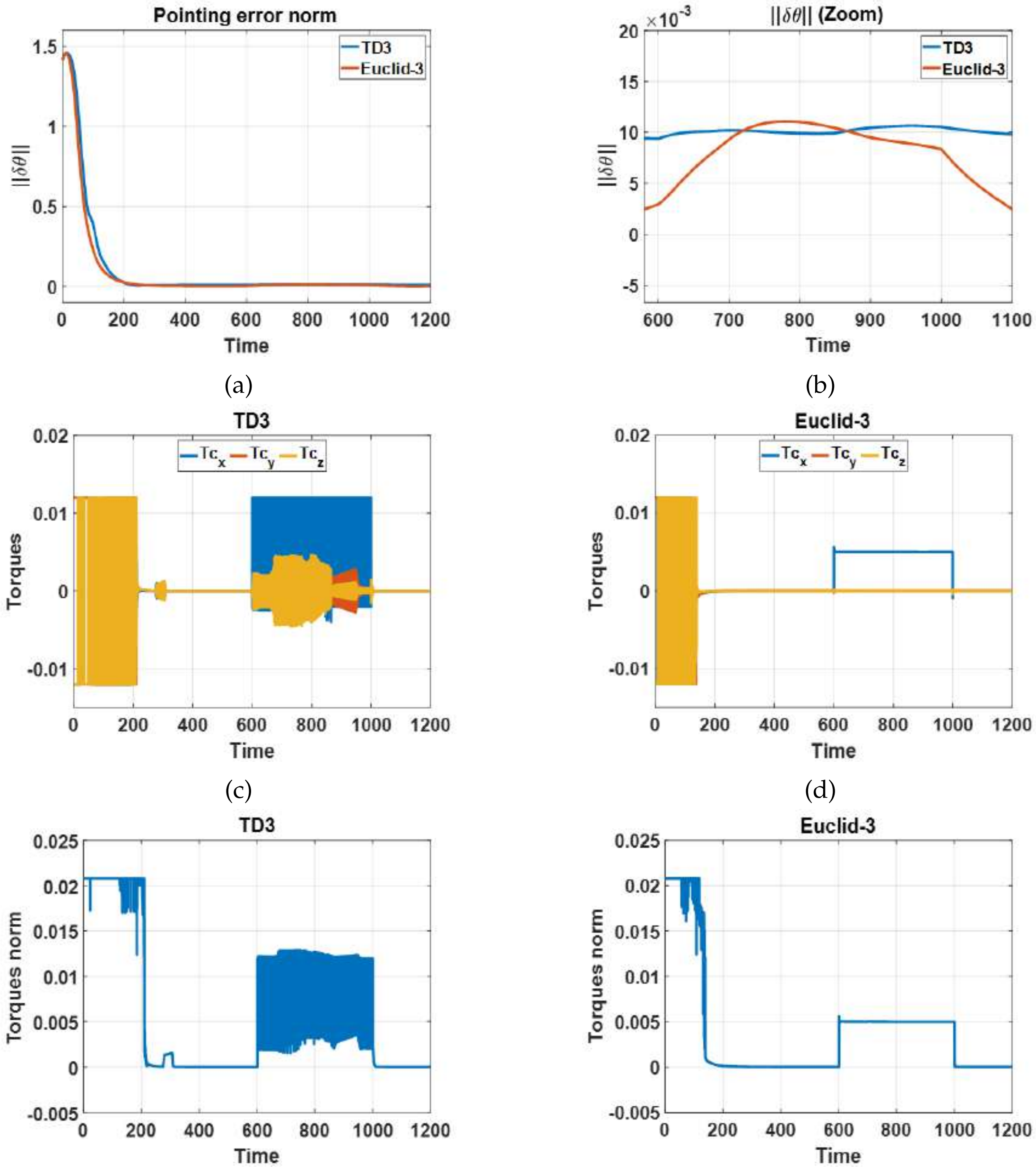
Figure 4.11: Regulation performance in gyro fault case.



(a) (b)

Figure 4.12: $||\delta\theta||$ in regulation case with gyro fault.

mizing pointing accuracy and energy consumption within the same training horizon, leading to a more optimal outcome.



(a)

(b)

(c)

(d)

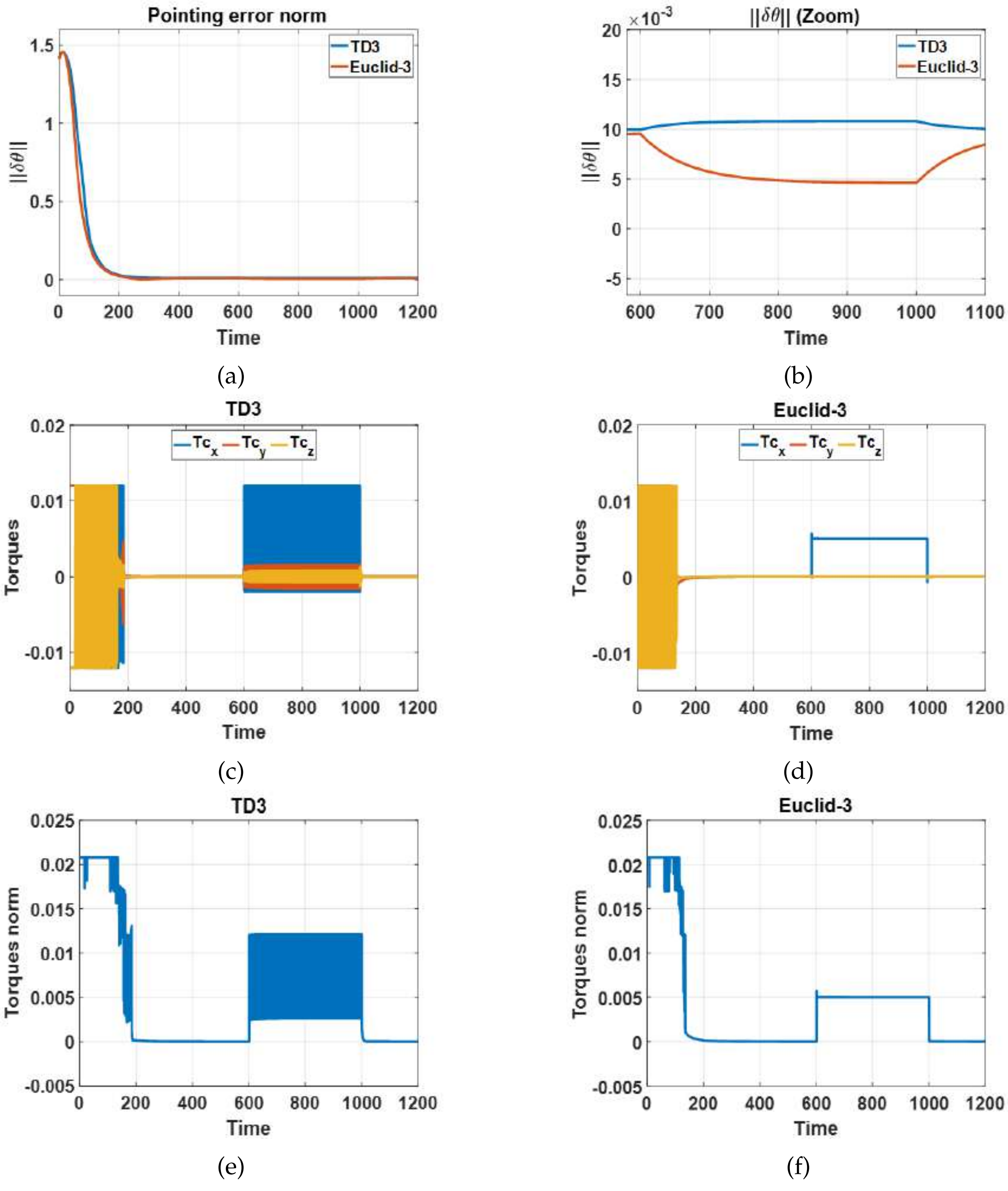
(e)

(f)

Figure 4.13: $\delta\theta$ and Tc in tracking case with actuator fault.

4.5.5 | RL agent performance in case of inertia uncertainty

To assess the robustness of the proposed method, we introduce a time-varying inertia matrix into the MDP simulation. In the previous simulations, we utilized the inertia



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.14: $\delta\theta$ and Tc in regulation case with actuator fault.

$[I]$ such as: $[I] = \begin{bmatrix} 14 & 0 & 0 \\ 0 & 14 & 0 \\ 0 & 0 & 14 \end{bmatrix} Kg.m^2$, which aligns with microsatellite configurations. However, for the Y axis, we modified the value of $[I]$ according to the formula: $I_{yy}^m = I_{yy} + 3sin(0.02\pi t)$.

Furthermore, the reference trajectory is determined based on the angular velocity $\omega_{ref} = \begin{cases} 0.0075sin(0.01\pi t) \\ 0 \qquad\qquad\qquad\qquad rad/s \\ 0.005cos(0.015\pi t) \end{cases}$ while maintaining the same initial conditions as discussed in the preceding section concerning tracking.

To evaluate the robustness of EGE-TD3 in comparison to TD3, we compare three key quantities: Euler angles (ψ, θ, ϕ), control signals (torques), and the norm of the vector part of the attitude quaternion error $(\|q_v\|)$. This last metric measures the disparity between the reference and actual attitude quaternions. Figures 4.15 and 4.16 illustrate the Euler angles and quaternion error, respectively.

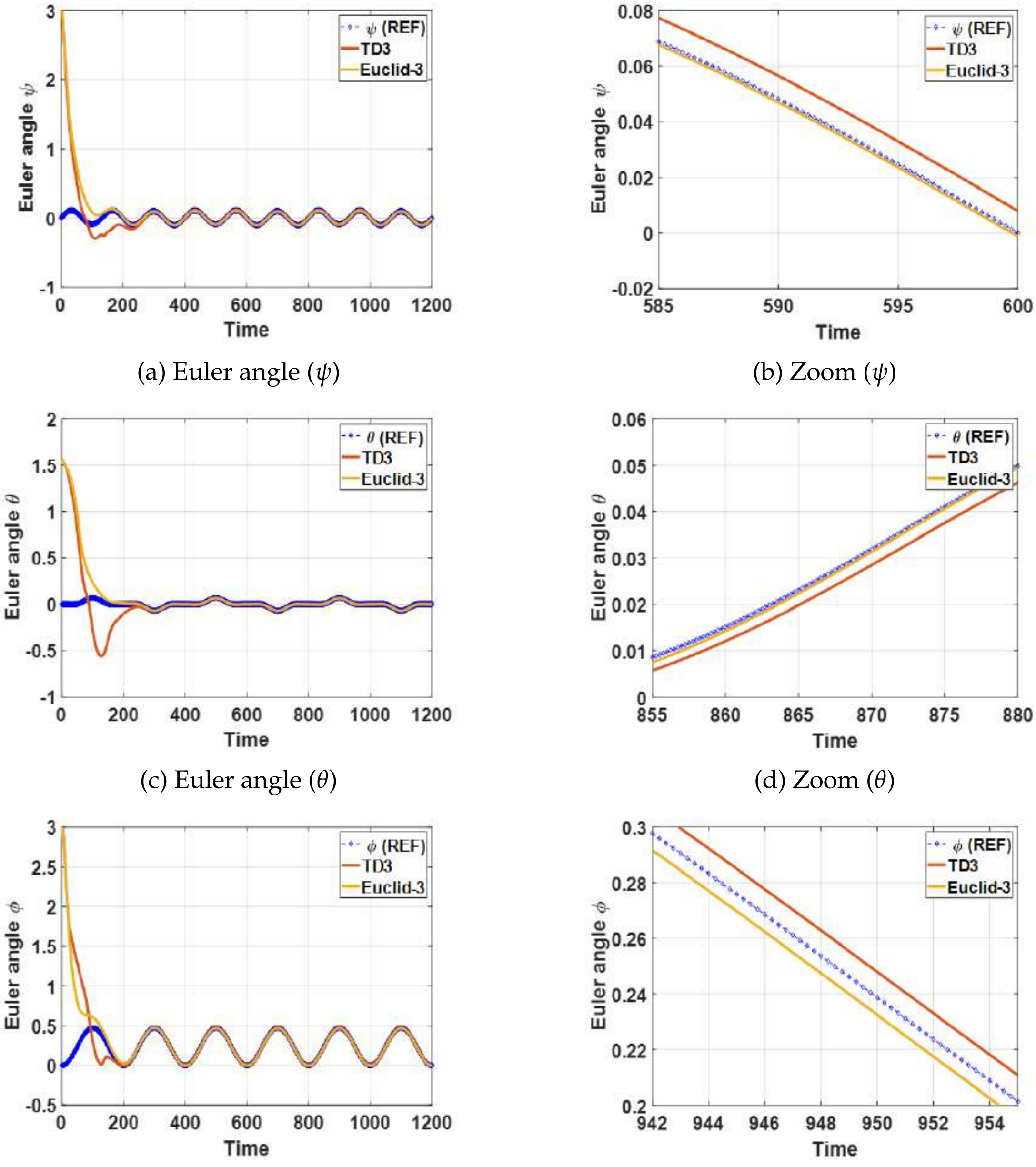
In the steady-state regime, the Euler angles obtained with our EGE agent exhibit significantly closer alignment with the reference values, as evident in Figures 4.15(b), 4.15(d), and 4.15(f). Furthermore, the results concerning $\|q_v\|$ (Figure 4.16) highlight enhanced uncertainty rejection with reduced peaks in $\|q_v\|$.

Additionally, when examining the control signals generated by the agents, as depicted in Figure 4.17, it becomes apparent that the EGE agent produces more suitable torque profiles across all axes without the noisy fluctuations observed around 700s and 1100s caused by the cumulative effects of inertia uncertainty. These findings underscore the EGE agent's superior ability to cope with inertia uncertainty, enhancing overall robustness.

4.6 | Summary

In this chapter, we introduced a novel method aimed at guiding the exploration of reinforcement learning (RL) agents to expedite training and uncover superior policies. The proposed Expert Guided Exploration (EGE) approach demonstrates remarkable efficiency when dealing with Markov decision processes that involve a substantial number of training episodes. It's important to note that our technique is not a self-directed learning algorithm; rather, it facilitates the agent's exploration of the action space by evaluating the similarity of its outputs to a reference control, which is then integrated into the reward-shaping process.

Through numerical simulations in the context of RL-based satellite attitude control, we validate the effectiveness of our proposed method in enhancing agent learning. Additionally, EGE showcases its ability to enable the instructed agent to stabilize the spacecraft's attitude more efficiently than the TD3 agent. Notably, the EGE agent exhibits superior robustness when facing challenges such as sensor issues, actuator failures, and



(a) Euler angle (ψ)

(b) Zoom (ψ)

(c) Euler angle (θ)

(d) Zoom (θ)

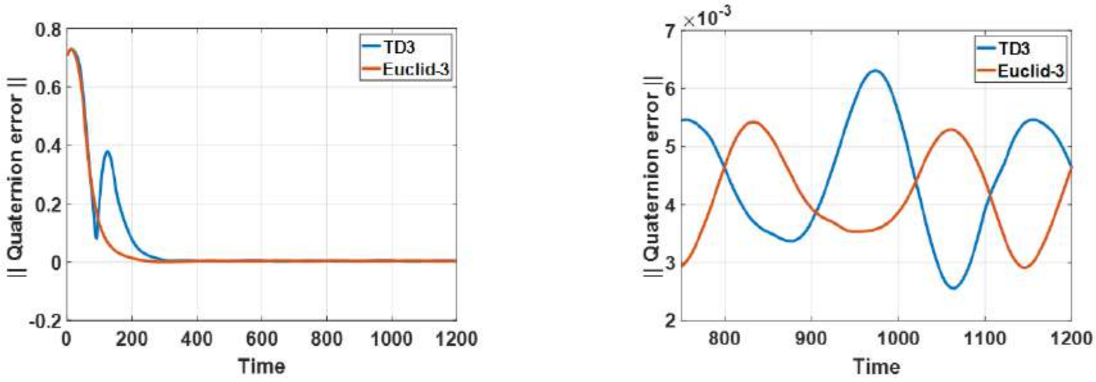
(e) Euler angle (ϕ)

(f) Zoom (ϕ)

Figure 4.15: ψ, θ, ϕ in case of inertia uncertainty.

parameter uncertainties.

However, it is essential to recognize that our approach may not excel in scenarios characterized by relatively straightforward dynamics, such as the control of a water tank system. In such cases, the TD3 agent may rapidly identify optimal solutions, while the EGE agent may not offer more advantageous policies. Furthermore, configuring



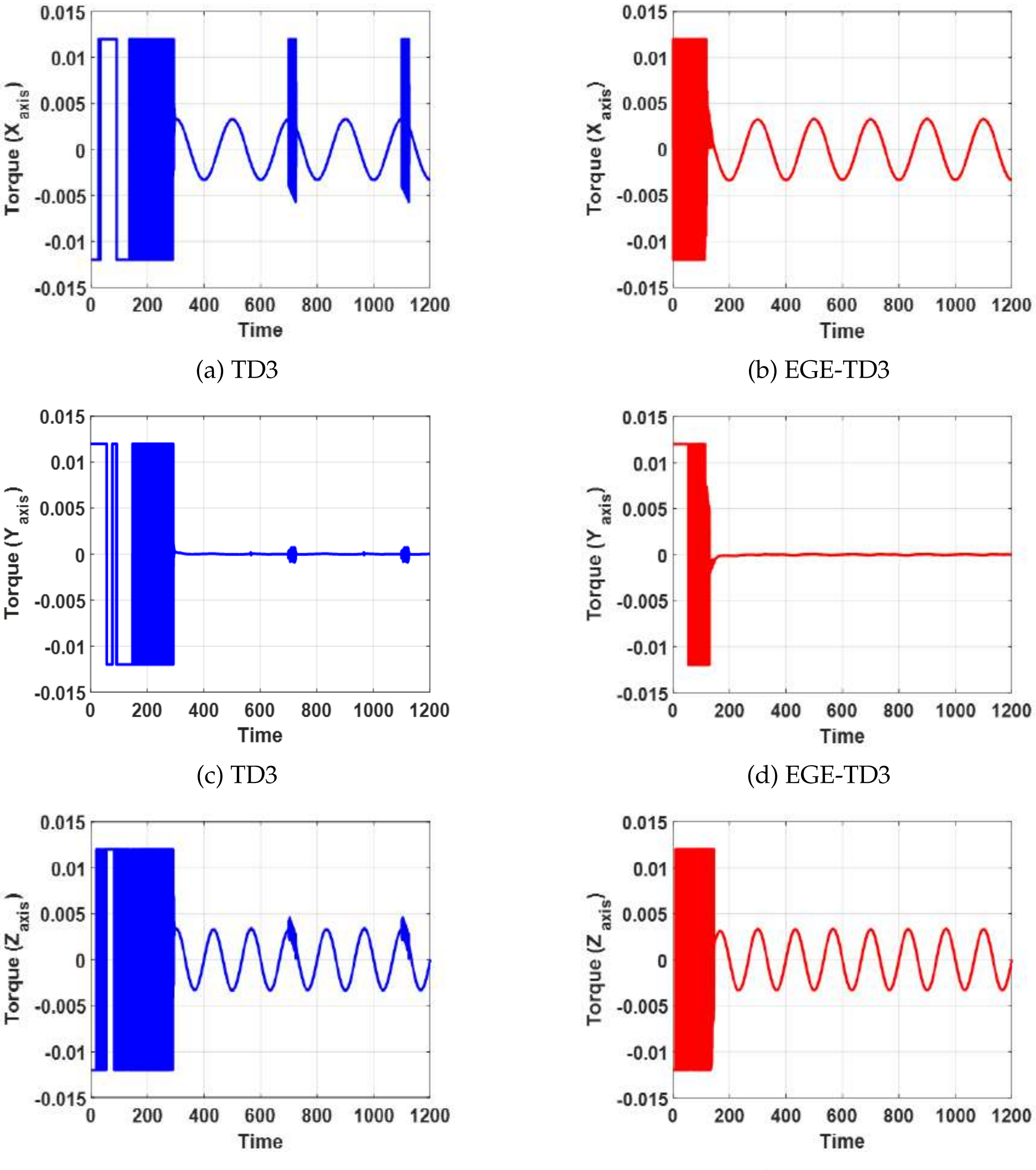
(a) $\|q_v\|$

(b) Zoom ($\|q_v\|$)

Figure 4.16: $\|q_v\|$ in case of inertia uncertainty.

EGE settings for reference control and similarity metrics demands careful consideration. Typically, this task is carried out by a human expert, preventing the approach from achieving full autonomy and necessitating multiple trial-and-error iterations to determine the optimal EGE configuration.

Future research endeavours aimed at further enhancing the proposed approach will focus on developing adaptive mechanisms for specifying the similarity metric. This will enable the algorithm to autonomously determine the optimal shaping interval without the need for human parameterization. The future work and directions to extend the proposed scheme will be further discussed in chapter 5.



(a) TD3

(b) EGE-TD3

(c) TD3

(d) EGE-TD3

(e) TD3

(f) EGE-TD3

Figure 4.17: Torques comparison in case of inertia uncertainty.

5

Conclusions and future work

5.1 | Achieved Aims and Objectives

Faults in spacecraft subsystems can significantly affect mission efficiency and safety. Since spacecraft often operate autonomously and independently, developing reconfiguration and decision-making strategies is of utmost importance. As demonstrated in Chapter 2, recent literature has been dedicated to methods and approaches to enhance fault diagnosis and fault-tolerant systems over the past few decades. However, attaining this objective is a challenging task due to various factors:

- In the context of space-based platforms, the feasibility of prompt ground operator interventions is extremely challenging, if not entirely impossible. This challenge is particularly evident in interplanetary missions like *Cassini* and *New Horizons*,
- The development of model-based approaches sometimes falls short in achieving the desired performance, primarily due to the difficulty in obtaining a reliable physical model, especially when dealing with multiple flexible modes,
- As explained in the example of MMAE in Section 2.3.1.2, the allocation of an entire block (such as KF) for each fault or failure mode is computationally inefficient and cannot effectively handle all conceivable scenarios,
- Relying solely on ML techniques, as seen in the case of supervised learning, demands extensive human expert work, particularly during the labelling stage.

In response to these challenges, the primary focus of this dissertation is on the fusion between model-based and data-driven approaches to enhance fault-tolerant performance and increase system autonomy. As a result, this dissertation presents three key contributions aimed at developing satellite FDD and FTC schemes that rival or surpass the performance of other methods.

The first contribution addresses Gyro fault classification by exclusively relying on statistical features extracted from pre-processed sensor data. This led to the creation of the Variability-based Self-Adaptive Dynamical Classification (VSADC) algorithm, which is extensively detailed in Chapter 2, Section 2.3.2.2.

Subsequently, to enhance the performance of VSADC and enable it to estimate fault magnitudes in Gyro readings roughly, a novel gradient-based FDD algorithm was developed in Chapter 3. This algorithm, Variability-Gradient-Based Self-Adaptive and Dynamical Classification (VGSADC) evaluates additional features from predefined residuals to enhance classification and identification performance. It leverages information related to closed-loop control dynamics to expedite FDD and improve its accuracy. Furthermore, the Active Fault Tolerant Control (AFTC) scheme, elaborated in the same chapter, combines VGSADC and model-based FDD to ensure superior pointing performance by setting the appropriate system reconfiguration.

Recognizing the insufficient exploration of the action space in certain prior research efforts, the final contribution of this dissertation (See Chapter 4) centres on leveraging model-based control signals to assist a reinforcement learning agent in discovering improved policies. It's worth noting that our reward-shaping technique is strategically introduced to avoid interfering with the core policy gradient algorithm. Therefore, this shaping is exclusively applied during the initial training phase.

The implementation of our approach has demonstrated its effectiveness when contrasted with the standard TD3 agent. This comparison highlights our method's enhanced performance and robustness in the context of fault-tolerant control and its ability to withstand uncertainties associated with satellite inertia.

5.2 | Critique and Limitations

While the methods presented in this dissertation have demonstrated superior performance compared to certain state-of-the-art approaches, it is important to delve further into some aspects to elucidate the limitations of our proposed techniques.

Firstly, the definition of the width of the sliding window (SW), as employed in the VSADC algorithm introduced in Chapter 2, relies on human expert judgment. Consequently, it lacks full autonomy. Furthermore, determining the optimal trade-off for the SW can be challenging. Larger windows enhance classification accuracy but at the cost of delayed detection, while shorter windows accelerate detection but result in a higher false positive rate.

On the other hand, although the second algorithm (VGSADC) employed in Chapter 3 still experiences detection delays due to the utilization of the SW technique, the gain adaptation integrated into that scheme helps mitigate excessive pointing errors. Furthermore, improvements in handling transients lead to enhanced fault identification results. Regarding the issue of autonomy, the scheme proposed in the same chapter necessitates the pre-definition of a reconfiguration database, which entails additional design effort and time.

Finally, our proposed EGE scheme, as elaborated in Chapter 4, becomes less advantageous when it comes to reinforcement learning (RL) agents governing MDPs with relatively straightforward dynamics. In such cases, fundamental algorithms such as DDPG and TD3 can efficiently and rapidly learn, rendering EGE an unnecessary overhead. Moreover, implementing EGE necessitates the augmentation of the MDP by configuring an appropriate similarity metric and reference control scheme, thus preventing the approach from achieving full autonomy and requiring multiple trial-and-error scenarios to find the optimal solution.

5.3 | Future Work

The progress achieved in this dissertation, whether through integrating model-based and data-driven methods for FTC or implementing an enhanced RL agent for attitude control, has paved the way for potential future expansions. These future directions can be outlined as follows:

5.3.1 | Feature Engineering and its promising application in spacecraft FDD

Sensor fusion plays a pivotal role in attitude determination. The use of Kalman filters as a sensor fusion method is well-established in both academic and industrial literature, contributing significantly to Gyro drift estimation, which can be viewed as a fault identification system [43, 44, 2, 129]. Similarly, when one sensor is considered more reliable than another, defining residuals becomes the cornerstone for designing various data-driven diagnosis systems. Our initial contribution was conceived based on this fundamental principle, as presented in [45]. Consequently, we envision the potential for leveraging new variations of parameters, measurements, residuals, and more through the application of feature engineering. This enriched feature space can then feed data-based or ML systems, yielding a more reliable, precise, and autonomous FDD model. While a rudimentary version of this framework has been addressed in this thesis, a more intricate scheme can be developed by introducing and engineering novel features. For instance, certain inherent characteristics of linear transformations can be utilized to assess the degree of divergence associated with faults such as drifts. Preliminary results, as illustrated in Figure 5.1, lend support to this concept. An illustrative outline of the proposed new FDD/FTC system is depicted in Figure 5.2.

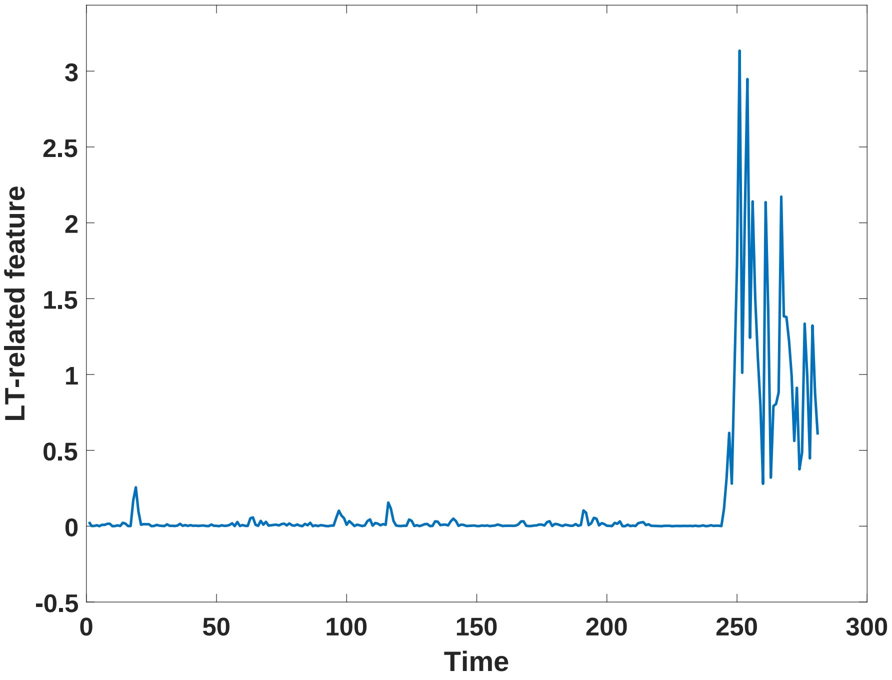


Figure 5.1: Residual evolution using new feature engineering.

5.3.2 | Improving the RL agent training by efficient sampling

The primary reason for using the policy gradient theorem in RL pertains to the challenges associated with implementing a greedy policy improvement in the context of a continuous action space [117]. In the realm of deterministic policy gradient algorithms, for each visited state s, the updating of policy parameters θ^{k+1} is executed in proportion to the gradient of the action-value function $\nabla_\theta Q^{\mu^k}(s, \mu_\theta(s))$. Notably, each state suggests a distinct direction for policy enhancement, which has been addressed within the RL literature by aggregating these gradients with respect to the state distribution

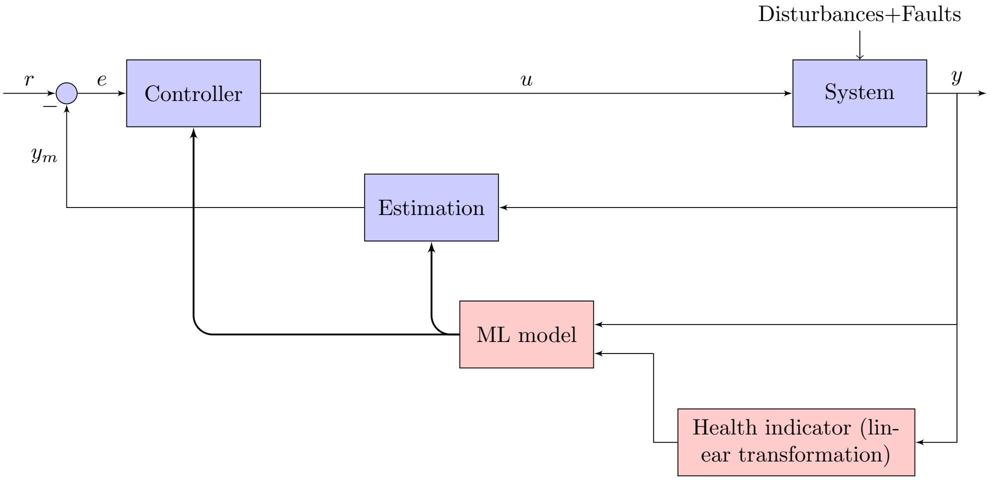


Figure 5.2: FDD/FTC perspective based on new feature engineering.

denoted as $\rho^{\mu}(s)$. The policy update can thus be formally represented as follows:

$$\theta_{k+1} = \theta_k + \alpha \mathbb{E}_{s \sim \rho^{\mu^k}} [\nabla_\theta Q^{\mu^k}(s, \mu_\theta(s))] \tag{5.1}$$

While this formulation serves as the general approach aimed at mitigating the effects of state diversity, the second contribution of this thesis harnesses the expert-guided exploration technique to ameliorate this issue further.

Nevertheless, additional improvements can be made by applying an efficient state trajectory sampling strategy. Knowing that the latter results from agent-environment interactions, it is noteworthy that system dynamics can give rise to organized concentrations or clusters within the state space. Consequently, a judicious sampling method informed by these observations can augment learning efficiency by mitigating the challenges posed by state diversity, as previously alluded to.

A

Appendix A

In appendix A, the reinforcement learning setup used in Chapter 4 will be detailed. This setup has two parts, the RL model (See Figure A.1) and the training setup. The latter is given below:

```
% Setup program AOCS Reinforcement learning framwork
% Create AOCS Environment Interface with Bus
% This script interacts with the model specified by the
   variable "mdl" below,
% in "mdl", the environment dynamics and reward function will
   be defined.
% Episode random initialization, agent and training options
   are defined here.

% Specify the model

mdl = 'rlAOCSModel_one_agent_2022_03_09_reduced_q';
% open_system(mdl);

% Specify the agent block path.
blks = mdl + "/TD3_Agent_for_AOCS";

% Constants
qe_max = 0.18;
qe_min = 0.09;
omega_max = 0.015;
```

```
gain_sim = .45;% gain of similarity metric as specidied in
   Table 4.4
% Similarity function type
DIST = 3;% gauss = 1, none = 2, Euclidean = 3
shaping = [1,1200];%Shaping interval
Tlim = 0.012; % Maximum torque value
% Create the action and observation specification objects.

numobs = 17;
numAct = 3;
lm = -ones(numobs,1);
obsv_Info = rlNumericSpec([numobs 1],...
'LowerLimit',lm,...
'UpperLimit',-lm);
obsv_Info.Name = 'observations';
obsv_Info.Description = 'Q, Q_error, W, W_error, Tc_ref';

action_Info = rlNumericSpec([numAct 1],...
'LowerLimit',-Tlim*ones(numAct,1),...
'UpperLimit',Tlim*ones(numAct,1));
action_Info.Name = 'tau';
action_Info.Description = 'tau_X, tau_Y, tau_Z';

% RL environment creation for spacecraft AOCS.

ACS_env = rlSimulinkEnv(mdl,blks,obsv_Info,action_Info);
% disp(ACS_env)

% Reset function
ACS_env.ResetFcn = @(in)localResetFcn(in);

% Specify "T_f": the episode duration, and "T_s": the agent
   sample time in seconds.

T_s = 0.25;
T_f = 35;
```

```
rng(0)
max_stps = ceil(Tf/Ts);

% Create TD3 Agent

% Agent options and creation
Ts_agent = T_s;
ag_Options = rlTD3AgentOptions("SampleTime",Ts_agent, ...
"DiscountFactor", 0.995, ...
"SaveExperienceBufferWithAgent",false,...
"ResetExperienceBufferBeforeTraining",false,...
"ExperienceBufferLength",2e6, ...
"MiniBatchSize",512, ...
"NumStepsToLookAhead",1, ...
"TargetSmoothFactor",0.005, ...
"TargetUpdateFrequency",10);

kernal_value = ones(3,1);
ag_Options.ExplorationModel = rl.option.
   OrnsteinUhlenbeckActionNoise;
ag_Options.ExplorationModel.StandardDeviation = 2e-3*
   kernal_value;
ag_Options.ExplorationModel.StandardDeviationDecayRate = 9e-7*
   kernal_value;
ag_Options.ExplorationModel.StandardDeviationMin = 0*
   kernal_value;
ag_Options.TargetPolicySmoothModel.Variance = 0.1;
ag_Options.TargetPolicySmoothModel.VarianceDecayRate = 1e-4;

TD3_AOCS = rlTD3Agent(getActor(obsv_Info,numobs,action_Info,
   numAct),...
getCritic(obsv_Info,numobs,action_Info,numAct),...
ag_Options);

% Agent training options
```

```
max_episds = 2e3;

training_Options = rlTrainingOptions(...
'MaxEpisodes',max_episds,...
'MaxStepsPerEpisode',max_stps,...
'ScoreAveragingWindowLength',40,...
'Verbose',false,...
'Plots','training-progress',...
'StopTrainingCriteria','AverageReward',...
'StopTrainingValue',0.5e6);

% Agent training

w=warning('off','all');
% tr_noshaping = train(TD3_AOCS,ACS_env,training_Options);
tr_euclid_1 = train(TD3_AOCS,ACS_env,training_Options);
% tr_gauss_1_sig1 = train(TD3_AOCS,ACS_env,training_Options);

% Reset function
function in = localResetFcn(in)
Wmax = evalin('base','omega_max');
mdl = evalin('base','mdl');
persistent numep
if isempty(numep)
numep = 0;
end

%1- Randomely generate Q_init for REF generation and SAT
   kinematics
Q = 2*rand(1,4)-(ones(1,4));
Q = quatnormalize(Q);
blk = mdl +"/Q";
in = setBlockParameter(in,blk,'Value',mat2str(Q));
Q0 = Q + (0.005*(rand(1,4)-(.5*ones(1,4))));
Q0 = quatnormalize(Q0);
```

```
blk = mdl +"/Q0";
in = setBlockParameter(in,blk,'Value',mat2str(Q0));

%2- Randomely generate W_ref (REF generation) and W0 for SAT
    dynamics
W = (rand([3 1])-(.5*ones(3,1)))*2*Wmax;
blk = mdl +"/W";
in = setBlockParameter(in,blk,'Value',mat2str(W));

W0 = W + .0005*(rand([3 1])-(.5*ones(3,1)));
W0 = max(-Wmax*ones(3,1),W0);
W0 = min(Wmax*ones(3,1),W0);
blk = mdl +"/W0";
in = setBlockParameter(in,blk,'Value',mat2str(W0));
blk = mdl +"/CLK";
in = setBlockParameter(in,blk,'Value',mat2str(numep));
numep = numep + 1;
end
function actor_RL = getActor(obsv_Info,num_Obsv,act_Info,
    num_Act)
act_Net = [
featureInputLayer(num_Obsv,'Normalization','none','Name','Stat
    ')
fullyConnectedLayer(64, 'Name','actor_FC1')
reluLayer('Name','relu1')
fullyConnectedLayer(32, 'Name','actor_FC2')
reluLayer('Name','relu2')
fullyConnectedLayer(num_Act,'Name','Act')
tanhLayer('Name','tanh_1')];
act_Options = rlRepresentationOptions('LearnRate',3e-3,'
    GradientThreshold',1,'L2RegularizationFactor',0.001);
actor_RL = rlDeterministicActorRepresentation(act_Net,
    obsv_Info,act_Info,...
'Observation',{'Stat'},'Action',{'tanh_1'},act_Options);
end
```

```
function critic = getCritic(obsv_Info,num_Obsv,act_Info,
   num_Act)

state_Path = [featureInputLayer(num_Obsv,'Normalization','none
   ','Name','Stat')
fullyConnectedLayer(64,'Name','fc_1')];
action_Path = [featureInputLayer(num_Act, 'Normalization', '
   none', 'Name','Act')
fullyConnectedLayer(64, 'Name','fc_2')];
common_Path = [additionLayer(2,'Name','add')
reluLayer('Name','relu2')
fullyConnectedLayer(32, 'Name','fc_3')
reluLayer('Name','relu3')
fullyConnectedLayer(16, 'Name','fc_4')
fullyConnectedLayer(1, 'Name','Cri_Output')];
cri_Net = layerGraph();
cri_Net = addLayers(cri_Net,state_Path);
cri_Net = addLayers(cri_Net,action_Path);
cri_Net = addLayers(cri_Net,common_Path);
cri_Net = connectLayers(cri_Net,'fc_1','add/in1');
cri_Net = connectLayers(cri_Net,'fc_2','add/in2');

cri_Options = rlRepresentationOptions('LearnRate',3e-4,'
   GradientThreshold',1);
cri_1 = rlQValueRepresentation(cri_Net,obsv_Info,act_Info,...
'Observation',{'Stat'},'Action',{'Act'},cri_Options);
cri_2 = rlQValueRepresentation(cri_Net,obsv_Info,act_Info,...
'Observation',{'Stat'},'Action',{'Act'},cri_Options);

critic = [cri_1,cri_2];

end
```

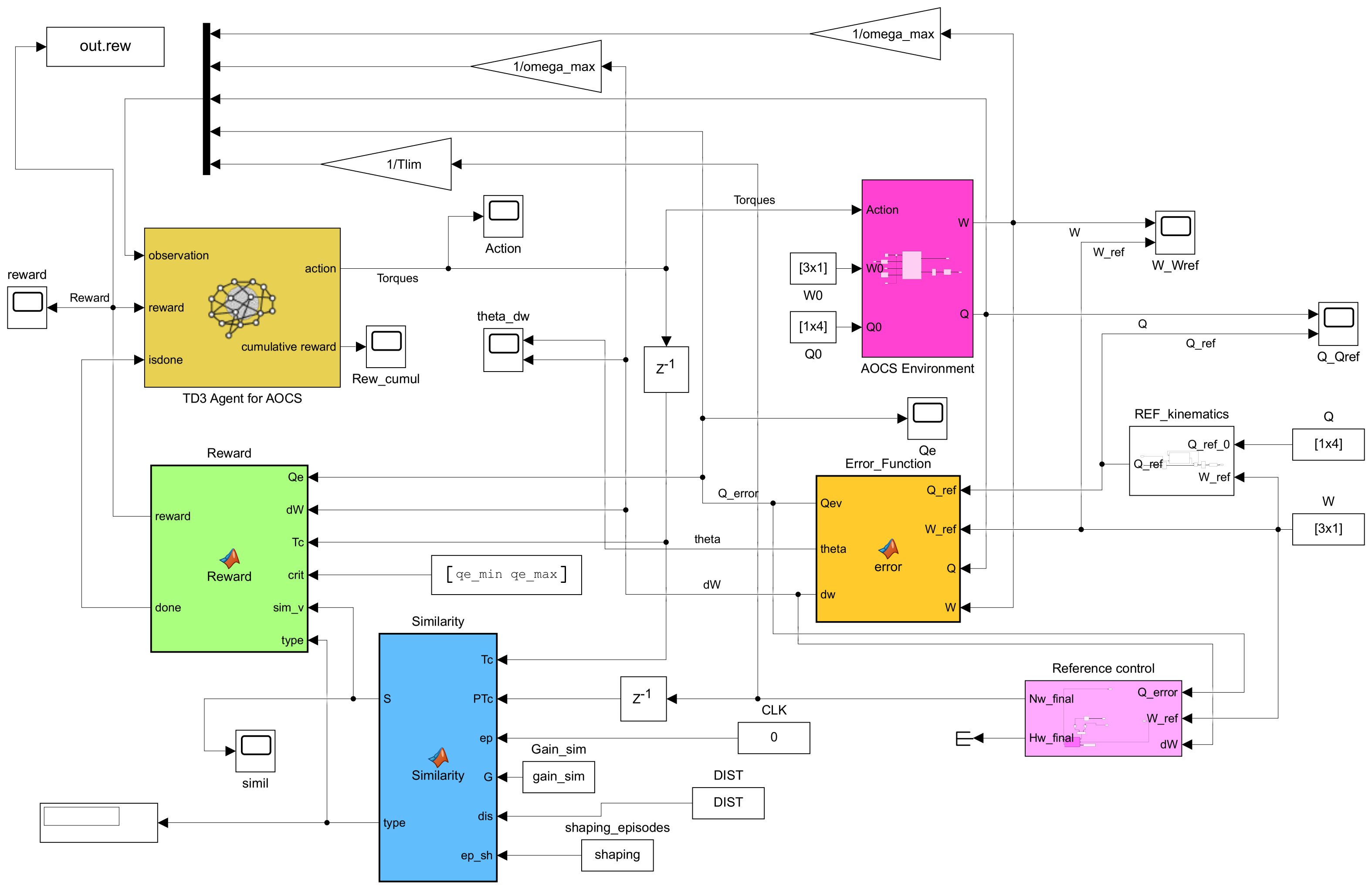


Figure A.1: Simulink model of RL setup (Chapter 4)

Within this model, a conventional attitude control framework employing reinforcement learning has been enhanced through the integration of a reward shaping module. The specifics of the latter are outlined as follows:

```
function [S,type] = Similarity(Tc,PTc,ep,G,dis,ep_sh)
% Inputs:
% Tc(1x3 real array): Last applied torque
% PTc(1x3 real array): Last computed torque of
   reference
% ep(integer): Current episode number
% G: gain. In principle, max output similarity measure
    is 1.
%       ==> This param is used to scale the max output
   .
% dis(string): Statistical distribution used to
   evaluate similarity.
%       Values: 'Gauss', 'Euclid', 'None'.
% ep_sh(integer): Episode number after which no more
   similarity measure.
%       ==> After this value, the standard reward is
   used.
% Outputs:
% S (real): Similarity value,
% type: the type used.

switch dis
case 1
distrib = 'Gauss';
case 2
distrib = 'None';
case 3
distrib = 'Euclidean';
otherwise
distrib = 'None';
end

type = 0;S = 0;
```

```
if ep > ep_sh(1) && ep <= ep_sh(2)
% Gaussian function
sigma_angle = 1; % std
mean_angle = 0;% mean
sigma_norm = 1;% std
mean_norm = 1;% mean
max_ratio = 1e3;

% evaluate the angle between vectors:
CosTheta = max(min(dot(PTc,Tc)/(norm(PTc)*norm(Tc)),1)
   ,-1);
Theta = real(acos(CosTheta));
Theta = max(0,Theta);
Gauss_Theta = gaussian(Theta,sigma_angle,mean_angle);
if norm(PTc)*norm(Tc) == 0
Ratio = max_ratio;
else
Ratio = max(min(norm(PTc)/norm(Tc),max_ratio),1/
   max_ratio);
end

Gauss_ratio = gaussian(Ratio,sigma_norm,mean_norm);


if strcmp(distrib, 'Euclidean')
v = 1/(0.01+(sum((PTc-Tc).^2).^0.5));
S = G*v;
type = 3;
end

if strcmp(distrib, 'Gauss')
S = G*(Gauss_Theta*Gauss_ratio);
type = 1;
end
```

```
if strcmp(distrib, 'None')
S = 0;
type = 2;
end

else % Remaining Episodes of training, no more reward
   adjustment.
S = 0;
end

function g = gaussian(value,sigma,mean)
% gaussian function
g = exp(-.5*(((value-mean)/sigma)^2));
end

end
```

References

[1] H.J. Kramer. Observation of the earth and its environment: survey of missions and sensors, electronic update of 4th ed, accessible as well on esa earth observation portal directory. https://www.eoportal.org/satellite-missions/alsat-2. Accessed: Apr, 28th 2024.

[2] F. Landis Markley and John L. Crassidis. *Fundamentals of spacecraft attitude determination and control*. Springer Science & Business Media, 2014.

[3] Pedro Larrañaga, David Atienza, Javier Diaz-Rozo, Alberto Ogbechie, Carlos Puerto-Santana, and Concha Bielza. *Industrial applications of machine learning*. Data Mining and Knowledge. CRC press, 2018.

[4] A H de Ruiter, C Damaren, and J R Forbes. *Spacecraft Dynamics and Control: An Introduction*. John Wiley & Sons, 2012.

[5] Hanspeter Schaub and John L. Junkins. *Analytical Mechanics of Space Systems, Third Edition*. AIAA education series, 2014.

[6] Yuan Chai, Jianjun Luo, and Weihua Ma. Data-driven game-based control of microsatellites for attitude takeover of target spacecraft with disturbance. *ISA Transactions*, 119, 2022. doi: https://doi.org/10.1016/j.isatra.2021.02.037.

[7] Zhifeng Gao, Zepeng Zhou, Moshu S. Qian, and Jinxing Lin. Active fault tolerant control scheme for satellite attitude system subject to actuator time-varying faults. *IET Control Theory & Applications*, 12(3):405–412, feb 2018. doi: https://doi.org/10.1049/iet-cta.2017.0969.

[8] Jean-Marc Biannic, Christelle Pittet, L. Lafourcade, and C. Roos. LPV analysis of switched controllers in satellite attitude control systems. In *AIAA Guidance, Navigation, and Control Conference*, Reston, Virigina, aug 2010. American Institute of Aeronautics and Astronautics.

[9] C. Pittet and D. Arzelier. Demeter: A benchmark for robust analysis and control of the attitude of flexible micro satellites. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 5, 2006.

[10] Hicham Henna, Houari Toubakh, Mohamed Redouane Kafi, and Moamar Sayed-Mouchaweh. Towards fault-tolerant strategy in satellite attitude control systems: A review. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, volume 12, 2020.

[11] G.J.J. Ducard. *Fault-tolerant Flight Control and Guidance Systems: Practical Methods for Small Unmanned Aerial Vehicles*. Advances in Industrial Control. Springer London, 2009.

[12] ESA (F.C. Vandenbussche). Soho's recovery - an unprecedented success story (bulletin 97). https://www.esa.int/esapub/bulletin/bullet97/vandenbu.pdf, 1999. Accessed: 21.07.2022.

[13] Michele Johnson (Ames Research Center). Nasa ends attempts to fully recover kepler spacecraft - potential new missions considered. https://www.nasa.gov/feature/ames/kepler/nasa-ends-attempts-to-fully-recover-kepler-spacecraft-potential-new-missions-considered, 2013. Accessed: 21.07.2022.

[14] JAXA. Supplemental handout on the operation plan of the x-ray astronomy satellite astro-h (hitomi). https://global.jaxa.jp/press/2016/04/files/20160428_hitomi.pdf, 2016. Accessed: 21.07.2022.

[15] I. Samy and D.W. Gu. *Fault Detection and Flight Data Measurement: Demonstrated on Unmanned Air Vehicles using Neural Networks*. Lecture Notes in Control and Information Sciences. Springer Berlin Heidelberg, 2012.

[16] Sheng Gao, Wei Zhang, and Xu He. Observer-based multiple faults diagnosis scheme for satellite attitude control system. *Asian Journal of Control*, 22(1):307–322, 2020. doi: https://doi.org/10.1002/asjc.1873.

[17] H Alwi, C Edwards, and A Marcos. Fdi for a mars orbiting satellite based on a sliding mode observer scheme. In *2010 Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 125–130. IEEE, 2010.

[18] Indira Nagesh and Christopher Edwards. A sliding mode observer based fdi scheme for a nonlinear satellite systems. In *2011 IEEE International Conference on Control Applications (CCA)*, pages 159–164. IEEE, 2011.

[19] Farzin Nemati, Seyed Mostafa Safavi Hamami, and Ali Zemouche. A nonlinear observer-based approach to fault detection, isolation and estimation for satellite formation flight application. *Automatica*, 107:474–482, 2019. doi: https://doi.org/10.1016/j.automatica.2019.06.007.

[20] David Henry. Fault diagnosis of microscope satellite thrusters using H -infinity/H_ filters. *Journal of Guidance, Control, and Dynamics*, 31(3):699–711, 2008. doi: https://doi.org/10.2514/1.31003.

[21] P Baldi, Mogens Blanke, P Castaldi, N Mimmo, and S Simani. Fault diagnosis for satellite sensors and actuators using nonlinear geometric approach and adaptive observers. *International Journal of Robust and Nonlinear Control*, 29(16):5429–5455, 2019. doi: https://doi.org/10.1002/rnc.4083.

[22] Sheng Gao, Zhaowei Zhang, Wei Zhang, Xu He, and Xiaodong Lu. Fault Diagnosis for Satellite Attitude Control System with Using Extended Kalman Filter. In *2019 Chinese Control Conference (CCC)*, pages 4789–4794. IEEE, jul 2019. doi: https://doi.org/10.23919/ChiCC.2019.8865903.

[23] SooYung Byeon, Sung-Hoon Mok, Hyunwook Woo, and Hyochoong Bang. Sensor-fault tolerant attitude determination using two-stage estimator. *Advances in Space Research*, 63(11):3632–3645, 2019. doi: https://doi.org/10.1016/j.asr.2019.02.007.

[24] Raman Mehra, Constantino Rago, and Sanjeev Seereeram. Autonomous failure detection, identification and fault-tolerant estimation with aerospace applications. In *1998 IEEE Aerospace Conference Proceedings (Cat. No. 98TH8339)*, volume 2, pages 133–138. IEEE, 1998.

[25] Liliang Li, Chuan Liu, Yanzhao Zhang, Zhenhua Wang, and Yi Shen. Gyroscope fault accommodation based on dedicated Kalman filters. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 234(2):254–266, 2020. doi: https://doi.org/10.1177/0954410019861188.

[26] E Lopez-Encarnacion, R Fonod, and P Bergner. Model-based FDI for agile spacecraft with multiple actuators working simultaneously. *IFAC-PapersOnLine*, 52(12):436–441, 2019. doi: https://doi.org/10.1016/j.ifacol.2019.11.282.

[27] Zhi-Feng Gao, Bin Jiang, Peng Shi, and Yue-Hua Cheng. Sensor fault estimation and compensation for microsatellite attitude control systems. *International Journal of Control, Automation and Systems*, 8(2):228–237, 2010. doi: https://doi.org/10.1007/s12555-010-0207-7.

[28] Kirill Djebko, Frank Puppe, and Hakan Kayal. Model-based fault detection and diagnosis for spacecraft with an application for the sonate triple cube nano-satellite. *Aerospace*, 6(10):105, 2019. doi: https://doi.org/10.3390/aerospace6100105.

[29] Yuri Yun, Junyong Lee, Hwa-suk Oh, and Joo-Ho Choi. Remaining useful life prediction of reaction wheel motor in satellites. *JMST Advances*, 1:219–226, 2019. doi: https://doi.org/10.1007/s42791-019-00020-5.

[30] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, 32(2):229–252, 2008. doi: https://doi.org/10.1016/j.arcontrol.2008.03.008.

[31] Zhifeng Gao, Zepeng Zhou, Guoping Jiang, Moshu Qian, and Jinxing Lin. Active Fault Tolerant Control Scheme for Satellite Attitude Systems: Multiple Actuator Faults Case. *International Journal of Control, Automation and Systems*, 16(4), 2018. doi: https://doi.org/10.1007/s12555-016-0667-5.

[32] Aboul Ella Hassanien, Ashraf Darwish, and Hesham El-Askary. *Machine Learning and Data Mining in Aerospace Technology*. Springer, 2020.

[33] Gan-hua Li, Jian-cheng Li, Ya-ni Cao, Min-qiang Xu, Ke-qiang Xia, Jun Wei, Bao-jun Lan, and Li Dong. The flywheel fault detection based on kernel principal component analysis. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 425–432. IEEE, 2019.

[34] Sara K Ibrahim, Ayman Ahmed, M Amal Eldin Zeidan, and Ibrahim E Ziedan. Machine learning techniques for satellite fault diagnosis. *Ain Shams Engineering Journal*, 11(1):45–56, 2020. doi: https://doi.org/10.1016/j.asej.2019.08.006.

[35] Zhi Qu, Kai Xu, Zhigang Chen, Xin He, Yanhao Xie, Mengmeng Liu, Feng Li, and Shuangxue Han. Fault detection method of luojia1-01 satellite attitude control system based on supervised local linear embedding. *IEEE Access*, 7:105489–105502, 2019. doi: https://doi.org/10.1109/access.2019.2932392.

[36] Hasan Abbasi Nozari, Paolo Castaldi, Hamed Dehghan Banadaki, and Silvio Simani. Novel non-model-based fault detection and isolation of satellite reaction wheels based on a mixed-learning fusion framework. *Ifac-papersonline*, 52(12):194–199, 2019. doi: https://doi.org/10.1016/j.ifacol.2019.11.222.

[37] Dengyun Wu, Hong Wang, Hongxing Liu, Tian He, and Tao Xie. Health monitoring on the spacecraft bearings in high-speed rotating systems by using the clustering fusion of normal acoustic parameters. *Applied Sciences*, 9(16):3246, 2019. doi: https://doi.org/10.3390/app9163246.

[38] Ying Liu, Qiang Pan, Hong Wang, and Tian He. Fault diagnosis of satellite flywheel bearing based on convolutional neural network. In *2019 Prognostics and System Health Management Conference (PHM-Qingdao)*, pages 1–6. IEEE, 2019.

[39] Bowen Sun, Jiongqi Wang, Zhangming He, Haiyin Zhou, and Fengshou Gu. Fault identification for a closed-loop control system based on an improved deep neural network. *Sensors*, 19(9):2131, 2019. doi: https://doi.org/10.3390/s19092131.

[40] Kwang-Hyun Lee, SeongMin Lim, Dong-Hyun Cho, and Hae-Dong Kim. Development of fault detection and identification algorithm using deep learning for nanosatellite attitude control system. *International Journal of Aeronautical and Space Sciences*, 21:576–585, 2020. doi: https://doi.org/10.1007/s42405-019-00235-9.

[41] Alexandra Wander and Roger Förstner. Innovative fault detection, isolation and recovery strategies on-board spacecraft: state of the art and research challenges. In *Deutscher Luft- und Raumfahrtkongress 2012*. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV Bonn, Germany, 2013.

[42] Ali Zolghadri, David Henry, Jérôme Cieslak, Denis Efimov, and Philippe Goupil. *Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles - From Theory to Application*. Springer Science & Business Media, 2014.

[43] S Winkler, G Wiedermann, and W Gockel. Gyro-stellar attitude estimation considering measurement noise correlation and time-variant relative sensor misalignment. In *International Astronautical Congress, Glasgow, Scotland, September*, 2008.

[44] Alex Torres, Christine Fallet, Alain Peus, Christelle Pittet, and Pascal Prieur. Myriade microsatellites: Current developments around a proven aocs concept. In *4S Symposium, Small Satellites, Systems and Services*, 2006.

[45] Hicham Henna, Houari Toubakh, Mohamed Redouane Kafi, and Moamar Sayed Mouchaweh. Unsupervised data-driven approach for fault diagnostic of spacecraft gyroscope. In *Annual Conference of the PHM Society*, volume 14, 2022.

[46] Mehdi Ghezal, Bernard Polle, Christophe Rabejac, and Johan Montel. Gyro stellar attitude determination. In *6th International ESA Conference on Guidance, Navigation and Control Systems*, 2005.

[47] Houari Toubakh, Moamar Sayed-Mouchaweh, Mohammed Benmiloud, Michael Defoort, and Mohamed Djemai. Self adaptive learning scheme for early diagnosis of simple and multiple switch faults in multicellular power converters. *ISA transactions*, 113:222–231, 2021. doi: https://doi.org/10.1016/j.isatra.2020.03.025.

[48] Hicham Henna, Elhadi Gasmi, Houari Toubakh, Mohamed Redouane Kafi, Mohamed Djemai, and Moamar Sayed-Mouchaweh. Fault tolerant attitude estimation for satellite at low earth orbit*. In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 275–280, 2023.

[49] Mohamed Abdelbasset Mahboub, Boubakeur Rouabah, Mohamed Redouane Kafi, and Houari Toubakh. Health management using fault detection and fault tolerant control of multicellular converter applied in more electric aircraft system. *Diagnostyka*, 23(2), 2022. doi: https://doi.org/10.29354/diag/151039.

[50] Boubakeur Rouabah, Mohammed Abdelbasset Mahboub, Mohamed Redouane Kafi, Houari Toubakh, Mohamed Djemai, and Lazhar Ben-Brahim. Medium dc voltage power converter for electrified railway with fault tolerant control strategy. In *2023 IEEE International Conference on Smart Mobility (SM)*, pages 121–125, 2023.

[51] Lei Zhang, Zihao Wang, Xiaolin Ding, Shaohua Li, and Zhenpo Wang. Fault-tolerant control for intelligent electrified vehicles against front wheel steering angle sensor faults during trajectory tracking. *IEEE Access*, 9:65174–65186, 2021. doi: https://doi.org/10.1109/ACCESS.2021.3075325.

[52] Arslan Ahmed Amin and Khalid Mahmood ul Hasan. Robust passive fault tolerant control for air fuel ratio control of internal combustion gasoline engine for sensor and actuator faults. *IETE Journal of Research*, 69(5):2846–2861, 2023. doi: https://doi.org/10.1080/03772063.2021.1906767.

[53] Jonas Zinn, Birgit Vogel-Heuser, and Marius Gruber. Fault-tolerant control of programmable logic controller-based production systems with deep reinforcement learning. *Journal of Mechanical Design*, 143(7):072004, 2021. doi: https://doi.org/10.1115/1.4050624.

[54] Pawel Majdzik, Bogdan Lipiec, Marcin Witczak, Lothar Seybold, Ralf Stetter, and Zbigniew Banaszak. A fuzzy logic approach to fault-tolerant scheduling of semi-automated assembly systems. In *2021 5th International Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 261–266, 2021.

[55] H. Alwi, C. Edwards, and C.P. Tan. *Fault Detection and Fault-Tolerant Control Using Sliding Modes*. Advances in Industrial Control. Springer London, 2011.

[56] C. Hajiyev and F. Caliskan. *Fault Diagnosis and Reconfiguration in Flight Control Systems*. Cooperative Systems. Springer US, 2013.

[57] Lang Bai and Zhifeng Gao. Integrated fault-tolerant stabilization control for satellite attitude systems with actuator and sensor faults. *Journal of Control, Automation and Electrical Systems*, 30(6):864–878, 2019. doi: https://doi.org/10.1007/s40313-019-00498-3.

[58] Hui Hu, Lei Liu, Yongji Wang, Zhongtao Cheng, and Qinqin Luo. Active fault-tolerant attitude tracking control with adaptive gain for spacecrafts. *Aerospace Science and Technology*, 98:105706, 2020. doi: https://doi.org/10.1016/j.ast.2020.105706.

[59] Xiaohui Liang, Qing Wang, Changhua Hu, and Chaoyang Dong. Observer-based $\mathcal{H}_\infty$ fault-tolerant attitude control for satellite with actuator and sensor faults. *Aerospace Science and Technology*, 95:105424, 2019. doi: https://doi.org/10.1016/j.ast.2019.105424.

[60] Qingkai Meng, Hao Yang, and Bin Jiang. Second-order sliding-mode on so(3) and fault-tolerant spacecraft attitude control. *Automatica*, 149:110814, 2023. doi: https://doi.org/10.1016/j.automatica.2022.110814.

[61] Yiqi Xu, Xiaodong Shao, Qinglei Hu, and Bin Song. Anti-disturbance fault-tolerant control with flexible performance for combined spacecraft attitude tracking under limited thrusts. *Control Engineering Practice*, 134:105455, 2023. doi: https://doi.org/10.1016/j.conengprac.2023.105455.

[62] Zejun ZHANG, Hao YANG, and Bin JIANG. Fault tolerant attitude control of under-actuated spacecraft: Theory and experiment. *Chinese Journal of Aeronautics*, 36(5):465–474, 2023. doi: https://doi.org/10.1016/j.cja.2023.02.027.

[63] Amin Ziaei, Behzad Sinafar, Hamed Kharrati, and Afshin Rahimi. Concurrent-learning-based event-triggered fault tolerant attitude control for spacecraft with actuator faults. *Advances in Space Research*, 2023. doi: https://doi.org/10.1016/j.asr.2023.09.051.

[64] Maria Khodaverdian and Maryam Malekzadeh. Fault-tolerant model predictive sliding mode control with fixed-time attitude stabilization and vibration suppression of flexible spacecraft. *Aerospace Science and Technology*, 139:108381, 2023. doi: https://doi.org/10.1016/j.ast.2023.108381.

[65] Guan Wang and Hongwei Xia. Fault-tolerant learning control of air-breathing hypersonic vehicles with uncertain parameters and actuator faults. *Expert Systems with Applications*, 238:121874, 2024. doi: https://doi.org/10.1016/j.eswa.2023.121874.

[66] Tiancai Wu, Honglun Wang, Yue Yu, Yiheng Liu, and Jianfa Wu. Hierarchical fault-tolerant control for over-actuated hypersonic reentry vehicles. *Aerospace Science and Technology*, 119:107134, 2021. doi: https://doi.org/10.1016/j.ast.2021.107134.

[67] Batuhan Eroglu, M. Cagatay Sahin, and Nazim Kemal Ure. Autolanding control system design with deep learning based fault estimation. *Aerospace Science and Technology*, 102:105855, 2020. doi: https://doi.org/10.1016/j.ast.2020.105855.

[68] Chengxi Zhang, Ming-Zhe Dai, Jin Wu, Bing Xiao, Bo Li, and Mingjiang Wang. Neural-networks and event-based fault-tolerant control for spacecraft attitude stabilization. *Aerospace Science and Technology*, 114:106746, 2021. doi: https://doi.org/10.1016/j.ast.2021.106746.

[69] Daikun Chao, Ruiyun Qi, and Bin Jiang. Adaptive fault-tolerant attitude control for hypersonic reentry vehicle subject to complex uncertainties. *Journal of the Franklin Institute*, 359(11):5458–5487, 2022. doi: https://doi.org/10.1016/j.jfranklin.2022.05.011.

[70] Mingyue Shi, Baolin Wu, and Danwei Wang. Neural-network-based adaptive quantized attitude takeover control of spacecraft by using cellular satellites. *Advances in Space Research*, 70(7):1965–1978, 2022. doi: https://doi.org/10.1016/j.asr.2022.06.034.

[71] Yuhan Liu, Guangfu Ma, Yueyong Lyu, and Pengyu Wang. Neural network-based reinforcement learning control for combined spacecraft attitude tracking maneuvers. *Neurocomputing*, 484:67–78, 2022. doi: https://doi.org/10.1016/j.neucom.2021.07.099.

[72] Jingbo Fu, Ming Liu, Xibin Cao, and Ang Li. Robust neural-network-based quasi-sliding-mode control for spacecraft-attitude maneuvering with prescribed performance. *Aerospace Science and Technology*, 112:106667, 2021. doi: https://doi.org/10.1016/j.ast.2021.106667.

[73] Runqi Chai, Antonios Tsourdos, Al Savvaris, Senchun Chai, Yuanqing Xia, and C. L. Philip Chen. Six-dof spacecraft optimal trajectory planning and real-time attitude control: A deep neural network-based approach. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):5005–5013, 2020. doi: https://doi.org/10.1109/TNNLS.2019.2955400.

[74] Jixing Lv, Changhong Wang, and Yonggui Kao. Adaptive fixed-time quantized fault-tolerant attitude control for hypersonic reentry vehicle. *Neurocomputing*, 520:386–399, 2023. doi: https://doi.org/10.1016/j.neucom.2022.11.057.

[75] Shen Yin, Bing Xiao, Steven X. Ding, and Donghua Zhou. A review on recent development of spacecraft attitude fault tolerant control system. *IEEE Transactions on Industrial Electronics*, 63(5):3311–3320, 2016. doi: https://doi.org/10.1109/TIE.2016.2530789.

[76] M.A. Djeziri, S. Benmoussa, M. Sayed Mouchaweh, and E. Lughofer. Fault diagnosis and prognosis based on physical knowledge and reliability data: Application to mos field-effect transistor. *Microelectronics Reliability*, 110:113682, 2020. doi: https://doi.org/10.1016/j.microrel.2020.113682.

[77] Edwin Lughofer and Moamar Sayed-Mouchaweh. *Predictive maintenance in dynamic systems: advanced methods, decision support tools and real-world applications*. Springer, 2019.

[78] S.D. Shuster, M.D.; Oh. Three-axis attitude determination from vector observations. *Journal of Guidance and Control*, 4(1):70–77, 1981. doi: https://doi.org/10.2514/3.19717.

[79] Grace Wahba. A least-squares estimate of satellite attitude. *SIAM Review*, 7(3):409, 1965. doi: https://doi.org/10.1137/1007077.

[80] Gerald M Lerner. Three-axis attitude determination. *Spacecraft attitude determination and control*, 73:420–428, 1978.

[81] Hicham Henna, Houari Toubakh, Mohamed Redouane Kafi, Moamar Sayed-Mouchaweh, and Mohamed Djemai. Hybrid supervision scheme for satellite attitude control with sensor faults. *CEAS Space Journal*, pages 1–15, 2024. doi: https://doi.org/10.1007/s12567-024-00548-w.

[82] John L Crassidis, F Landis Markley, and Yang Cheng. Survey of nonlinear attitude estimation methods. *Journal of guidance, control, and dynamics*, 30(1):12–28, 2007. doi: https://doi.org/10.2514/1.22452.

[83] Ern J Lefferts, F Landis Markley, and Malcolm D Shuster. Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, control, and Dynamics*, 5(5):417–429, 1982. doi: https://doi.org/10.2514/3.56190.

[84] F Landis Markley. Attitude error representations for Kalman filtering. *Journal of guidance, control, and dynamics*, 26(2):311–317, 2003. doi: https://doi.org/10.2514/2.5048.

[85] Mak Tafazoli. A study of on-orbit spacecraft failures. *Acta Astronautica*, 64(2-3):195–205, 2009. doi: https://doi.org/10.1016/j.actaastro.2008.07.019.

[86] A.R.; Williams I.J DiStefano, J.J.; Stubberud. *Feedback and control systems, SCHAUM'S outlines*. McGraw-Hill, 2014.

[87] E. Mendelson F. Ayres. *Calculus, SCHAUM'S outlines*. McGraw-Hill, 6 edition, 2013.

[88] Alexandru-Razvan Luzi, Jean-Marc Biannic, Dimitri Peaucelle, and Jean Mignot. Time varying attitude control strategies for the myriade satellites. In *2012 IEEE International Conference on Control Applications*, pages 1328–1333. IEEE, 2012.

[89] F Gustafsson. *Adaptive filtering and change detection*. John Wiley & Sons, 2000.

[90] Martin L. Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 2008.

[91] Qiying Hu and Wuyi Yue. *Markov Decision Processes With Their Applications*. Springer Science & Business Media, 2008.

[92] CJCH Watkins and P Dayan. Q-learning. *Machine Learning*, 8(3):279–92, 1992. doi: https://doi.org/10.1007/BF00992698.

[93] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[94] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[95] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An Introduction (2nd edition 2018)*, volume 3. The MIT Press, 2018.

[96] Dimitri Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

[97] Dimitri P. Bertsekas. Feature-based aggregation and deep reinforcement learning: a survey and some new implementations. *IEEE/CAA Journal of Automatica Sinica*, 6(1):1–31, 2019. doi: https://doi.org/10.1109/JAS.2018.7511249.

[98] Brian Gaudet, Richard Linares, and Roberto Furfaro. Adaptive guidance and integrated navigation with reinforcement meta-learning. *Acta Astronautica*, 169:180–190, 2020. doi: https://doi.org/10.1016/j.actaastro.2020.01.007.

[99] Brian Gaudet, Richard Linares, and Roberto Furfaro. Terminal adaptive guidance via reinforcement meta-learning: Applications to autonomous asteroid close-proximity operations. *Acta Astronautica*, 171:1–13, 2020. doi: https://doi.org/10.1016/j.actaastro.2020.02.036.

[100] Brian Gaudet, Richard Linares, and Roberto Furfaro. Six degree-of-freedom body-fixed hovering over unmapped asteroids via lidar altimetry and reinforcement meta-learning. *Acta Astronautica*, 172:90–99, 2020. doi: https://doi.org/10.1016/j.actaastro.2020.03.026.

[101] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017. doi: https://doi.org/10.1109/MSP.2017.2743240.

[102] Brian Gaudet, Richard Linares, and Roberto Furfaro. Deep reinforcement learning for six degree-of-freedom planetary landing. *Advances in Space Research*, 65(7):1723–1741, 2020. doi: https://doi.org/10.1016/j.asr.2019.12.030.

[103] Giulia Ciabatti, Shreyansh Daftry, and Roberto Capobianco. Autonomous planetary landing via deep reinforcement learning and transfer learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2031–2038, 2021.

[104] Brian Gaudet and Roberto Furfaro. Adaptive pinpoint and fuel efficient mars landing using reinforcement learning. *IEEE/CAA Journal of Automatica Sinica*, 1(4):397–411, 2014. doi: https://doi.org/10.1109/JAS.2014.7004667.

[105] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992. doi: https://doi.org/10.1007/BF00992696.

[106] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[107] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. doi: https://doi.org/10.48550/arXiv.1707.06347.

[108] Jin Xu, Jinfeng Bu, Na Qin, and Deqing Huang. SCA-MADRL: Multiagent deep reinforcement learning framework based on state classification and assignment for intelligent shield attitude control. *Expert Systems with Applications*, 235:121258, 2024. doi: https://doi.org/10.1016/j.eswa.2023.121258.

[109] Ji Qi, Haibo Gao, Haitao Yu, Mingying Huo, Wenyu Feng, and Zongquan Deng. Integrated attitude and landing control for quadruped robots in asteroid landing mission scenarios using reinforcement learning. *Acta Astronautica*, 204:599–610, 2023. doi: https://doi.org/10.1016/j.actaastro.2022.11.028.

[110] Run-Ze Chen, Yuan-Xin Li, and Choon Ki Ahn. Reinforcement-learning-based fixed-time attitude consensus control for multiple spacecraft systems with model uncertainties. *Aerospace Science and Technology*, 132:108060, 2023. doi: https://doi.org/10.1016/j.ast.2022.108060.

[111] Zhong Shi, Fanyu Zhao, Xin Wang, and Zhonghe Jin. Satellite attitude tracking control of moving targets combining deep reinforcement learning and predefined-time stability considering energy optimization. *Advances in Space Research*, 69(5):2182–2196, 2022. doi: https://doi.org/10.1016/j.asr.2021.12.014.

[112] ZhiBin Zhang, XinHong Li, JiPing An, WanXin Man, and GuoHui Zhang. Model-free attitude control of spacecraft based on PID-guide TD3 algorithm. *International Journal of Aerospace Engineering*, 2020:1–13, 2020. doi: https://doi.org/10.1155/2020/8874619.

[113] Ignacio Carlucho, Mariano De Paula, and Gerardo G Acosta. An adaptive deep reinforcement learning approach for MIMO PID control of mobile robots. *ISA transactions*, 102:280–294, 2020. doi: https://doi.org/10.1016/j.isatra.2020.02.017.

[114] Xuan Wang, Jinwen Cai, Rui Wang, Gequn Shu, Hua Tian, Mingtao Wang, and Bowen Yan. Deep reinforcement learning-PID based supervisor control method for indirect-contact heat transfer processes in energy systems. *Engineering Applications of Artificial Intelligence*, 117:105551, 2023. doi: https://doi.org/10.1016/j.engappai.2022.105551.

[115] T Shuprajhaa, Shiva Kanth Sujit, and K Srinivasan. Reinforcement learning based adaptive PID controller design for control of linear/nonlinear unstable processes. *Applied Soft Computing*, 128:109450, 2022. doi: https://doi.org/10.1016/j.asoc.2022.109450.

[116] Hicham HENNA, Houari TOUBAKH, Mohamed Redouane KAFI, Ömer GÜRSOY, Moamar SAYED-MOUCHAWEH, and Mohamed DJEMAI. Satellite fault tolerant attitude control based on expert guided exploration of reinforcement learning agent. *Journal of Experimental & Theoretical Artificial Intelligence*, 2024. doi: https://doi.org/10.1080/0952813X.2024.2321152.

[117] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.

[118] Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1057–1063. The MIT Press, 1999.

[119] Jette Randløv and Preben Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *15th International Conference on Machine Learning(ICML)*, volume 98, pages 463–471, 1998.

[120] Burrhus Frederic Skinner. *The behavior of organisms: An experimental analysis*. BF Skinner Foundation, 2019.

[121] RL Atkinson, RC Atkinson, EE Smith, DJ Bem, and S Nolen-Hoeksema. *Hilgard's introduction to psychology*. Harcourt Brace College Publishers, 12 edition, 1996.

[122] John Eric Rayner Staddon. *Adaptive behavior and learning*. Cambridge University Press, 2016.

[123] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*, volume 99, pages 278–287. Citeseer, 1999.

[124] Aveen Dayal, Linga Reddy Cenkeramaddi, and Ajit Jha. Reward criteria impact on the performance of reinforcement learning agent for autonomous navigation. *Applied Soft Computing*, 126:109241, 2022. doi: https://doi.org/10.1016/j.asoc.2022.109241.

[125] Adam Laud and Gerald DeJong. The influence of reward on the speed of reinforcement learning: An analysis of shaping. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 440–447, 2003.

[126] Maksim Shirobokov, Sergey Trofimov, and Mikhail Ovchinnikov. Survey of machine learning techniques in spacecraft control design. *Acta Astronautica*, 186:87–97, 2021. doi: https://doi.org/10.1016/j.actaastro.2021.05.018.

[127] Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Reward function and initial values: Better choices for accelerated goal-directed reinforcement learning. In *International Conference on Artificial Neural Networks*, pages 840–849. Springer, 2006.

[128] Qinglei Hu and Bing Xiao. Fault-tolerant sliding mode attitude control for flexible spacecraft under loss of actuator effectiveness. *Nonlinear Dynamics*, 64:13–23, 2011. doi: https://doi.org/10.1007/s11071-010-9842-z.

[129] Enrico Canuto, Carlo Novara, Donato Carlucci, Carlos Perez Montenegro, and Luca Massotti. *Spacecraft dynamics and control: the embedded model control approach*. Butterworth-Heinemann, 2018.