Simulation and Implementation of Fault Tolerant Controllers for Attitude Control System of Spacecraft Test Bench

Sebastián León Serna

Degree work report as requirement to obtain the Title of Aerospace Engineer

External Advisor

Ph.D. Hever Moncayo

Embry-Riddle Aeronautical University, Daytona Beach, FL, EEUU

Internal Advisor

Ph.D. Felipe Andrés Obando Vega

Universidad de Antioquia, Medellin, Antioquia, Colombia

Universidad de Antioquia

Engineering Faculty

Aerospace Engineering

Carmen de Viboral, Antioquia, Colombia

2023

Dedication

For my parents Susana and Yovanny,

my younger brother Samuel,

my grandparents Sara, Nohelia,

Alejandro, and Pedro,

my love Valentina,

family, and friends,

who were and continue to be my main support throughout my career.

- S.L.S.

Acknowledgments

I am immensely grateful to the individuals whose unwavering support and guidance have been instrumental in the completion of this thesis. Their encouragement, expertise, and patience have left an indelible mark on my academic journey.

I am gratheful to Dr. Hever Moncayo, whose mentorship and dedication have been invaluable throughout my time in his laboratory. Under his supervision, I was fortunate to be exposed to cutting-edge research and gain hands-on experience in the field. He was my guide throught the development of this project, and trusted in me to work on his laboratory. I am indebted to him for the numerous discussions, feedback sessions, and opportunities that have significantly shaped my academic and personal growth.

To Professor Felipe Obando, who never hesitated to lend a helping hand whenever I faced challenges despite the geographical barrier. His timely advice, encouragement, and willingness to engage in extensive discussions over virtual meetings were crucial in overcoming various obstacles during this journey.

I cannot forget the support of my family, girlfriend, and friends. Their unwavering belief in my abilities, encouragement during difficult times, and understanding of my commitments were pivotal in reaching this milestone.

To ADCL members, for our fruitful discussions. To ASTRA Group, Delta V, and Voyager research hotbeds from Universidad de Antioquia, who contributed to this research in one way or another, as well as Embry-Riddle Aeronautical University. And finally and very importantly, to my alma mater Universidad de Antioquia.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS AND ABBREVIATIONS

ACS	Attitude Control System
ADCL	Advanced Dynamics and Control Laboratory
AIS	Artificial Immune System
ASTRA	Aerospace Science and Technology Research group
CG	Center of gravity
DCM	Direct Cosine Matrix
DOF	Degree Of Freedom
DRL	Deep Reinforcement Learning
EASY	Extreme Access System
ERAU	Embry-Riddle Aeronautical University
FIS	Fuzzy Inference System
FLC	Fuzzy Logic Controller
HIL	Hardware-in-the-Loop
HJB	Hamilton–Jacobi–Bellman
IGOR	Integrated Gravity Offloading Robotic System
IQR	Interquartile Range
IMU	Inertia Measurement Unit
LQR	Linear-Quadratic Regulator
LSL	Lower Saturation Limit
MEC	Mass-Expulsion Control
MF	Membership Function
ML	Machine Learning
MPC	Model Predictive Controller
MRAC	Model Reference Adaptive Controller
MRAIS	Model Reference Adaptive Immune System
N	Negative
NB	Negative Big
NLDI	Non-Linear Dynamic Inversion
NN	Neural Networks
NS	Negative Small
P	Positive
PB	Positive Big
PC-104	Onboard Computer PCM-3355 Reference
PID	Proportional-Integral-Derivative
PR	Pressure Regulator
PS	Positive Small
PS	Pressure Sensor
PWM	Pulse Width Modulated
Q	Solenoid Valve
R	Reservoir
RCS	Reaction Control System
RMS	Root Mean Square
RMSE	Root Mean Square Error
RV	Relief Valve
TVC	Thrust Vectoring Control
UdeA	Universidad de Antioquia
USL	Upper Saturation Limit
Z	Zero

SYMBOLS

Symbol	Description	Unit
$I_{3\times3}$	3×3 identity matrix	
R_{air}	Air gas constant	J/(kg K)
M_{air}	Air molar mass	kg/mol
γ_{air}	Air specific heat	
P_{tank}	Air tank pressure	Pa
T_{tank}	Air tank temperature	K
$tank_1$, $tank_2$	Air tanks pressure	psi
$\vec{H}$	Angular momentum	kg m^2/sec
C_Ω	Angular rate activity cut-off value	deg
$\tilde{e}_\Omega$	Angular rate activity metric	
$\vec{\tilde{\omega}}$	Angular rate bias vector	deg
ω_{bias}	Angular rate sensor bias	deg
$\vec{\omega}$	Angular rate vector	deg
B	Body reference frame	
ω_x, ω_y, ω_z	Components of $\vec{\omega}$	deg
n_1, n_2, n_3	Components of unit vector	
J_{cost}	Cost function	
S	Covariance	
J_{xy}, J_{xz}, J_{yz}	Cross-moments of inertia	kg m^2
m_{air}	Current air mass	kg
P'_{tank}	Current air tank pressure	Pa
L_x, L_y, L_z	Distances from CG	m
ϕ, θ, ψ	Euler angles (roll, pitch, yaw)	deg
$\vec{\Phi}$	Euler angles vector	deg
F_x, F_y, F_z	Force vector components	N
$\vec{F}_{Friction}$	Force vector from friction disturbances	N
$\vec{F}_{Gravity}$	Force vector from gravitational disturbances	N
$\vec{F}$	Forces vector	N
c_f	Friction coefficient	
C_{P_I}	Global performance cut-off value	
$\tilde{P}_I$	Global performance index	
I, I_q	Inertial reference frame	

m_0	Initial mass	kg
u	Input of a system	
$\Omega(\omega)$, $\Xi(q)$	Kinematic quaternion matrices	deg, -
CCC	Lin's coefficient of concordance	
m_{pt}	Mass change per thruster	kg/s
J	Matrix of products of inertia	kg m^2
μ	Mean	
R	Molar gas constant	J/(mol K)
M_x, M_y, M_z	Moment vector components	N m
$\vec{M}$	Moments vector	N m
K_Ω, K_q	NLDI pseudo gains	
A^*	Nozzle throat area	m^2
D^*	Nozzle throat diameter	m
$\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$	Orthonormal basis	
y	Output of a system	
ρ	Pearson's correlation coefficient	
P_I	Performance index	
τ	Period of oscillations	s
J_{xx}, J_{xx}, J_{xx}	Principal moments of inertia	kg m^2
q_0, q_1, q_2, q_3	Quaternion components	
δq	Quaternion error	
$C_{\Delta \boldsymbol{Q}}$	Quaternion tracking error activity cut-off value	
$\tilde{\delta \boldsymbol{q}}$	Quaternion tracking error activity metric	
$\vec{q}$	Quaternions	
α_r	Rotation angle for rotation quaternion	deg
$R^B_{I_q}$	Rotation matrix from I to B	
$\vec{q_R}$	Rotation quaternion	
$C_{\Delta S}$	Solenoid activity cut-off value	
$\tilde{s}$	Solenoid activity metric	
σ	Standard deviation	
x	States of a system	
T	Thruster	N
T_{fail}	Thruster failure numbers	
t_{bias}	Time of sensor bias	s
t_{fail}	Time of thruster failure	s

$\vec{n}$	Unit vector	
$\vec{q}_{R\psi}$, $\vec{q}_{R\theta}$, $\vec{q}_{R\phi}$	Vector components of $R_{I_q}^B$ respect to Euler angles	
v	Virtual control input	
V	Volume	m^3
Q, R	Weighting matrices	

ABSTRACT

The challenging and unpredictable conditions of space present various obstacles to spacecraft operations. To overcome these challenges, spacecraft require resilient systems that can withstand faults and failures. This is where the crucial role of fault-tolerant controllers comes into play, which are developed and implemented to mitigate failures and ensure the continued functionality of spacecraft. This thesis presents the design, validation, and comparison of a fault-tolerant controller applied into spacecraft attitude control system (ACS) by spacecraft test bench implementation. The design process of each controller starts with a literature review, which gives the basis of fault-controller methodologies, and also provides updated information of current used technologies for spacecraft ACSs and test benches. The validation and comparison processes are performed facing two architectures of NLDI controller with the proposed Fuzzy Logic Controller (FLC), in order to evaluate its performance and feasibility of its application in the aerospace industry. Simulation and implementation data are compared using multiples correlation metrics in order to assess the agreement between each set of data, getting a more accurate simulation model for spacecraft attitude controllers design before moving into implementation. Several failure scenarios are conducted to analyze the performance and robustness of each attitude controller. FLC performance shows to be on a par with NLDI controllers, being better in some cases for both nominal and failure scenarios. This demonstrates the capacities that a FLC can have despite that its nature is highly human logic dependent.

Keywords **— Fault-Tolerant Controller, Spacecraft Attitude Control System, Fuzzy Logic Controller, Spacecraft Test Bench**

RESUMEN

Las difíciles e impredecibles condiciones del espacio plantean diversos obstáculos a las operaciones de las naves espaciales. Para superarlos, las naves espaciales necesitan sistemas resistentes que puedan soportar fallos y averías. Aquí es donde entra en juego el papel crucial de los controladores tolerantes a fallos, que se desarrollan e implementan para mitigar los fallos y garantizar la funcionalidad continuada de las naves espaciales. Esta tesis presenta el diseño, validación y comparación de un controlador tolerante a fallos aplicado al sistema de control de actitud (ACS) de una nave espacial mediante su implementación en un banco de pruebas. El proceso de diseño de cada controlador comienza con una revisión de la literatura, que proporciona la base de las metodologías de control de fallos, y también proporciona información actualizada de las tecnologías utilizadas actualmente para ACSs de naves espaciales y bancos de pruebas. Los procesos de validación y comparación se realizan enfrentando dos arquitecturas de controlador NLDI con el controlador de lógica difusa (FLC) propuesto, con el fin de evaluar su rendimiento y la viabilidad de su aplicación en la industria aeroespacial. Los datos de simulación y de implementación se comparan mediante multiples métricas de correlación con el fin de evaluar la concordancia entre cada conjunto de datos, obteniendo un modelo de simulación más preciso para el diseño de controladores de actitud de naves espaciales antes de pasar a su implementación. Se llevan a cabo varios escenarios de fallo para analizar el rendimiento y la robustez de cada controlador de actitud. El rendimiento del FLC muestra estar a la par con los controladores NLDI, siendo mejor en algunos casos tanto para escenarios nominales como de fallo. Esto demuestra las capacidades que puede tener un FLC a pesar de que su naturaleza es altamente dependiente de la lógica humana.

Palabras clave **— Controlador Tolerante a la Falla, Control de Actitud de Vehiculos Espaciales, Controlador por Lógica Difusa, Banco de Pruebas de Vehiculos Espaciales**

I. INTRODUCTION

A. Motivation

Space exploration and satellite missions play a crucial role in advancing our understanding of the universe, supporting vital communication networks, and monitoring Earth's resources. However, the harsh and unpredictable space environment poses numerous challenges to spacecraft operations, demanding robust systems capable of withstanding faults and failures. One critical aspect of ensuring mission success is the development and implementation of fault-tolerant controllers that can mitigate failures to maintain spacecraft functionality [1].

Spacecraft operate in extreme conditions, exposed to radiation, temperature variations, micrometeoroids, and other environmental hazards. Despite meticulous engineering and testing, hardware and software failures can still occur during mission operations, potentially threatening the entire mission. Faults can manifest in various forms, ranging from sensor malfunctions and actuator failures to communication errors and power supply issues. The consequences of such failures can range from compromised scientific data acquisition to catastrophic mission losses [2], [3].

To mitigate the impact of failures, fault-tolerant control techniques have emerged as a promising approach to enhancing the reliability and robustness of spacecraft systems. Fault-tolerant controllers aim to detect, diagnose, and accommodate faults in real-time, allowing the spacecraft to continue operating and completing its mission objectives, even in the presence of failures [4], [5]. By dynamically adapting control strategies, fault-tolerant controllers enable spacecraft to maintain stability, performance, and functionality, thereby ensuring mission success [6].

The design and implementation of fault-tolerant controllers for spacecraft pose unique challenges due to the stringent resource constraints, limited computational capabilities, and stringent real-time requirements imposed by space missions [5]. The controllers must be capable of reconfigure control strategies, and seamlessly transitioning between redundant systems or backup components. Additionally, they must consider the intricate interdependencies between subsystems and prioritize fault-tolerance without compromising overall system per-

formance.

B. Problem Statement

This thesis aims to explore the design and application of fault-tolerant controllers in the context of spacecraft operations involving their attitude control system. It will investigate various fault-tolerant controller methods that can be integrated into the control systems of aerospace systems, specially spacecraft. The research will encompass both theoretical investigations and practical implementations, utilizing simulation models and experimental data to assess the performance and effectiveness of different fault-tolerant control approaches.

By examining the current state-of-the-art in fault-tolerant control techniques and their suitability for spacecraft applications, this project intends to contribute to the advancement of robust and reliable spacecraft systems. Ultimately, the findings and insights obtained from this thesis will help improve the fault-tolerant controllers implementation and enhance resilience of future spacecraft missions, ensuring their continued operation and enabling valuable scientific discoveries in the vastness of space.

C. Thesis Outline

This project begins with Chapter I, which describes the motivation that led to the development of this research work and the problem statement that will be worked throughout the project.

Then, Chapter II shows the general and specific objectives, giving a work route for this project, helping the reader to always have present the reasons why every chapter comes from the basics to specific developments, so the objectives can be accomplished. Also, with the objectives it is possible to outline the structure of the entire project.

Next, a review of some of the controllers used in spacecraft applications in Chapter III, giving a brief explanation of how they work and their benefits compared to other controllers. In addition, this chapter depicts newly techniques in spacecraft controls such as Machine Learning (ML) controllers, giving the reader with an update on research projects currently

underway.

Subsequently, Chapter IV provides an overview of the spacecraft dynamics and equations of motion that govern the attitude determination and control system. This chapter goes through the quaternions representation, showing the benefits of its use since it deals with the "Gimbal lock" phenomenon. Then, the quaternions and spacecraft kinematics are explained, so the equations described can be used for the development of the controllers.

As this thesis will incorporate either simulations and implementation of spacecraft attitude controllers, Chapters V and VI give the methodology used for these purposes. A simulation framework is developed for the characterization and tuning of the controllers to be deployed into a spacecraft test bench. Two simulation models are proposed, one with disturbances and the other without them, in order to simulate a more accurate model with respect to the test bench before deploying the algorithms into implementation. For implementation the Extreme Access System (EASY) spacecraft hardware framework is described, showing its components and functionality, also providing details about the software framework used for its operation, as it uses xPC Target, an useful tool for the development of applications in Simulink [7] that can be tested and run on a dedicated target computer or hardware. Additionally, the equations for the stability and performance metrics are described, so a comparison analysis between the controllers developed can be done.

For the main objective of this thesis, Chapter VII describes the controller architectures developed for simulations and implementation on the spacecraft test bench. This project focused on the development of a fault-tolerant controller, so Fuzzy-logic was chosen because it is easy to understand once spacecraft dynamics and kinematics are well understood, and because some challenges are currently being faced in its implementation due to computational limitations. Furthermore, the other nonlinear controllers previously developed are described, one of them with an Adaptive Immune System (AIS) algorithm. Thus, the Fuzzy-logic controller can be compared.

Then, the results are shown and discussed in Chapter VIII, displaying the tracking performance, stability, and performance metrics of the controllers. Also, the Bland-Altman plot is used to give a better understanding of the fitness of simulation data to real data.

Then, Chapter IX shows the conclusions of the previously presented results.

Finally, Chapter X indicate some recommendations about further work that can be done in order to improve the Fuzzy-logic controller and the EASY spacecraft test bench operation.

II. OBJECTIVES

A. General Objective

Design, validate and compare a fault-tolerant controller for spacecraft attitude control systems.

B. Specific Objectives

- Develop a dynamic model of the spacecraft used in the test bench.
- Design and simulate a fault-tolerant controller for spacecraft attitude control.
- Compare the performance of the designed controller with the performance of the controllers available on the test bench.
- Validate the dynamic performance of the designed controller on the spacecraft test bench.

III. LITERATURE REVIEW

In order to give an overview of the aspects related to the area of spacecraft attitude control, a literature review of systems, test bench methodologies approaches, and attitude control algorithms are described below.

A. Spacecraft Attitude Control Systems

The systems that generally make up a space vehicle are: structure, electrical power, communications, propulsion, on-board computer and attitude control system (ACS). The latter serves the function of controlling the orientation (also called attitude) of the vehicle in space in order to point the instruments and subsystems to a desired point, so certain scientific data can be acquired and specific missions can be accomplished.

The design of the attitude control system is generally made for three-axis stabilization, this means the spacecraft is capable of perform rotation around its own three axes. For this, several methodologies are implemented according to the mission requirements.

1) Magnetorquers

One of the attitude control methods is the use of Magnetorquers. A Magnetorquer, also known as a magnetic torquer, is a device designed to regulate attitude, mitigate tumbling, and provide stabilization. Its functionality relies on the interaction between a magnetically induced dipole and the magnetic field of the orbited celestial body. As a result of this interplay, a torque is produced, which is harnessed to manage the satellite's rotational movement around its center of gravity [8].

To build a magnetorquer, electromagnetic coils are employed. The application of an electric current to the coil leads to the generation of the magnetic dipole that governs the control mechanism. Figure 1 shows a set of magnetorquers used in a CubeSat attitude control system.

Some of the advantages in the use of magnetorquers are: the volume usage, as they are usually small; the power consumption and less probability to failure, as they only use electrical

Fig. 1. Magnetorquer for CubeSat. Source: [9].

power and no mechanical motion, compared to thrusters and reaction wheels, respectively; they produce torque without generating significant vibrations or disturbances, a characteristic that can be advantageous for sensitive payloads or instruments onboard the spacecraft that require a stable environment for optimal performance [10].

2) Reaction Wheels

Reaction wheels, also called momentum wheels, are spinning devices that exert torque on the spacecraft by either adding or extracting energy, inducing rotation along a specific axis. Sustaining the rotation of a flywheel, known as momentum, leads to the stabilization of the corresponding spacecraft axis. Incorporating multiple reaction/momentum wheels enables comprehensive three-axis attitude control and stability to be achieved [11].

Figure 2 shows a set of reaction wheel actuators used for CubeSat applications.

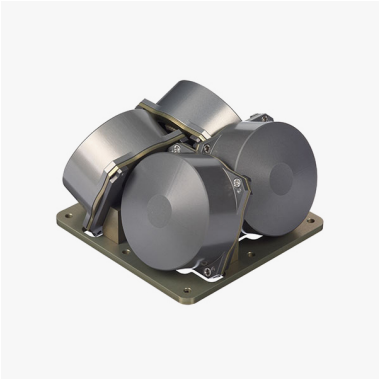


Fig. 2. Reaction wheels for CubeSat. Source: [12].

Reaction wheels have been widely employed for controlling the orientation of satellites, encompassing both large and small satellite missions. Almost all satellite missions that utilized reaction wheels as their primary attitude actuators have demonstrated exceptional performance in achieving precise attitude control, effectively meeting the requirements of their respective missions [13]. So far research on improving reaction wheels performance focus on power consumption and sizing optimization. Essentially, the power consumption of the reaction wheels can be reduced by strategically configuring their orientation within the satellite's onboard system, thus some research have focused on this topic [13]–[18].

3) Spherical Reaction Wheel

The reaction wheel system stands out among other systems for its superior capability to achieve precise spacecraft attitude control. However, a significant drawback of this system is its challenge in miniaturization, as mentioned before. As a result, utilizing the reaction wheel system for controlling small satellites is not straightforward, despite its impressive capabilities. To address this issue, an alternative approach involves the implementation of a spherical rotor in place of flywheels. This spherical reaction wheel system incorporates 3 stators that consist of piezoelectric actuators as main driving division [19]. Some research papers have worked on the development and improvement of these kind of systems, as shown in Figure 3.

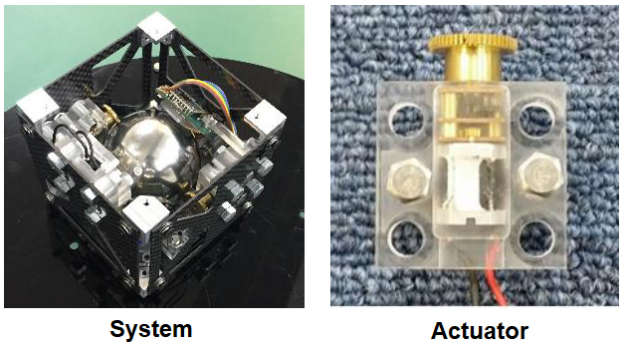


Fig. 3. Spherical Reaction Wheel. Source: [19].

4) Thrusters

Thrusters are one of the most commonly used systems to achieve spacecraft stabilization. They consist in a set of nozzles distributed along the spacecraft, which using a reservoir of fuel, create a reaction that accelerates certain amount of mass, creating a force that is converted into a moment of rotation for the spacecraft. For this reason, these type of devices are also referred to as mass-expulsion control (MEC) systems or reaction control systems (RCS) [20]. A RCS for attitude control of the Apollo Moon landing program is shown in Figure 4.

Fig. 4. RCS for Apollo Moon landing program. Source: [21].

Actuators like the ones mentioned earlier may produce minor reaction torques to counterbalance perturbation torques arising from errors and uncertainties in component alignment, resulting in undesired angular rotations. Nevertheless, the torques originating from these uncertainties during orbital maneuvers are considerably larger. Hence, a more robust actuator is required for attitude compensation, prompting certain research studies to propose the utilization of thrusters [22].

As this actuator is fuel dependent, its main problem is fuel consumption. For this reason, research works have focused on fuel consumption optimization without loosing its properties of forces and moments for attitude control of spacecraft [23], [24].

B. Spacecraft Attitude Control Test Bench

Developing a test bench for the verification of spacecraft attitude control systems is crucial for several reasons. Firstly, it allows for the assessment and validation of the performance and functionality of the control system in a controlled environment before actual deployment

in space. Secondly, it provides an opportunity to simulate and evaluate various operational scenarios and mission-specific conditions that the spacecraft may encounter during its mission. Moreover, a test bench facilitates the verification of the fault-tolerance and resilience of the attitude control system. By intentionally introducing faults or failures and observing the system's response, engineers can assess the system's ability to detect, isolate, and mitigate faults, ensuring the spacecraft's safe and stable operation even under adverse conditions.

For this reason, several spacecraft attitude control test benches have been developed. Some of them use a three degree of freedom (DOF) air bearing, which offer a nearly torque-free condition, nearly as close as possible to that of space. This kind of set up is usually used within a Helmholtz cage that generates a magnetic field, useful for orbit simulations around celestial bodies, and for the test of magnetorquers [25]. Also, these test benches are considered Hardware-in-the-Loop (HIL), as they incorporate multiple hardware coupled with software, such as sun sensors, sun simulators, stereo cameras, and magnetometers working through a software like MATLAB Simulink [25]–[27]. Figure 5 shows a CubeSat magnetic attitude control system test bench developed at the Politecnico di Torino, Italy.

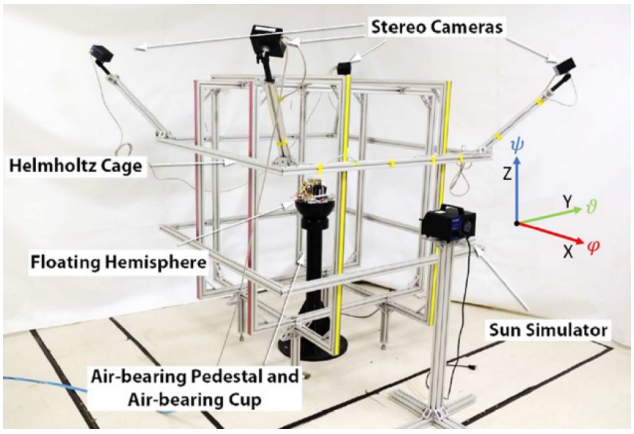


Fig. 5. CubeSat magnetic attitude control system test bench. Source: [26].

C. Spacecraft Attitude Controllers Review

Spacecraft attitude control is essential for maintaining the desired orientation and stability of spacecraft during space missions. Attitude control systems consist of sensors and actuators

to determine and change the orientation of spacecraft, respectively. Moreover, the control algorithms play a crucial role on the entire system, so precise pointing and mission success can be insured. For this reason, a brief description of some control architectures is given below.

1) States Feedback Control

In the control theory, the state of a dynamical system is defined by a collection of variables that permits prediction of the future development of a system. So it is possible to control a system through feedback of the state. The designing of a feedback control focuses on a main purpose: the positioning of closed loop eigenvalues in desired locations [28]. When the direct measurement of a system's state is not feasible, it is often possible to infer or estimate the state by utilizing our understanding of the system's dynamics and limited measurements. This is achieved by constructing an "observer" that leverages input and output measurements of a linear system, along with a model of the system's dynamics, to approximate the state.

The states of a system $\vec{x} \in \mathbb{R}^{\kappa}$ are related to the inputs $\vec{u} \in \mathbb{R}^{\iota}$, as well as the output $\vec{y} \in \mathbb{R}^{\upsilon}$, given the eq. (1).

$$\begin{aligned} \dot{\vec{x}} &= \boldsymbol{A}\vec{x} + \boldsymbol{B}\vec{u}, \\ \vec{y} &= \boldsymbol{C}\vec{x} + \boldsymbol{D}\vec{u} \end{aligned} \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{\kappa \times \kappa}$, $\boldsymbol{B} \in \mathbb{R}^{\kappa \times \iota}$, $\boldsymbol{C} \in \mathbb{R}^{\upsilon \times \kappa}$, and $\boldsymbol{D} \in \mathbb{R}^{\upsilon \times \iota}$ are called the state, input, output, and feedthrough matrices, respectively.

If the feedback is restricted to be a linear, the control law can be written as eq. (2).

$$\vec{u} = -\boldsymbol{K}\vec{x} + \boldsymbol{K_r}'\vec{r}\,, \tag{2}$$

where $\vec{r} \in \mathbb{R}^{\upsilon}$ is the reference value, $\boldsymbol{K} \in \mathbb{R}^{\kappa \times \kappa}$ is the feedback gain, and $\boldsymbol{K_r} \in \mathbb{R}^{\upsilon \times \upsilon}$ is the reference gain. Then, the closed loop system obtained when the feedback is applied to the

system, this is replacing eq. (2) into eq. (1), eq. (3) is obtained.

$$\dot{\vec{x}} = (\boldsymbol{A} - \boldsymbol{BK})\vec{x} + \boldsymbol{BK_r}\vec{r} \tag{3}$$

Now, the main purpose is trying to determine the feedback gain $\boldsymbol{K}$ so that the closed loop system has the characteristic polynomial in the form of eq. (4).

$$p(s) = s^n + p_1 s^{n-1} + ... + p_{n-1}s + p_n \tag{4}$$

This control problem is called the eigenvalue assignment problem or the pole placement problem, poles being the name of the values where the transfer function that describes the system becomes infinite [28], [29]. Figure 6 shows a block diagram of the states feedback controller architecture.

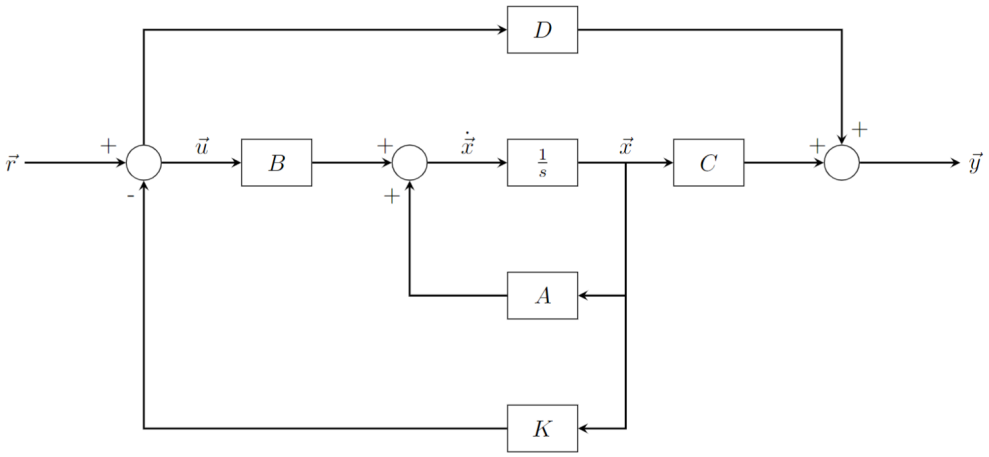


Fig. 6. States feedback controller block diagram.

Several research works have focused on the optimization or the integration of adaptive characteristics of these type of controller by looking for the best values that the poles can have, based on certain mission requirements [30]–[32].

2) Linear-quadratic Regulator (LQR)

As mentioned before, solving the dynamic programming problem for continuous systems is generally challenging, as there are many values that the poles can take. However, there are specific significant scenarios where solutions are more readily available. Many of these cases involve variations of linear dynamics and quadratic cost. The most straightforward instance, known as the linear quadratic regulator (LQR), involves stabilizing a time-invariant linear system to the origin. The LQR stands as a highly significant and influential outcome in the field of optimal control.

Considering a system in the state-space form shown in eq. (1), an infinite-horizon cost function can be described as in eq. (5).

$$J_{cost} = \int_0^\infty \left[\vec{x}^T \boldsymbol{Q} \vec{x} + \vec{u}^T \boldsymbol{R} \vec{u}\right] dt \ , \tag{5}$$

where $\boldsymbol{Q} = \boldsymbol{Q^T} \geq O \in \mathbb{R}^{\kappa \times \kappa}$ and $\boldsymbol{R} = \boldsymbol{R^T} > O \in \mathbb{R}^{\imath \times \imath}$ are named the weighting matrices (different from previous definition of $\boldsymbol{R}$). Then, the goal is to find the optimal cost-to-go function $J^*_{cost}(\vec{x})$ which satisfies the Hamilton–Jacobi–Bellman (HJB) equation shown by eq. (6), which by its definition, provides both a necessary and sufficient condition to determine the optimality of a control [33].

$$\forall \vec{x},\ 0 = \min_{\vec{u}} \left[\vec{x}^T \boldsymbol{Q} \vec{x} + \vec{u}^T \boldsymbol{R} \vec{u} + \frac{\partial J^*_{cost}}{\partial \vec{x}} (\boldsymbol{A}\vec{x} + \boldsymbol{B}\vec{u})\right] \ , \tag{6}$$

where the gradient function is defined by eq. (7), where $\boldsymbol{S} = \boldsymbol{S}^T \geq O \in \mathbb{R}^{\kappa}$ [34].

$$\begin{aligned} J^*_{cost}(\vec{x}) &= \vec{x}^T \boldsymbol{S} \vec{x} \\ \frac{\partial J^*_{cost}}{\partial \vec{x}} &= 2\vec{x}^T \boldsymbol{S} \end{aligned} \tag{7}$$

Some recent research works use LQR as point of comparison for new developed controllers, such as the work carried out by Guiggiani, Kolmanovsky, Patrinos, *et al.* [35], where

a Model Predictive Controller (MPC) is compared with LQR control, and is also used to emphasize some specific characteristics of the new controller benefits.

3) Nonlinear Dynamic Inversion (NLDI)

Spacecraft behavior is not always characterized by linearity. Nonlinear dynamics may manifest in certain flight regimes or failure scenarios. Consequently, to effectively control aircraft in such situations, a nonlinear controller becomes necessary. It is imperative for this controller to possess a high level of robustness, as the model never precisely matches the real-world dynamics.

The system model can be rewritten as in eq. (8).

$$\dot{\vec{x}} = f(\vec{x}) + g(\vec{x})\vec{u} \tag{8}$$

A virtual control input $\vec{v}$ can be defined as in eq. (9).

$$\vec{v} = b(\vec{x}) + a(\vec{x})\vec{u} \;\Leftrightarrow\; \vec{u} = a^{-1}(\vec{x})(\vec{v} - b(\vec{x})) \tag{9}$$

Now, the virtual control input $\vec{v}$ can be employed to control the entire system in a straightforward linear manner. The main hurdle lies in determining the appropriate setting for $\vec{v}$. For this, state feedback, as described before, is often used. So, the same procedure is carried out, as shown in eq. (10).

$$\vec{v} = -k_0\vec{x} - k_1\frac{d\vec{x}}{dt} - k_2\frac{d^2\vec{x}}{dt^2} - ... - k_{n-1}\frac{d^{n-1}\vec{x}}{dt^{n-1}} \tag{10}$$

Since $d^n\vec{x}/dt^n = \vec{v}$, this turns the system into a linear closed loop system governed by eq. (11).

$$\frac{d^n\vec{x}}{dt^n} + k_{n-1}\frac{d^{n-1}\vec{x}}{dt^{n-1}} + ... + k_1\frac{d\vec{x}}{dt} + k_0 = 0 \tag{11}$$

By appropriately selecting the parameter k_i, it becomes possible to define the properties of the closed-loop system. Utilizing eq. (9), the desired input $\vec{u}$ can then be determined. This process, referred to as the outer loop of NLDI, involves finding the value of $\vec{v}$. Subsequently, the inner loop involves obtaining the corresponding $\vec{u}$ value and incorporating it into the actual system [36].

Finally, the NLDI technique has proved to be an easy way of controlling nonlinear systems, so much so that adaptive characteristics incorporation to this controller is a recent investigation area [36], [37].

4) Model Reference Adaptive Controller (MRAC)

Adaptive control is a methodology employed to dynamically adjust the parameters of a system in real-time, ensuring a desired level of performance even when the system parameters are unknown or subject to changes. Model Reference Adaptive Control (MRAC) presents an approach to address challenges associated with adaptive control in practical settings. MRAC utilizes a closed-loop controller to compare the output of the system with a predefined reference response, enabling adjustments to the system's parameters in accordance with this reference response, which accommodates changes in various plant parameters [38]. Figure 7 describes the MRAC architecture in a block diagram.

Based on the architecture of MRAC, it can be classified into two types: first, the direct control, whereby the system adjusts itself to the error signal, described as the difference between the plant and the reference response. Here, the controller parameters are updated in real time by adaptive laws. Second, the indirect control, in which the system adjusts itself by comparing the plant output to online standard reference. In this case, the controller parameters are obtained by solving linear algebraic equations that correlate the controller parameters with online model of the plant for each time [38], [39].

Recent research works have purposed novel control algorithms by augmenting the NLDI controller with a direct MRAC, in order to improve the slow mode of the NLDI due to a performance degradation because of disturbances increases [40].

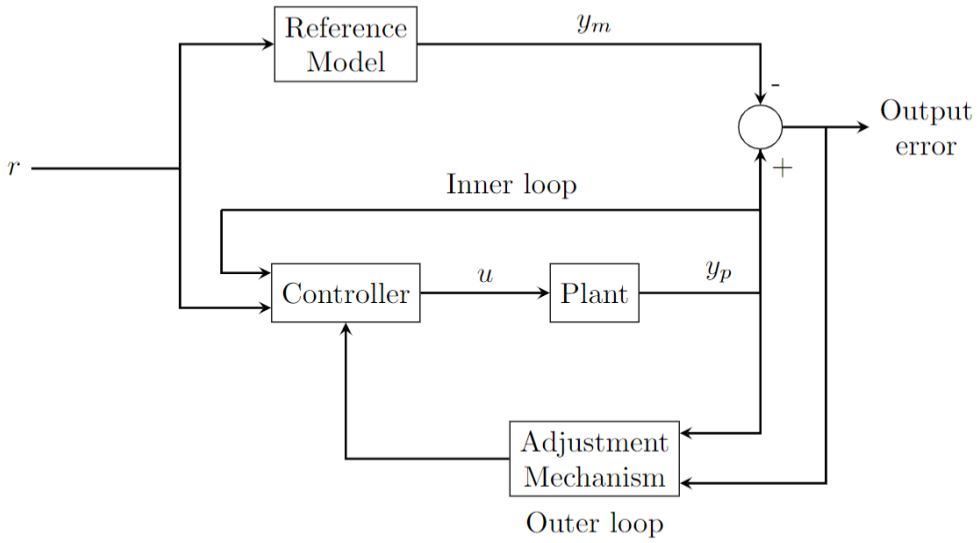


Fig. 7. MRAC architecture block diagram. Source: Adapted from [38].

5) Machine Learning (ML) Controllers

As stated in the LQR controller description, one of the challenges is finding the parameters that best fit to certain mission requirements. Therefore, within this optimal control problem, the HJB equation can transform into the Riccati Equation [41], [42]. Research endeavors have revealed that Neural Networks (NN) can be trained to serve as approximate solutions to this equation [43]. Here, the control objective becomes minimize a well-defined cost function J_{cost} within the space of possible control laws. An online learning loop provides experiential data to train the controller. Furthermore, the machine learning (ML) control can be a Deep Reinforcement Learning (DRL) architecture, which uses the HJB equation as a value function to be approximated by a set of critic-actor NN for approximation of the control policy [44]. Figure 8 illustrates a block diagram of a ML control architecture, where W_d refer to the NN weights.

6) Fuzzy Logic Controller (FLC)

Fuzzy logic controller (FLC) is an established control theory that simplifies the design process by utilizing the expertise of satellite designers, since its principle of functionality is based on a set of decision-based rules that analyze the system behavior and the linguistic input

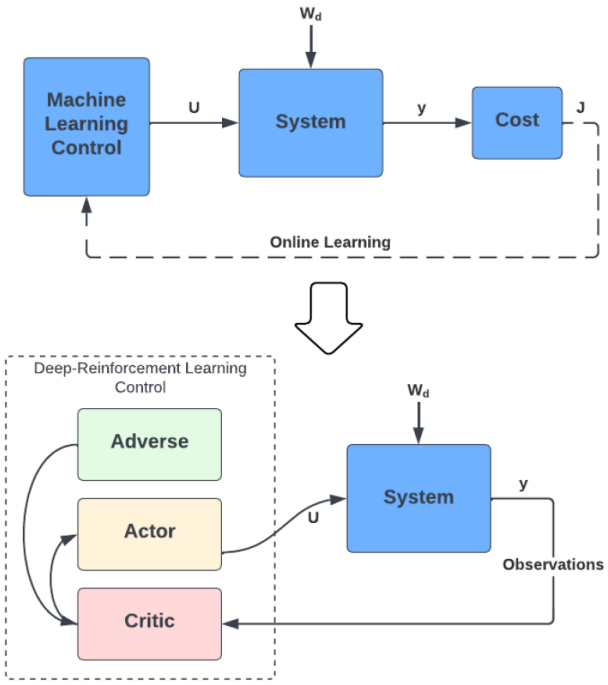


Fig. 8. Machine Learning Control architecture block diagram. Source: [43].

variables within the framework of the system [45]. FLC follows a three-step process consisting of fuzzification, inference (or decision-making), and defuzzification stages to generate its output. During the fuzzification stage, input variables are converted into linguistic variables using predefined membership functions (MFs). The fuzzified output is then determined based on the rules defined. In the defuzzification stage, the fuzzified output is transformed into the desired output used for system control. One notable advantage of FLC is that it does not require an exact system model during development. As a result, FLC finds extensive applications in industrial control, systems characterized by high uncertainty and nonlinearities [45]. For this reason, FLC may be classified as a nonlinear controller, whose parameters are determined online [46], [47].

There are two types of fuzzy inference systems (FIS), Mamdani and Sugeno. Mamdani uses a fuzzy set as the output of each rule, which is derived from an output MF and an implication method [48], [49]. These individual fuzzy sets from the output are merged into a unified fuzzy set using the aggregation method of the FIS. Finally, to obtain a precise output value, the combined output fuzzy set undergoes defuzzification employing a method, which

can be: centroid, bisector, middle of maximum, smallest of maximum, largest of maximum, or any other custom defuzzification method. Each of the previously mentioned methods is explained in *Defuzzification Methods* [50].

In the other hand, Sugeno FIS, also known as Takagi-Sugeno-Kang FIS, employs output membership functions that are either constant or linear functions based on the input values. In comparison to the Mamdani system, the defuzzification process for a Sugeno system is computationally efficient. This is because it calculates a weighted average or sum of a few data points instead of determining the centroid of a two-dimensional area, as mentioned before with the defuzzification methods for Mamdani [48], [51].

For this type of FIS in which the a MF has a single membership value is considered to be a type-1 FIS. Hence, while a type-1 membership function represents the membership degree within a specific linguistic set, it does not account for uncertainty in that degree. To address this uncertainty, interval type-2 membership functions can be employed. These functions allow for a range of values to express the degree of membership [52].

Table I describes each advantage in the use of each type of FIS.

TABLE I
ADVANTAGES OF EACH TYPE OF FLC. SOURCE: [48]

Mamdani	**Sugeno**
Intuitive	Computational efficient
Well-suited to human input	Work well with linear techniques, such as PID control
More interpretable rule base	Work well with optimization and adaptive techniques
Have widespread acceptance	Guarantee output surface continuity
	Well-suited to mathematical analysis

In the application of FLC, it is particularly suitable for nano and pico satellites with limited power budgets. By incorporating the designer's knowledge, fuzzy logic reduces the design complexity for attitude control. Notably, it requires significantly less power while meeting the attitude control requirements. Although fuzzy logic has not been widely adopted in space applications, it has shown promising results in specific space missions [45]. Addi-

tionally, from a computational point of view, fuzzy controllers cost much lower than LQR [53], [54].

Finally, as mentioned in Table I, one of the main advantages of Sugeno FIS is its capacity of being an adaptive controller. Therefore, several research papers have developed fault-tolerant controllers whose architectures rely on adaptive parameters coupled with Proportional-Integral-Derivative (PID) controllers [47], [54]–[56].

IV. SPACECRAFT DYNAMICS AND KINEMATICS

In order to apply control techniques in space vehicles, it is necessary to know the dynamics of a spacecraft, the systems that compose them, specifically the attitude control system, and its dynamics; what is automatic control, how it is applied in these vehicles and the types of controllers to be applied; and finally, the methodologies to evaluate the performance of these controllers. For that propose, this chapter provides a general description of the dynamics and kinematics governing a spacecraft. Also, the importance of working with quaternions representation is described, as well as how they provide a framework for the spacecraft's dynamics. Some of their properties are described in order to get the Direction Cosine Matrix (DCM) in quaternions representation. Finally, the equations governing the control architecture, getting the moments and forces, are described.

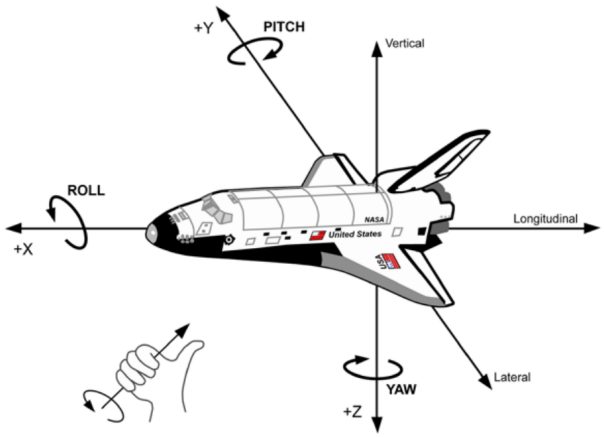


Fig. 9. Euler angles. Source: [57].

A. Quaternions Representation

As with aircraft, the orientation of a spacecraft can be defined by Euler angles, which are a sequence of angles, called pitch (θ), yaw (ψ), and roll (ϕ) (Figure 9), from an inertial reference frame [58]. However, the equations governing the Euler angles present a limitation for maneuvers requiring ± 90 degrees of rotation due to the phenomenon called "Gimbal lock". Therefore, the use of rotational *quaternions* is necessary for the representation of the

orientation [59].

A quaternion is a 4-component tuple, which defines an element in $\mathbb{R}^4$, as following:

$$q = (q_0, q_1, q_2, q_3),$$

where q_0, q_1, q_2 and q_3 are real or scalar numbers. An alternative representation is to define a scalar part (q_0) associated to a vector part ($\boldsymbol{q}$) in $\mathbb{R}^3$, like this:

$$\mathbf{q} = \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3,$$

where $\mathbf{i}, \mathbf{j}$ and $\mathbf{k}$ are the orthonormal basis in $\mathbb{R}^3$. Thus, a quaternion can be set as in eq. (12).

$$\vec{q} = q_0 + \mathbf{q} = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 \tag{12}$$

The rotation quaternion $\vec{q_R}$ can be defined as a rotation around a unit vector, as shown in eq. (13), where α_r is the rotation angle, and n_1, n_2 and n_3 are the components of the unit vector of the rotation axis $\vec{n}$.

$$\vec{q_R} = \begin{bmatrix} \cos\left(\frac{\alpha_r}{2}\right) \\ sin\left(\frac{\alpha_r}{2}\right) \cdot n_1 \\ sin\left(\frac{\alpha_r}{2}\right) \cdot n_2 \\ sin\left(\frac{\alpha_r}{2}\right) \cdot n_3 \end{bmatrix} \tag{13}$$

Taking into account a rotation order like $Z \rightarrow Y \rightarrow X$ the rotation matrix from body fixed reference frame B to inertial reference frame I_q using quaternions can be calculated using eq. (14).

$$R^B_{I_q} = \vec{q}_{R_\psi} \cdot \vec{q}_{R_\theta} \cdot \vec{q}_{R_\phi} \ , \tag{14}$$

where $\vec{q}_{R_\psi}$, $\vec{q}_{R_\theta}$, and $\vec{q}_{R_\phi}$ are the vector components of the rotational matrix with respect to the yaw, pitch and roll angles, respectively. Then, they can be defined as eq. (15).

$$\vec{q}_{R_\psi} = \begin{bmatrix} \cos\left(\frac{\psi}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{\psi}{2}\right) \end{bmatrix}, \ \vec{q}_{R_\theta} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ 0 \\ \sin\left(\frac{\theta}{2}\right) \\ 0 \end{bmatrix}, \ \vec{q}_{R_\psi} = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right) \\ 0 \\ 0 \end{bmatrix} \tag{15}$$

Then, by quaternions properties, the expression to determine the rotation quaternion in terms of the Euler angles is as shown in eq. (16). This expression is also called the Rotational Matrix from body fixed reference frame to inertial reference frame [60].

$$R^B_{I_q} = \vec{q} = \pm \begin{bmatrix} cos\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)cos\left(\frac{\psi}{2}\right) + sin\left(\frac{\phi}{2}\right)sin\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right) \\ sin\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)cos\left(\frac{\psi}{2}\right) - cos\left(\frac{\phi}{2}\right)sin\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right) \\ cos\left(\frac{\phi}{2}\right)sin\left(\frac{\theta}{2}\right)cos\left(\frac{\psi}{2}\right) + sin\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right) \\ cos\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right) - sin\left(\frac{\phi}{2}\right)sin\left(\frac{\theta}{2}\right)cos\left(\frac{\psi}{2}\right) \end{bmatrix} \tag{16}$$

The Rotational Matrix can be implemented to obtain the Direct Cosine Matrix (DCM), taking advantage of quaternions properties [61]. To do this, let's define the expression to find a point P_2 relative to point P_1 as quaternions, as shown in eq. (17).

$$P_2 = \vec{q} \cdot P_1 \cdot \vec{\bar{q}} \, , \tag{17}$$

where $\vec{\bar{q}}$ is the conjugate of $\vec{q}$.

Then, the Direction Cosine Matrix (DCM) can be determined in order to convert from an inertial reference frame to a body-fixed reference frame, in terms of quaternions or the Euler angles, as shown in eqs. (18) and (19), respectively.

$$DCM_q = \begin{bmatrix} 1-2q_3^2-2q_2^2 & 2(q_1q_2-q_3q_0) & 2(q_1q_3-q_2q_0) \\ 2(q_1q_2+q_3q_0) & 1-2q_3^2-2q_1^2 & 2(q_2q_3-q_1q_0) \\ 2(q_1q_3-q_2q_0) & 2(q_2q_3-q_1q_0) & 1-2q_2^2-2q_1^2 \end{bmatrix} \tag{18}$$

$$DCM^T = R_B^I = \begin{bmatrix} cos\,(\theta)\,cos(\psi) & sin\,(\phi)\sin\,(\theta)\cos\,(\psi) - cos\,(\phi)\,sin(\psi) & cos\,(\phi)\sin\,(\theta)\cos\,(\psi) + sin\,(\phi)\,sin(\psi) \\ sin\,(\psi)\,cos(\theta) & sin\,(\phi)\sin\,(\theta)\sin\,(\psi) + cos\,(\phi)\,cos(\psi) & cos\,(\psi)\sin\,(\theta)\sin\,(\psi) - sin\,(\phi)\,cos(\psi) \\ -sin(\theta) & sin(\phi)cos(\theta) & cos(\phi)cos(\theta) \end{bmatrix} \tag{19}$$

B. Quaternions Kinematics

As mentioned in Chapter IV Section A, quaternions representation is desired for spacecraft attitude description, as they avoid singularities. Furthermore, their differential equations are linear, property that makes computational process easier and improves accuracy in integration. For the following description, the spacecraft will be considered as a rigid body for simplicity. Thus, taking into account the quaterions description, the differential vector equation for quaternions is described as in eq. (20).

$$\dot{\vec{q}} = \frac{1}{2}\Omega(\omega)\vec{q}\,, \tag{20}$$

where $\dot{\vec{q}}$ is the derivate of quaternions vector, $\vec{q}$ is the quaternions vector, and $\Omega(\omega)$ is known as the kinematic quaternion matrix, and represents the skew symmetric matrix of the angular velocity vector, as shown in eq. (21).

$$\Omega(\omega) = \begin{bmatrix} -[\omega\times] & \omega \\ -\omega^T & 0 \end{bmatrix}\,, \tag{21}$$

where $\vec{\omega} = \vec{\omega}(t) = [\omega_x\ \omega_y\ \omega_z]^T$ is the angular time-dependent velocity vector, and $[\omega\times]$ refers to a cross-product matrix described by eq. (22).

$$[\omega\times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{22}$$

Then, expanding eq. (21) it takes the form shown in eq. (23).

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \tag{23}$$

Thus, eq. (20) becomes eq. (24).

$$\begin{bmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{24}$$

Eq. (20) can also be expressed as in eq. (25).

$$\dot{\vec{q}} = \frac{1}{2}\Xi(q)\vec{\omega} \ , \tag{25}$$

where $\Xi(q)$ is defined by eq. (26).

$$\Xi(q) = \begin{bmatrix} q_4 I_{3\times 3} + [q_{1:3}\times] \\ -q_{1:3}^T \end{bmatrix} \ , \tag{26}$$

where $I_{3\times 3}$ is a 3×3 identity matrix [62].

Finally, if $\omega(t)$ is known, the differential equations for the quaternions are linear [43].

C. Spacecraft Kinematics

The relationship between the angular momentum rate in the inertial reference frame and the moment summation can be summarized as the kinematic equation of motion for rigid bodies, as shown in eq. (27).

$$\sum \vec{M} = \frac{d\vec{H}^I}{dt} \ , \tag{27}$$

where $\vec{H}$ is the angular momentum, which can be described as in eq. (28)

$$\vec{H} = J \cdot \vec{\omega}_{B/I} \ , \tag{28}$$

where $\vec{\omega}_{B/I}$ is the angular velocity vector respect to the body reference frame obtained from the inertial reference frame (B/I), and J is the matrix of products of inertia of the rigid body. J is a real positive definite and symmetric matrix defined by eq. (29).

$$J = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{yz} & J_{zz} \end{bmatrix} \ , \tag{29}$$

where J_{xx}, J_{yy}, and J_{zz} are the principal moments of inertia, and J_{xy}, J_{xz}, and J_{yz} are the cross-moments of inertia.

In order to calculate the rate of angular momentum in the inertial reference frame based on eq. (27), it is essential to utilize the transport theorem in the way shown in eq. (30) [63].

$$\sum \vec{M} = \frac{d\vec{H}^I}{dt} = \frac{d\vec{H}^B}{dt} + \vec{\omega}_{B/I} \times \vec{H} \tag{30}$$

Using eq. (28), the derivative of the angular momentum vector $\vec{H}$ as seen by the body reference frame is as expressed by eq. (31).

$$\vec{M} = \frac{d\vec{H}^I}{dt} = J\dot{\vec{\omega}} + \vec{\omega} \times J\vec{\omega} \tag{31}$$

Then, the equation that describes the change of $\vec{\omega}$ along time can be defined as in eq. (32) [63].

$$\dot{\vec{\omega}} = J^{-1}[-\vec{\omega} \times (J\vec{\omega}) + \vec{M}] \tag{32}$$

Then, from eqs. (20) and (32), eq. (33) shows the system describing the dynamics and kinetics of a spacecraft when it is modeled as a rigid body.

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) = \begin{cases} \dot{\vec{q}} = \frac{1}{2}\Omega(\omega)\vec{q} \\ \dot{\vec{\omega}} = J^{-1}[-\vec{\omega} \times (J\vec{\omega}) + \vec{M}] \end{cases} \tag{33}$$

Here, the states of the system $\vec{x}$ are the four quaternions and the three angular rates. The instantaneous net torque $\vec{M}$ is determined by the contribution of the moments from the attitude control system, which are the control inputs $\vec{u}$. This leads to the fact that the derivative of the state vector $\dot{\vec{x}}$ is function of the states and the inputs of the system, $\vec{x}$ and $\vec{u}$, respectively.

V. SIMULATION FRAMEWORK

Simulating spacecraft attitude controllers before deploying them into an actual spacecraft is of paramount importance in the field of aerospace engineering. Spacecraft attitude control is critical for numerous spacecraft functions, including maintaining communication with Earth, pointing scientific instruments accurately, and stabilizing the spacecraft during maneuvers. Simulating spacecraft attitude controllers allows engineers to thoroughly test and validate the performance of the control algorithms and software in a controlled and cost-effective environment before implementation. The process of simulation is crucial to ensure the following:

1. **Safety:** Since space missions involve significant risks and expenses. By simulating various nominal and failure scenarios, engineers can improve the system's robustness and reliability, minimizing risks associated with deploying unproven control algorithms directly on a spacecraft.

2. **Cost-effectiveness:** Developing, building, and launching a spacecraft is an expensive endeavor. Simulating the attitude controller allows engineers to detect and address issues early in the development process, reducing the likelihood of expensive design changes or costly modifications during the testing phase.

3. **Realistic Representation:** Simulations provide a realistic representation of the spacecraft's dynamics, environmental conditions, and actuator inputs. By incorporating accurate models of the spacecraft's physical properties and the external forces acting upon it, engineers can evaluate the controller's performance under a wide range of operational scenarios.

4. **Iterative Design Process:** Simulations enable an iterative design process, allowing engineers to refine the attitude controller continuously. This iterative approach improves the overall effectiveness of the attitude control system and enhances its ability to meet mission requirements.

A simulation framework was developed in MATLAB Simulink to support the analysis of flight-operation scenarios along with data processing and evaluation of different guidance, navigation and control algorithms in normal and abnormal conditions.

The simulation framework provides a simple way to incorporate the algorithms needed for spacecraft implementation. A block diagram representation of its main components is illustrated in Figure 10. Here, a Spacecraft Plant consisting of a 6 degrees of freedom rigid body dynamics, which equations were explained in Chapter IV, gives the states of the spacecraft. These states are taken by the attitude controller and compares them with a commanded attitude. Then, having the actual and desired attitude, the controller computes the torques T_x, T_y, and T_z that the spacecraft attitude control system has to perform.

In order to translate the controller output torques into the actual torques that the spacecraft will experience, a Compressed Air Thrusters Model is developed, as shown in Figure 11. Since the spacecraft test bench consists of a thrusters ACS, this system is modeled. This model consists of a set of operations described as follow:

1. **Control Allocation:** Since the control architectures provide a desired control action to be performed by the vehicle, those values are translated into actual actuator control actions, depending on the hardware configuration, thus determining the force that each actuator has to execute.

2. **Failure Injection:** The force coming from the Control Allocation for each actuator is passed through a user-defined actuator failure injection filter and failure time. Here, the actuators to which the user wishes to inject a failure are imposed to be zero at a desired time.

3. **PWM Signal:** The force for each actuator is normalized taking into account a maximum force that the actuator can perform. Then, a Duty Cycle value is obtained according to the force magnitude. This Duty Cycle value is interpolated from a set of data obtained from a Compressed Air Thrusters test bench which was developed previously by Betancur Vesga [59]. After obtaining the Duty Cycle value for each actuator, the Pulse Width Modulation (PWM) signal can be generated.

4. **Thruster On/Off:** Finally, each PWM signal is sent to a set of solenoid valves that open and close the air flow at high pressure, thus generating a force. Each force is converted to torques by calculating the product between the force and the distance of the solenoid valve respect to the Center of Gravity of the spacecraft.

This Compressed Air Thrusters model in addition to enabling thruster failure injection, it also allows to calculate a fuel mass change based on the PWM signal, as will be described below.

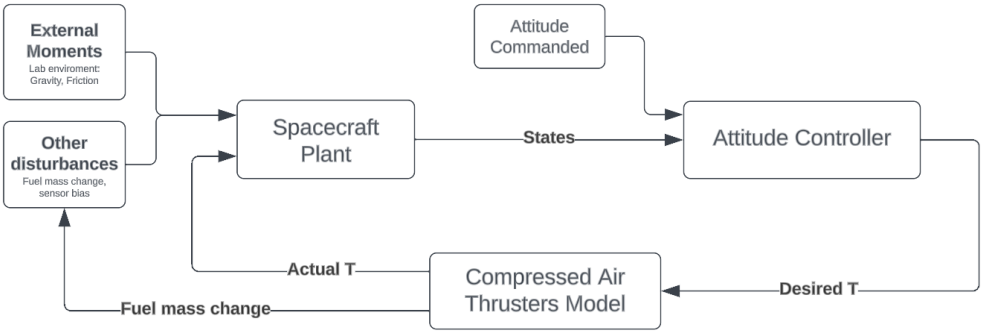


Fig. 10. Simulation framework.

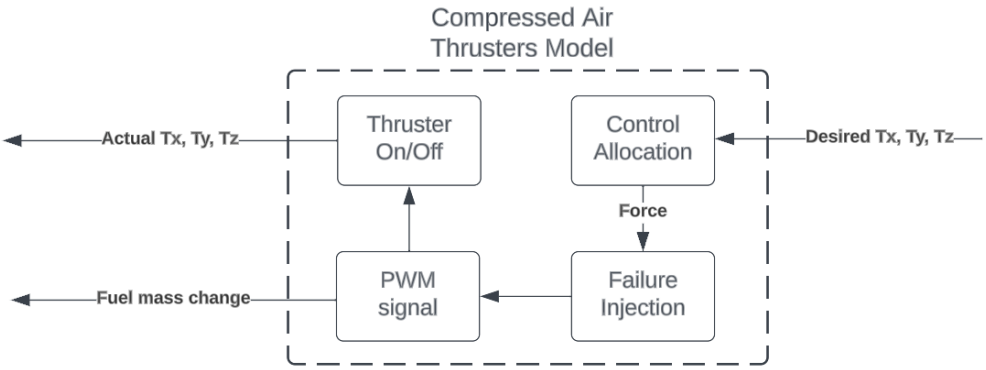


Fig. 11. Compressed Air Thrusters Model.

A. Disturbances Model

Considering disturbances in spacecraft attitude control system simulations is of utmost importance as the results may provide more accurate information about the actual phenomena.

Disturbances refer to external forces or factors that can perturb the spacecraft's orientation and stability, such as gravitational effects, atmospheric drag, solar radiation pressure, and thruster misalignments. These disturbances can significantly impact the performance and effectiveness of the attitude control system.

The scope of the disturbances modeling process for this project is to consider the disturbances due to the laboratory environment, such as moments due to gravity acting on spacecraft mass distribution, which also taking into account fuel mass change due to air lost caused by the control action; friction due to mechanical joints on the test bench, and sensor errors such as angular rates bias that can be caused as a consequence of sensor deterioration or environmental effects. This helps to achieve results similar to those that a spacecraft test bench might have.

1) Sensor Errors

Sensors play a vital role in measuring the spacecraft's attitude and providing feedback to the control system. However, sensors are prone to errors due to various factors such as noise, biases, calibration inaccuracies, temperature variations, and aging. By simulating sensor errors, the performance of the attitude control system under realistic conditions can be assessed, taking into account the limitations and inaccuracies associated with sensor measurements. This is why sensor errors are considered into the spacecraft simulation model.

For this work, gyroscopic sensor bias are implemented, thus generating angular rate steady state errors, modeled using eq. (34).

$$\vec{\tilde{\omega}} = \vec{\omega} + \vec{\omega}_{bias} \tag{34}$$

where $\vec{\omega}_{bias}$ is a user-defined angular rate bias vector and $\vec{\tilde{\omega}}$ is the final angular rates vector after sensor bias.

2) Mass Change Disturbances

In order to incorporate a spacecraft attitude control model with changes on its mass

distribution due to fuel mass change, the spacecraft kinematics need attention. From eq. (32), eq. (35) is presented, which incorporates the change rate of the matrix of products of inertia $\dot{J}$.

$$\vec{\omega}_{disturbances} = J^{-1}[-\vec{\omega} \times (J\vec{\omega}) - \dot{J}\vec{\omega} + \vec{M}] \tag{35}$$

Then, $\dot{J}$ can be calculated using eq. (36).

$$\dot{J} = \dot{m}\frac{J_{full} - J_{empty}}{m_{full} - m_{empty}} \ , \tag{36}$$

where J_{full} and J_{empty}, and m_{full} m_{empty} are the matrices of products of inertia and the total mass with the spacecraft full of fuel mass, respectively. And $\dot{m}$ is the fuel mass change rate, and can be calculated using eq. (37).

$$\dot{m} = -\dot{m}_{pt}\sum_{i=1}^{16} PWM_i \ , \tag{37}$$

where $\dot{m}_{pt}$ is the fuel mass change rate per thruster and PWM_i is the PWM signal state (1 or 0) of each thruster.

In order to enhance the thrusting performance of each solenoid valve, a 3D printed nozzle was design and putted at the soleoind valve exit. Then, the fuel mass change rate per thruster $\dot{m}_{pt}$ can obtained using the flow through a nozzle theory described by Sutton and Biblarz [64]. Thus, eq. (38) can be deduced.

$$\dot{m}_{pt} = \left(\frac{A^* \ P_{tank}}{\sqrt{T_{tank}}}\right)\left(\sqrt{\frac{\gamma_{air}}{R_{air}}}\right)\left(\frac{\gamma_{air}+1}{2}\right)^{-\frac{\gamma_{air}+1}{2(\gamma_{air}-1)}} \ , \tag{38}$$

where $A^* = \pi(D^*/2)^2$ is the nozzle throat cross area, with D^* as the nozzle throat diameter, P_{tank} and T_{tank} are the full pressure and temperature of the air tank, respectively, and γ_{air} and R_{air} are the specific heat and air gas constant, respectively.

Now, having $\dot{m}$ from eq. (37), the current mass can be obtained by a continuous-time integration using eq. (39). Here, initial fuel mass m_0, an Upper Saturation Limit (USL), and a Lower Saturation Limit (LSL) are considered.

$$m = \int \dot{m} \qquad with \quad \begin{cases} m_0 = m_{air} \\ USL = m_{full} \\ LSL = m_{full} - m_{fuel} \end{cases} \tag{39}$$

The initial fuel mass m_0 can be calculated taking into account the initial pressure of the air tank at the beginning of the flight, using the ideal gas equation as shown in eq. (40).

$$m_{air} = 2M_{air}\frac{P'_{tank}V}{RT_{tank}} \ , \tag{40}$$

where M_{air} is the air molar mass, P'_{tank} is the current pressure of the air tank, V is the volume of the air tank, and R is the molar gas constant. The coefficient 2 is to get the total air mass, as there are two air tanks.

Then, having the current mass from eq. (39), the current matrix of products of inertia can be obtained by a matrix interpolation [65].

Finally, after having the current and change rate of the matrix of products of inertia, J and $\dot{J}$, $\vec{\omega}$ can be solved.

3) Gravitational Disturbances

The disturbance of gravity acting on spacecraft mass distribution is computed using eq. (41). This equation is derived by assuming that the spacecraft behaves like a pendulum when it is stabilized.

$$\vec{F}_{Gravity} = -\left(\frac{2\pi}{\tau}\right)^2 J \ sin(\vec{\Phi}) \ , \tag{41}$$

where $\vec{F}_{Gravity}$ is the vector of forces due to gravity, τ is the period of oscillations, J is the matrix of moments of inertia, and $\vec{\Phi}$ is the Euler angles vector.

4) Friction Disturbances

The friction due to mechanical joints in the test bench is also considered, as described by eq. (42), where c_f is the friction coefficient and $\vec{\omega}$ is the angular rates vector.

$$\vec{F}_{Friction} = -c_f \ \vec{\omega} \tag{42}$$

5) Center of Gravity Shifting

Since $\vec{F}_{Gravitational}$ and $\vec{F}_{Friction}$ are forces, they act on the CG of the spacecraft. As mass change is being considered, the CG changes its position along every flight. Hence, several measurements from the spacecraft test bench were performed to adjust the simulation dynamics, thus obtaining the frequency and amplitude of the oscillations without control action. A CG shifting coefficient equation was obtained from the measurements, as shown in Figure 12.

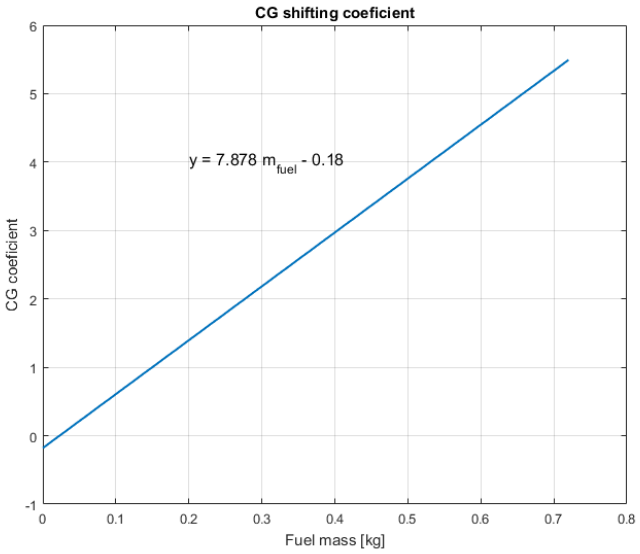


Fig. 12. CG Shifting Coefficient.

The final disturbances model considering mass change, gravitational, and friction disturbances, is shown in Figure 13.

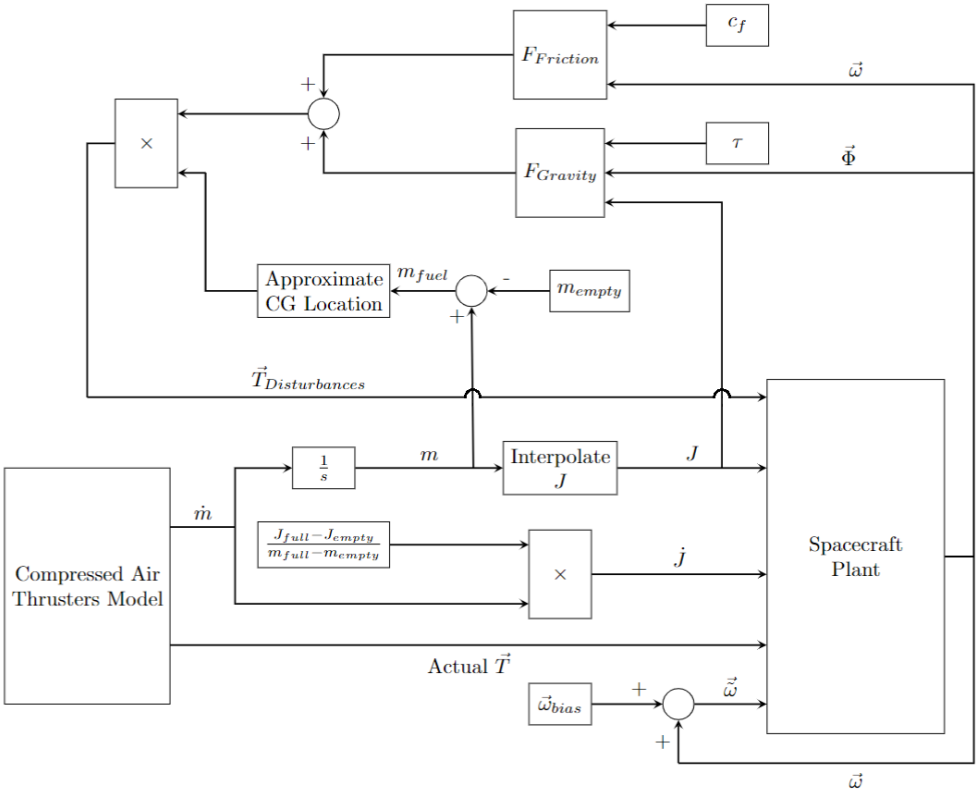


Fig. 13. Disturbances Model Block Diagram.

6) Disturbances Values

As mentioned previously, in order to fine-tune the simulation dynamics, various measurements were carried out on the spacecraft test bench. Thus, Table II describes the values used for the disturbances described above. It is important to note that period of oscillations and coefficient of friction were neglected for yaw axis due to the mechanical design of the test bench, as will be explained in Chapter VI.

To obtain the matrices of products of inertia J_{full} and J_{empty}, Autodesk Inventor 3D CAD software [66] was used to model the spacecraft test bench, taking into account the mass and position of each hardware element, obtaining the matrices shown in eq. (43).

TABLE II
VALUES USED FOR DISTURBANCES MODEL.

Variable	**Description**	**Value**
	Gravitational Disturbances	
τ	Period of oscillations (roll, pitch, yaw)	6.54, 7, N/A s
	Friction Disturbances	
c_f	Coefficient of friction (roll, pitch, yaw)	0.02, 0.05, N/A
	Mass Change Disturbances	
D^*	Nozzle throat diameter	0.00414 m
A^*	Nozzle throat area	$1.346\times10^{-}5$ m^2
P_{tank}	Tank pressure	997671.38 Pa
T_{tank}	Tank temperature	297 K
γ_{air}	Air specific heat	1.4
R_{air}	Air gas constant	287 J/(kgK)
m_{pt}	Mass change per thruster	0.0315 kg/s
m_{full}	Spacecraft mass with filled air tanks	17.17 kg
m_{fuel}	Full air tanks mass	0.72 kg
m_{empty}	Spacecraft mass with empty air tanks	16.45 kg

N/A: Not Applicable.

$$J_{full} = \begin{bmatrix} 0.405 & 2.679 \times 10^{-4} & 0.006 \\ 2.679 \times 10^{-4} & 0.410 & 1.579 \times 10^{-4} \\ 0.006 & 1.579 \times 10^{-4} & 0.558 \end{bmatrix}, \; J_{empty} = \begin{bmatrix} 0.450 & -0.027 & 0.022 \\ -0.027 & 0.482 & 0.013 \\ 0.022 & 0.013 & 0.624 \end{bmatrix} \quad (43)$$

B. Verification of Simulation Models

In order to verify the simulation models developed, multiple tests have to be performed. For this purpose, implementation data, as will be described in Chapter VI, is obtained to

generate the "actual" data sets. Then, simulations with disturbed and undisturbed models are run doing the same maneuver as in implementation tests. After doing this process, there will be three data sets: actual, disturbed model, and undisturbed model. Then, a correlation analysis and a RMSE of pairs actual-simulation models can be performed.

Correlation analysis refers to the examination of the association or connection between two or more quantitative variables. This analysis is primarily based on the assumption of a linear relationship between these variables. Like measures of association for binary variables, correlation evaluates the "strength" or "degree" of the association between the variables, as well as its direction.

The outcome of a correlation analysis is represented by a correlation coefficient, which can range from -1 to +1, indicating a perfect positive or negative linear relationship between the variables, respectively. On the other hand, a correlation coefficient of zero implies that there is no linear relationship between the two variables under investigation [67].

For this analysis, two correlation coefficients are considered, the Pearson correlation coefficient (ρ) and Lin's coefficient of concordance (CCC).

1) Pearson Correlation Coefficient

The bivariate Pearson Correlation calculates a sample correlation coefficient, which assesses the intensity and direction of linear associations between sets of continuous variables. In broader terms, it investigates whether there is statistical indication for a linear relationship among these variable pairs within the population, represented by the population correlation coefficient, described by eq. (44) [68], [69].

$$\rho(A,B) = \frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{A_i-\mu_A}{\sigma_A}\right)\left(\frac{B_i-\mu_B}{\sigma_B}\right) \ , \tag{44}$$

where A and B are the two data sets, N is the number of data points, μ_A and σ_A are the mean and standard deviation of A, respectively, and μ_B and σ_B are the mean and standard deviation of B.

2) Lin's Coefficient of Concordance

Lin's concordance correlation coefficient (CCC) evaluates the concordance between a fresh test (Y) and a reference test (X). This metric quantifies the level of agreement between these two measurements of the same variable. Similar to correlation, the CCC scale extends from -1 to 1, indicating perfect agreement when the value is 1. However, it is essential to note that CCC cannot surpass the absolute value of Pearson's correlation coefficient ρ. Lin's concordance correlation coefficient can be calculated using eq. (45) [70], [71].

$$CCC = \frac{2\ S_{XY}}{(\mu_X - \mu_Y)^2 + S_X^2 + S_Y^2}\ , \tag{45}$$

where S_{XY} is the covariance between X and Y, μ_X and μ_Y are the mean of X and Y, respectively, and S_X and S_Y are the variance of X and Y, respectively.

3) Data Distributions for p-value

It is important to note that a correlation coefficient should be interpreted together with a p-value, which helps to assess whether a correlation is real (statistically significant when p-value < 0.05). In order to do so, the data distribution of each data set has to be determined, so the p-value can be calculated.

To find the data distribution of each data set, they have to be compared and fitted with probability data distributions. An algorithm developed by Lorenz [72] in MATLAB tries to fit all valid parametric probability distributions to a data set and provides the best fitting probability data distribution with its parameters.

After having the data distributions, the p-value can be computed using several tests. For instance, if the data follows a normal distribution, a z-test or a t-test can be used to calculate the p-value. Or if the data follows a t-distribution, the t-test can be used as well [73].

VI. SPACECRAFT TEST BENCH IMPLEMENTATION

This chapter presents a test bench for relevant and successful implementation of the quaternion based controllers that will be developed and described in Chapter VII. The controllers were implemented on a concept spacecraft test bench named Extreme Access System (EASY), designed and built at the Advanced Dynamics and Control Laboratory of Embry-Riddle Aeronautical University (ADCL), FL (see Figure 14). The concept spacecraft aims to support the development of novel autonomous prospector space exploration spacecraft for in situ resource utilization in environments such as asteroids, where gravitational force is minimal. One of the major goals of this chapter is to describe a real way to test the algorithms developed in Chapter VII. The subsequent results intend to evaluate the performance of the developed control laws for missions in which the extreme environment might put the whole system at risk.

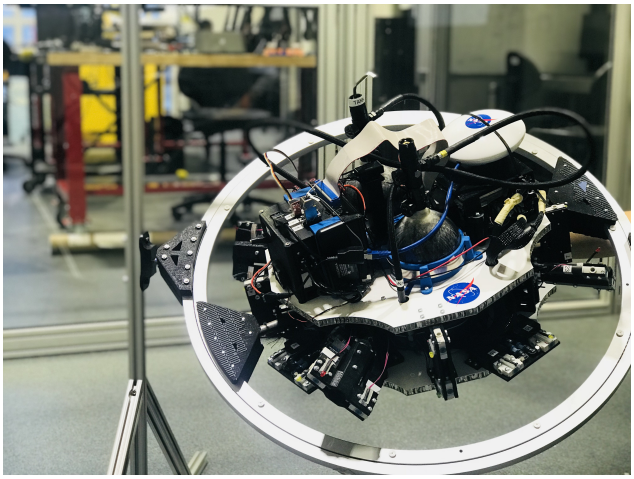


Fig. 14. Extreme Access System (EASY) Spacecraft Test Bench.

EASY spacecraft is mounted on a three degree of freedom gimbaled platform that allows free motion in roll, pitch, and yaw axes. The main purpose of this setup is to demonstrate full attitude control and angular rate regulation in gravity-less environments while tracking pre-defined attitude trajectories, and recovering from tumbles or other kind of abnormal conditions that might occur in space.

A. Spacecraft Hardware Description

EASY counts with 24 solenoid-valves acting as thrusters. This propulsion system consists of sixteen fixed thrusters grouped in pairs, parallel (horizontal configuration) and orthonormally (vertical configuration) located to the reference plane xy, allowing yaw and pitch rotations, respectively; and eight Thrust Vectoring Control (TVC) placed in pairs, distributed around the z axis, connected to a servo motor mechanism. This configuration allows each pair of solenoids to have a rotation around its position to achieve a yaw rotation. Figure 15 shows the location of each thruster around the x and y axes of EASY.

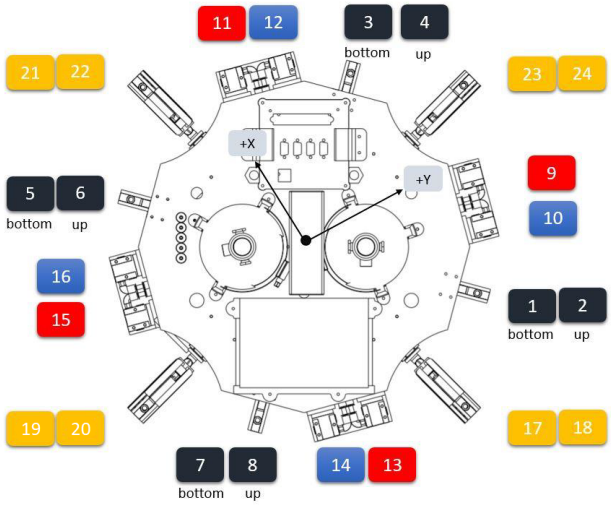


Fig. 15. Location of the thrusters in the x and y axes. Yellow for Thrust Vectoring Control; red and blue for horizontal configuration, left and right side respectively; dark blue for vertical configuration, denoting both bottom or up.

EASY spacecraft is mounted into an active system to allow gravity offloading, simulating partial gravity environments. This active system is called IGOR (Integrated Gravity Off-Loading System). IGOR is capable of simulate reduced gravity by integrating a single-point joint to suspend the spacecraft through a load cable and a pair of linear drive tracking stages in the horizontal plane. Since the friction in the horizontal movement allows only operation by humans, this system gives a vertical partial gravity motion. Figure 16 shows IGOR system attached to EASY spacecraft.

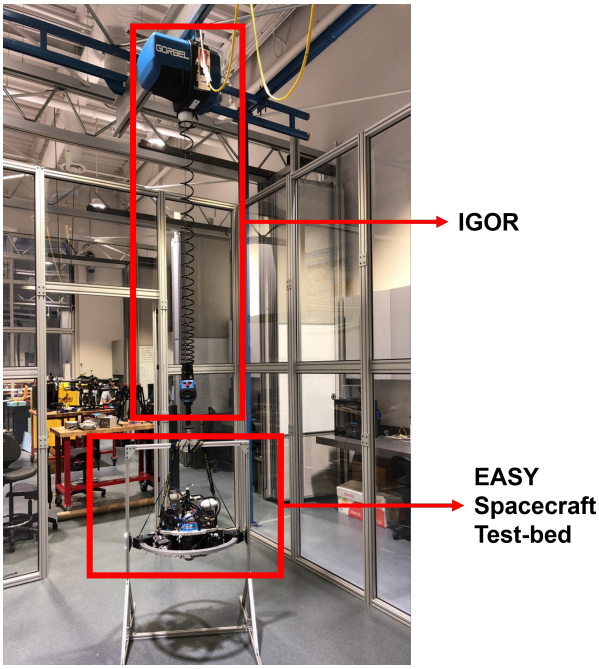


Fig. 16. Integrated Gravity Off-Loading System with EASY.

The propulsion system (see Figure 17) uses compressed air stored in a pair of reservoirs (R1 and R2) at high pressure, which can reach up to 4500 psi of pressure. Four regulators (PR1 to PR4) are used to drop the pressure from the reservoirs to 130 psi, which is the desired operating pressure of the thrusters. Additionally, four sensors (PS1 to PS4) are placed along the system to monitor the pressure levels. For safety purposes, four relief valves (RV1 to RV4) are used before each pressure regulator in case the system suffers over-pressure. Solenoid valves (Q1 to Q24) are placed at the end of the system with their corresponding nozzles. Each nozzle has been designed to produce at least 1 Newton of thrust.

The opening and closing of the solenoids are regulated by means of Pulse Width Modulated (PWM) signals from the digital IO pins on the onboard computer. The control architectures that will be described in Chapter VII were first tested in simulation to corroborate proper operation, as described in Chapter V. After confirming that the control laws work properly in simulation, they are deployed into the target flight computer through Simulink-Real time environment, which is an efficient code generation and prototyping tool from MathWorks software. It allows the development of applications directly from Simulink

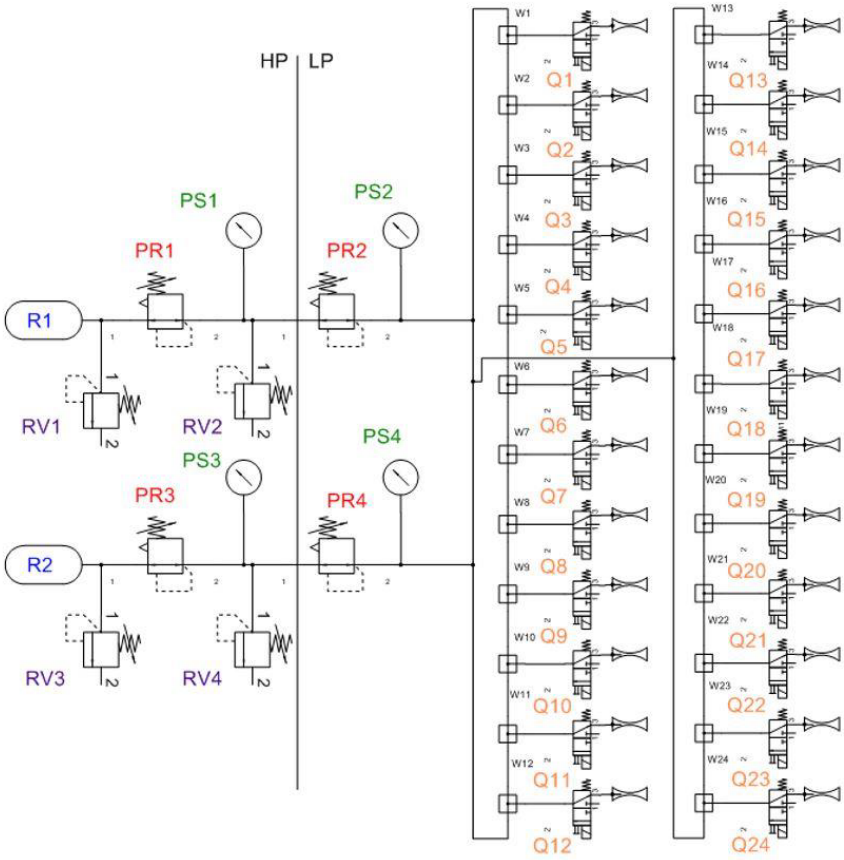


Fig. 17. Pneumatic system for propulsion. R_i are reservoirs, PR_i are pressure regulators, PS_i are sensors, RV_i are relief valves, and Q_i are solenoid valves.

so that they can be tested and run on a dedicated target computer or hardware [74].

The control algorithm code is generated automatically from Simulink and is compiled into the target machine using a real-time kernel. The host computer has the capability of downloading the code through a TCP/IP protocol. The flight computer selected for the target is the PC-104 PCM-3355 (see Figure 18), integrated with the serial module Emerald MM-4M-Port and the Onyx MM Digital I/O module (see Figures 19 and 20). This enables EASY spacecraft to include serial and analog input modules, along with digital I/O. The digital control signals are used to actuate each of the solenoid valves that regulate the proper amount of air that each thruster requires for attitude control.

The Inertial Measurement Unit (IMU) used in EASY is 3DM-GX3-45 by Microstrain

Fig. 18. PC-104 PCM-3355 Main computer.

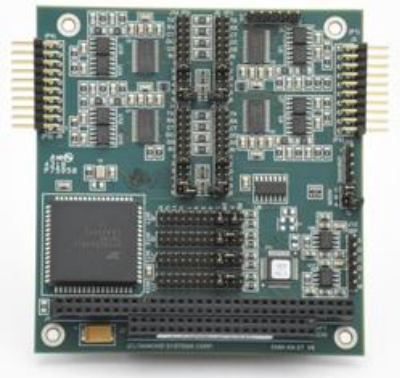


Fig. 19. Emerald MM-4M-Port Serial module.

Fig. 20. Onyx MM Digital I/O module.

[75] (see Figure 21), which is capable of providing accurate measurements of attitude and angular rates (signals required for the controllers). The IMU outputs can be read by one of the serial ports of the flight computer by means of a RS232 communication protocol. Table III describes some of the main characteristics of the sensors within the Microstrain.

TABLE III

CHARACTERISTICS OF MICROSTRAIN IMU. SOURCE: [75]

	Accelerometers	**Gyroscopes**	**Magnetometers**
Initial bias	$\pm 0.002\ g$	$\pm 0.25°/sec$	$\pm 0.003\ G$
Noise density	$\pm 80\ \mu g/\sqrt{Hz}$	$0.03°/sec/\sqrt{Hz}$	$100\ \mu G/\sqrt{Hz}$
Alignment error	$\pm 0.05°$	$\pm 0.05°$	$\pm 0.05°$
Sampling rate	$30\ kHz$	$30\ kHz$	$7.5\ kHz\ max$

Fig. 21. 3DM-GX3-45 IMU. Source: [75]

The sensors, actuators, flight computer and additional hardware were tested sepa-

rately to corroborate full functionality before they were mounted and fully incorporated into the EASY. The diagram in Figure 22 describes the main test framework. The prototype is connected via Wi-Fi to the host computer by a high data rate transmission 5.0 GHz connection, achieved through Simulink xPC Target linked with a Ubiquity Rocket M5 transmission station that is onboard EASY. The data transmission rate is crucial for online tuning of the controllers and signal monitoring. This architecture also aids test repeatability and the injection of failure scenarios.

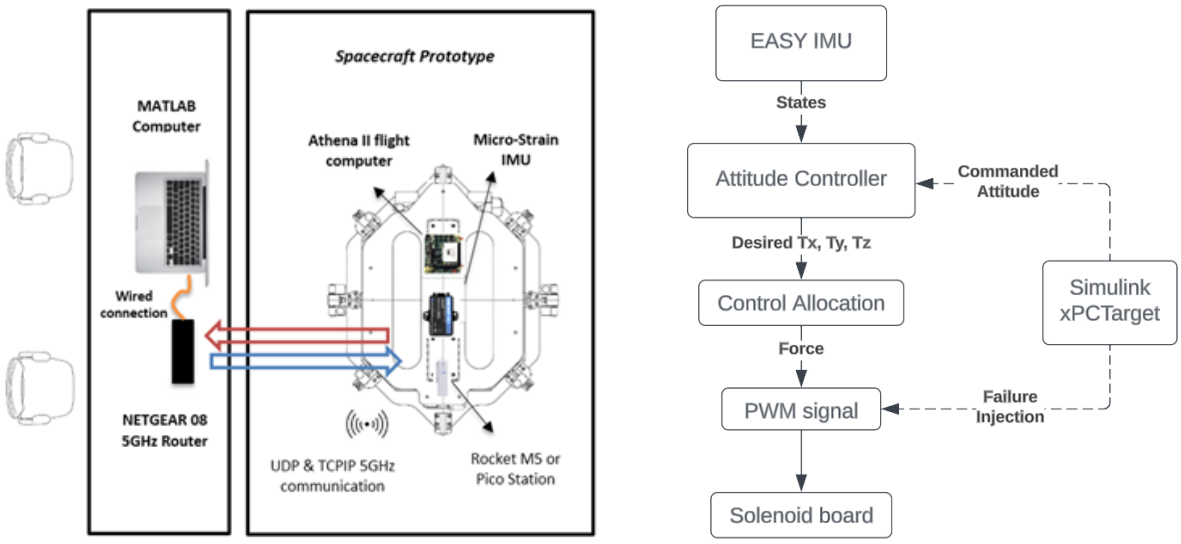


Fig. 22. Schematic and Block diagram of Test Bench and Hardware framework used on EASY.

B. Control Allocation

It is important to note that the control architectures provide a desired control action to be performed by the vehicle. Therefore, those values have to be translated into actual actuator control actions, depending on the hardware configuration. For this purpose, a control allocation process is defined to determine the output of each actuator.

As previously mentioned, there are a total of 24 thrusters in EASY spacecraft, each of them is capable of supplying a maximum of 1 Newton of force. In order to implement the attitude controllers, the first step is to develop a set of equations that describe the total forces and moments that act on EASY spacecraft due to its specific actuator configurations.

The actuator configurations can be defined as vertical, horizontal, and TVC, as described in Chapter VI Section A. With the vertical propulsion system configuration, pitching and rolling maneuvers are generated along the y and x axes, respectively (see Figures 23 and 24), while yawing maneuvers are generated with the horizontal configuration along the z axis (see Figure 25).

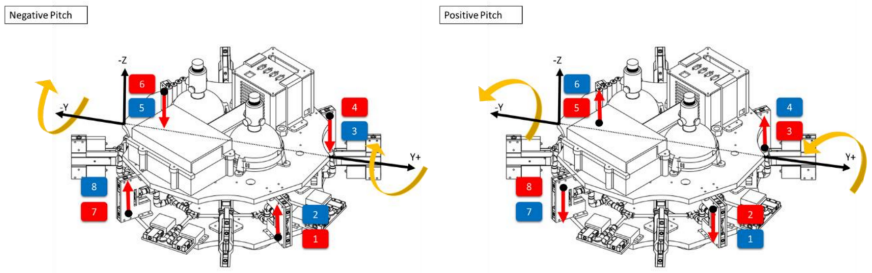


Fig. 23. Pitching maneuver. Negative with thrusters number 1, 4, 6, and 7. Positive with thrusters number 2, 3, 5, and 8.

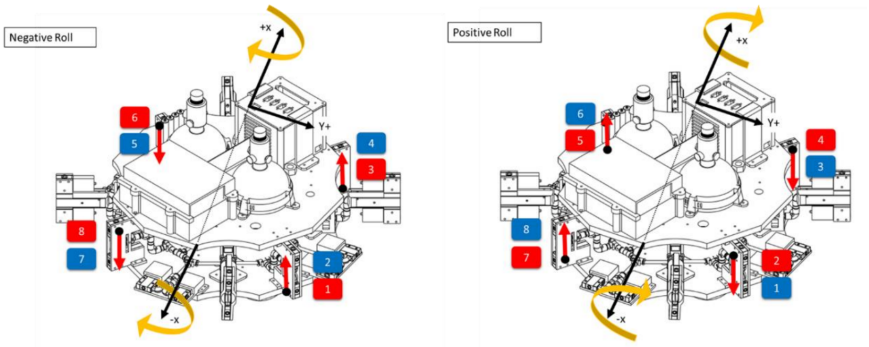


Fig. 24. Rolling maneuver. Negative with thrusters number 1, 3, 6, and 8. Positive with thrusters number 2, 4, 5, and 7.

Similarly, EASY spacecraft is able to perform pitching and rolling maneuvers with the TVC configuration when the four pair of thrusters are in vertical direction (see Figures 26 and 27), and yawing maneuvers when the TVC servomotors are rotated 90 degrees (see Figure 28).

Assuming an offset in the yaw's thrusters location of 15 degrees respect to z axis, eqs. (46) and (47) define the forces and moments with respect to the Center of Gravity (CG)

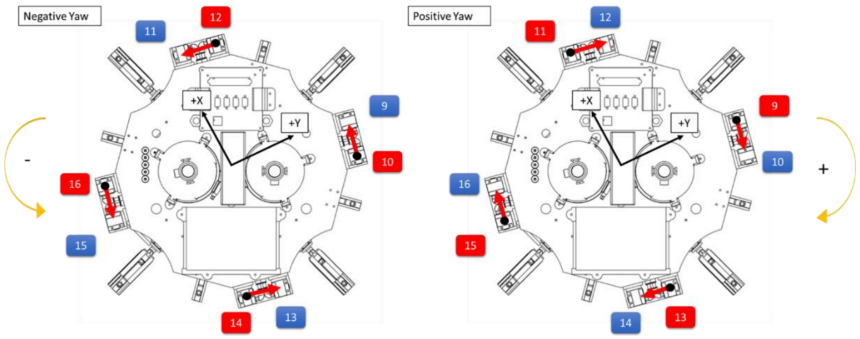


Fig. 25. Yawing maneuver. Negative with thrusters number 10, 12, 14, and 16. Positive with thrusters number 9, 11, 13, and 15.

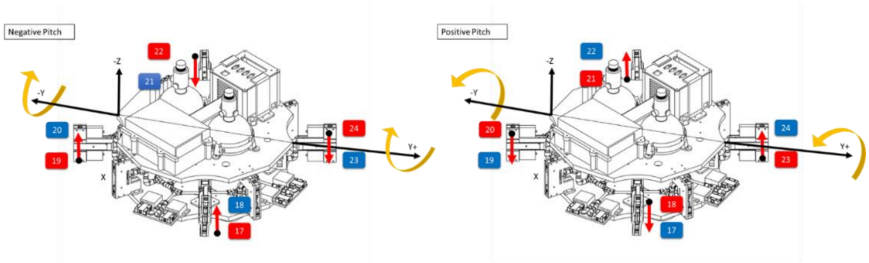


Fig. 26. Thrust Vectoring Control Pitching maneuver. Negative with thrusters number 17, 19, 22, and 24. Positive with thrusters number 18, 20, 21, and 23.

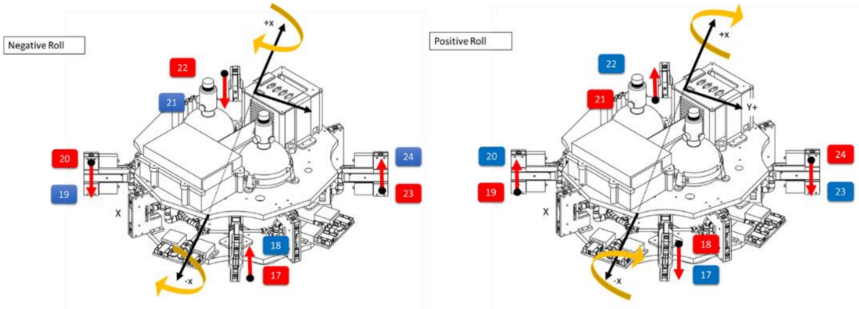


Fig. 27. Thrust Vectoring Control Rolling maneuver. Negative with thrusters number 17, 20, 22, and 23. Positive with thrusters number 18, 19, 21, and 24.

of EASY, respectively. It is important to note that this equations consider only the sixteen fixed thrusters configuration.

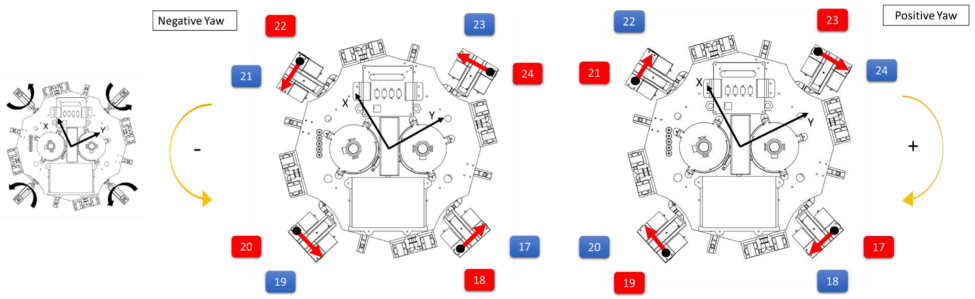


Fig. 28. Thrust Vectoring Control Yawing maneuver after rotating servomotors. Negative with thrusters number 18, 20, 22, and 24. Positive with thrusters number 17, 19, 21, and 23.

$$\sum \vec{F}_{Spacecraft} = \begin{bmatrix} (-T_9 + T_{10} + T_{15} - T_{16})cos(15°) + (-T_{11} + T_{12} + T_{13} - T_{14})sin(15°) \\ (-T_9 + T_{10} + T_{15} - T_{16})sin(15°) + (T_{11} - T_{12} - T_{13} + T_{14})cos(15°) \\ (T_2 + T_4 + T_6 + T_8) - (T_1 + T_3 + T_5 + T_7) \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}, \tag{46}$$

$$\sum \vec{M}_{Spacecraft} = \begin{bmatrix} L_y(T_2 + T_4 + T_5 + T_7) - L_y(T_1 + T_3 + T_6 + T_8) \\ L_x(T_2 + T_8 + T_3 + T_5) - L_x(T_1 + T_7 + T_4 + T_6) \\ L_z(T_9 + T_{11} + T_{13} + T_{15}) - L_z(T_{10} + T_{12} + T_{14} + T_{16}) \end{bmatrix} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}, \tag{47}$$

where L_x, L_y, and L_z are the corresponding arm distances from the CG of EASY to the actuator locations.

In order to achieve the required forces and moments commanded from the controllers, eqs. (46) and (47) must be solved to find the exact thrust required per each actuator. It can be seen that there are a total of six equations for six unknowns, but since there is a total of sixteen actuators, the system is over actuated. In order to guarantee that a global solution for the forces and moments equations is obtained for all instants of time, a control logic was employed in which each pair of upper and immediately lower thruster is treated as one actuator. If a positive thrust is demanded then the lower thruster is activated; on the other hand if a negative thrust is demanded the upper one is activated. This conducts to a new configuration based on thruster's pair, as described in Table IV.

Thus, eqs. (46) and (47) can be rewritten with the new configuration as in eqs. (48) and (49).

TABLE IV
CONFIGURATION OF THRUSTERS FOR CONTROL ALLOCATION.

Name	**Thruster numbers**
T_E	7, 8
T_F	3, 4
T_G	1, 2
T_H	5, 6
T_A, T_B, T_C, T_D	9 to 16

$$\sum \vec{F}_{Spacecraft} = \begin{bmatrix} (T_D - T_A)cos(15°) + (T_C - T_B)sin(15°) \\ (T_D - T_A)sin(15°) - (T_C - T_B)cos(15°) \\ -(T_E + T_F + T_G + T_H) \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{48}$$

$$\sum \vec{M}_{Spacecraft} = \begin{bmatrix} L_y(T_E + T_H) - L_y(T_F + T_G) \\ L_x(T_F + T_H) - L_x(T_E + T_G) \\ L_z(T_A + T_B + T_C + T_D) \end{bmatrix} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \tag{49}$$

It is important to note that the spacecraft will just perform rotational motion, no translation motion is considered acting in x and y, therefore, F_x and F_y are always zero. Since this would leave five variables and four equations, the force needed for translation motion in z axis will be divided in two. Eq. (50) describe the new set of equations used to find the thrust forces.

$$\begin{bmatrix} F_z/2 \\ F_z/2 \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 \\ L_y & -L_y & -L_y & L_y & 0 \\ -L_x & L_x & -L_x & L_x & 0 \\ 0 & 0 & 0 & 0 & L_z \end{bmatrix} \begin{bmatrix} T_E \\ T_F \\ T_G \\ T_H \\ T_{ABCD} \end{bmatrix} \tag{50}$$

Then, eqs. (51) to (55) are obtained after solving for each force and moment. It is important to notice that a maximum of five thrusters will be acting at the same time, and therefore there are a total of five equations for five unknowns at every instant of time.

$$T_E = -\frac{F_z}{2} + \frac{M_x}{4L_y} - \frac{M_y}{4L_x} \tag{51}$$

$$T_F = -\frac{F_z}{2} - \frac{M_x}{4L_y} + \frac{M_y}{4L_x} \tag{52}$$

$$T_G = -\frac{F_z}{2} - \frac{M_x}{4L_y} - \frac{M_y}{4L_x} \tag{53}$$

$$T_H = -\frac{F_z}{2} - \frac{M_x}{4L_y} + \frac{M_y}{4L_x} \tag{54}$$

$$T_{ABCD} = -\frac{M_z}{L_z} \tag{55}$$

It is important to mention that for this case of study, the z axis translation motion is neglected, as the IGOR system would require more work to get it up and running, and this work focuses just in attitude control, thus F_z is considered to be always zero. Therefore, the forces on the three axes are zero, so the spacecraft remains stationary, controlling only its rotation. However, it is necessary to describe these control allocation system of equations since EASY spacecraft could have the ability to control z axis translation motion, even if it is not being used for this work.

C. Stability and Performance Metrics

In order to have a proper assessment of the different controllers, it is important to establish some qualitative performance criteria that measure how good is the controller in terms of attitude tracking error and angular rates as well as the total activity of the actuators (in this case solenoid valves). This is achieved by recording the time history of different sensors from which tracking error and commanded values can be used for subsequent analysis. By their definitions, performance metrics are meant to penalize variables of interest for each flight, for this reason, a higher or lower value mean a worse or better performance of the controllers, respectively, except for the global performance index, which gives an overall value based on the performance metrics previously defined. This index is better when it has a higher value.

1) Angular Rate Activity

Eq. (56) shows the performance metric corresponding to the sum of the Root Mean Square (RMS) of the three angular rate signals. This performance metric is defined in order to quantify the total rotation motion of the spacecraft, since high and long angular rate values are undesired for attitude control.

$$\tilde{e}_{\Omega} = \sqrt{\int_0^T \omega_x^2 \, dt} + \sqrt{\int_0^T \omega_y^2 \, dt} + \sqrt{\int_0^T \omega_z^2 \, dt} \ , \tag{56}$$

2) Quaternion Tracking Error

The Quaternion Tracking Error performance metric corresponds to the RMS of the total unit quaternion tracking error, as shown in eq. (57). Additionally, eq. (58) shows a time-dependent quaternion error metric. These metrics are relevant as they give an idea of how the controllers perform in steady state.

$$\delta\tilde{\boldsymbol{q}} = \sqrt{\int_0^T [(1-\delta q_0)^2 + \delta q_1^2 + \delta q_2^2 + \delta q_3^2] \, dt} \ , \tag{57}$$

$$\delta\tilde{\boldsymbol{q}}_{\boldsymbol{t}} = t\ \delta\tilde{\boldsymbol{q}}\ , \tag{58}$$

where δq_i are the components of the quaternion error vector $\delta\vec{q}$, as will be described in Chapter VII.

3) Solenoid Activity

The Solenoid Activity performance metric corresponds to the total actuation activity of the solenoids. This is calculated using eq. (59) and refers to the RMS of the solenoid PWM commanded signal and is basically the RMS of the total time the solenoid valves remained open. This performance metric is important as fuel consumption is a main driver for spacecraft missions.

$$\tilde{s} = \sum_{i=1}^{16} \sqrt{\int_0^T S_i(t)\, dt}\ , \tag{59}$$

4) Global Performance Index

In order to have an overall performance metrics combining the previous set of metrics, a Global Performance Index can be computed using different weights for each of the indices in eqs. (56) to (59) considering relative importance or weight on the study. For that effect, the expression shown in eq. (60) was used to calculate the global performance index.

$$\begin{aligned} P_I &= 1 - \frac{1}{4}\left(\frac{\tilde{e}_\Omega}{C_\Omega} + \frac{\delta\tilde{\boldsymbol{q}}}{C_{\Delta\boldsymbol{Q}}} + \frac{\delta\tilde{\boldsymbol{q}}_{\boldsymbol{t}}}{C_{\Delta\boldsymbol{Q}_{\boldsymbol{t}}}} + \frac{\tilde{s}}{C_{\Delta S}}\right)\ , \\ \tilde{P}_I &= P_I / C_{P_I}\ , \end{aligned} \tag{60}$$

where C_Ω, $C_{\Delta\boldsymbol{Q}}$, $C_{\Delta\boldsymbol{Q}_{\boldsymbol{t}}}$, and $C_{\Delta S}$ are cut-off values used to normalize the resultant index with respect to the worst case (biggest value) from each set of tests, and C_{P_I} is a cutoff value to normalize the resultant global performance index with respect to the best nominal case of the set of tests considered. It can be seen that each metric is weighted equally.

VII. DEVELOPMENT OF FAULT-TOLERANT CONTROLLERS

Fault-tolerant controllers ensure reliable system operation by responding to faults. They incorporate redundancy and advanced recovery mechanisms to minimize disruptions and maintain stability. This chapter describes the development of a Fuzzy Logic Controller (FLC) and gives the mathematical theory for two types of NLDI controllers implemented.

A. Fuzzy Logic Controller (FLC) Development

The FLC used in this work was designed based on the Sugeno structure, this due to its benefits on computational processing on the defuzzification process, as described in Chapter I Section 6 [5]. Figure 29 shows the controller process.

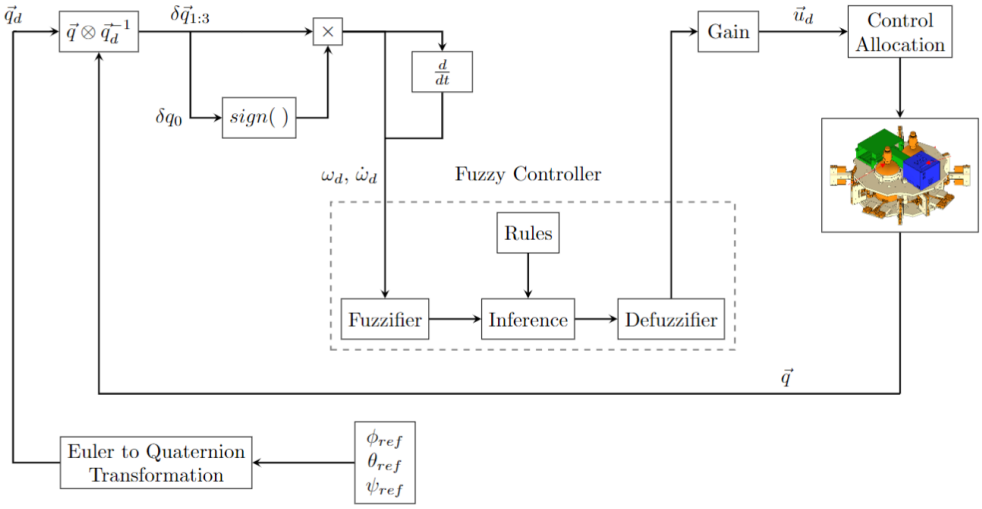


Fig. 29. Fuzzy controller architecture block diagram.

Initially, ϕ_{ref}, θ_{ref}, and ψ_{ref} are the Euler angles that describe the desired spacecraft attitude. Then, they are transformed into quaternions $\vec{q}_d$. Then, from the spacecraft dynamics, the quaternion error can be calculated using eq. (61).

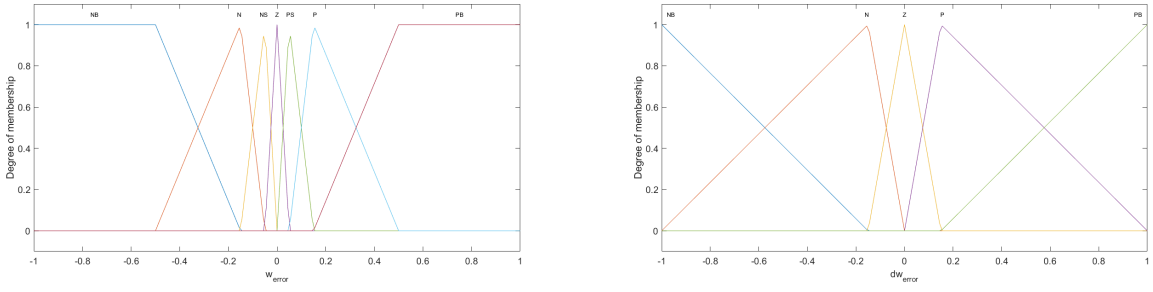
$$\delta\vec{q} = \vec{q}_d \otimes \vec{q}_a^{-1} \,, \tag{61}$$

where $\delta\vec{q}$ is the quaternion error vector, $\vec{q}_d$ is the desired quaternion, $\vec{q}_a^{-1}$ is the inverse of the actual quaternion, and $\otimes$ is the quaternion product [76].

Subsequently, the desired angular rate $\vec{\omega}_d$ can be defined as in eq. (62).

$$\vec{\omega}_d = [\delta q_0 \delta q_1 \quad \delta q_0 \delta q_2 \quad \delta q_0 \delta q_3]^T \tag{62}$$

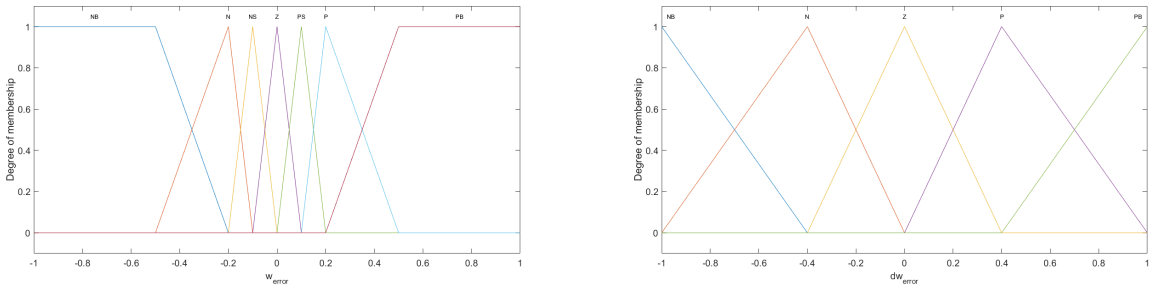
This vector and its rate of change are taken as input signals for the fuzzy controller. Then, the fuzzifier converts those signals into linguistic values defined as Negative Big (NB), Negative (N), Negative Small (NS), Zero (Z), Positive Small (PS), Positive (P), and Positive Big (PB), this process is called fuzzification, and is carried out by a set of membership functions, which are shown in Figures 30 and 31. Then, having the linguistic values, an inference process is carried out based on a collection of rules that relate both inputs to an output. These rules are governed according to Table V. As Sugeno structure suggest for defuzzification process, there are no output membership functions, just discrete values representing a fraction of the maximum actuation. This structure was implemented instead of Mamdani structure, as its defuzzification process requires less calculations and variables' size, thus requiring less computational effort [5], [77]. Finally, as the output value of the fuzzy controller is normalized, a tuning process could be done by varying an output gain from the controller to the spacecraft ACS. This could be seen as varying the fraction of the maximum actuation, thus adapting the controller properties to the real platform.



(a) $\boldsymbol{\omega_d}$ membership functions for roll and pitch.

(b) $\boldsymbol{d\omega_d/dt}$ membership functions for roll and pitch.

Fig. 30. Fuzzy input membership functions for roll and pitch axes.



(a) $\boldsymbol{\omega_d}$ membership functions for yaw.

(b) $\boldsymbol{d\omega_d/dt}$ membership functions for yaw.

Fig. 31. Fuzzy input membership functions for yaw axis.

TABLE V

RULE TABLE FOR MEMBERSHIP FUNCTIONS.

		$\boldsymbol{\omega_d}$						
		NB	**N**	**NS**	**Z**	**PS**	**P**	**PB**
$\frac{d\omega_d}{dt}$	**NB**	1	1	1	1	1	1	1
	N	1	1	1	1	1	0	-1
	Z	1	1	1	0	-1	-1	-1
	P	1	0	-1	-1	-1	-1	-1
	PB	-1	-1	-1	-1	-1	-1	-1

B. Nonlinear Dynamic Inversion (NLDI) Controller

In order to perform a comparison between the fuzzy-logic controller described above and other "high level" fault-tolerant controllers, an NLDI controller was implemented. The "high level" adjective is due to its nonlinearity architecture, being able to model phenomena in a non-linear way, which can be more accurate as real physics is nonlinear.

Recalling the system of equations of motion in nonlinear state space form from eq. (33) and replacing the quaternion definition to the quaternion error using eq. (61), the system becomes as described in eq. (63).

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) = \begin{cases} \delta\dot{\vec{q}} = \frac{1}{2}\Omega(\omega)\delta\vec{q} \\ \dot{\vec{\omega}} = J^{-1}[-\vec{\omega} \times J\vec{\omega} + \vec{M}] \end{cases} , \tag{63}$$

where the state vector corresponds to $\vec{x} = [\vec{\omega},\ \delta\vec{q}]^T$, the control law for the NLDI controller is proposed by Markley and Crassidis [62] as in eq. (64).

$$\vec{M}(t) = \vec{\omega} \times (J\vec{\omega}) + J\vec{u}_v(t) \ , \tag{64}$$

where $\vec{u}_v(t)$ is a virtual controller or pseudo-controller chosen to stabilize $\vec{\omega}$ to zero and $\delta\vec{q}$ to the identity quaternion, and has the form shown in eq. (65).

$$\vec{u}_v(\delta\vec{q}, \vec{\omega}) = -K_\Omega\vec{\omega} - K_q\delta\vec{q}_{1:3} \ , \tag{65}$$

where K_Ω and K_q are pseudo gains defined as:

$$K_\Omega = \begin{bmatrix} k_{\omega_x} & 0 & 0 \\ 0 & k_{\omega_y} & 0 \\ 0 & 0 & k_{\omega_z} \end{bmatrix} \quad K_q = \begin{bmatrix} k_{q_1} & 0 & 0 \\ 0 & k_{q_2} & 0 \\ 0 & 0 & k_{q_3} \end{bmatrix} \tag{66}$$

Substituting eq. (65) in eq. (64), the control law discribed by eq. (67) is obtained.

$$\vec{M}(t) = \vec{\omega} \times (J\vec{\omega}) + J[-K_\Omega\vec{\omega} - K_q\delta\vec{q}_{1:3}] \tag{67}$$

A slight modification to the virtual controllers is introduced to guarantee the shortest path in final orientation, as described by Markley and Crassidis [62] with eq. (68).

$$\vec{u}_v(\delta\vec{q}, \vec{\omega}) = -K_\Omega\vec{\omega} - K_q sign(\delta q_0)\delta\vec{q}_{1:3} \tag{68}$$

Then, the final control input for the Spacecraft would be the one shown in eq. (69).

$$\vec{M}(t) = \vec{\omega} \times (J\vec{\omega}) + J[-K_\Omega \vec{\omega} - K_q sign(\delta q_0)\delta\vec{q}_{1:3}] \tag{69}$$

Finally, the closed-loop system after implementing the control law in eq. (67) has the form shown in eq. (70).

$$f_{NLDI}(\delta\vec{q}, \vec{u}_v) = \begin{cases} \delta\dot{\vec{q}} = \frac{1}{2}\Omega(\omega)\delta\vec{q} \\ \dot{\vec{\omega}} = \vec{u}_v(t) \end{cases} \tag{70}$$

Figure 32 shows a block diagram for the architecture of the quaternion based NLDI controller described.

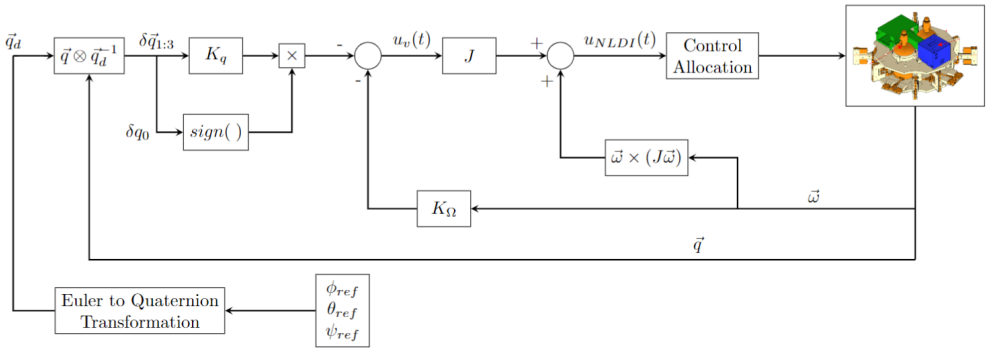


Fig. 32. Quaternion based NLDI controller block diagram.

C. NLDI Augmented with Artificial Immune System (NLDI+AIS)

Another controller previoslly developed at ADCL, called NLDI augmented with Artificial Immune System (NLDI+AIS) was implemented.

The NLDI+AIS controller is based on a dual approach compiling feedback linearization and Model Reference Adapative Immune System (MRAIS) to avoid disturbances. This

architecture guarantees that the closed loop response behaves as similar as possible to a desired system dynamics and also contains a set of nonlinear functions describing the response of the immune system of living organism against external hazards [37], [78].

Similar as NLDI, the control input is defined by eq. (71).

$$\vec{M}(t) = \vec{u}_{NLDI}(t) = \vec{\omega} \times (J\vec{\omega}) + J\vec{u}_v(\vec{q}, \vec{\omega}, t) \tag{71}$$

For this case, $\boldsymbol{u}_v(\delta\vec{q}, \vec{\omega}, t)$ will be a time-varying virtual adaptive controller. Then, the closed loop system after implementing the control law of eq. (71) becomes eq. (72).

$$f_{NLDI+AIS}(\delta\vec{q}, \vec{u}_v) = \begin{cases} \delta\dot{\vec{q}} = \frac{1}{2}\Omega(w)\delta\vec{q} \\ \dot{\vec{\omega}} = \vec{u}_v(\delta\vec{q}, \vec{\omega}, t) \end{cases} \tag{72}$$

It is important to notice that in this case, the control input $\vec{u}_v$ is a time-dependent function, as it varies according to certain conditions. The development of those variations are deduced after several mathematical operations shown by Perez and Moncayo [37] and Perez Rocha [78], where the authors show a bio-inspired controller, since it uses an architecture that mimics the autoimmune system of some living beings, hence its name. Thus, the final control law can be written as in eq. (73).

$$\vec{u}_v(\vec{x}, t) = -C(t)\vec{\omega} - K_Q(t)\delta\vec{q}\; sign(\delta q_4)\;, \tag{73}$$

where $C(t)$ and $K_Q(t)$ are defined by eq. (74).

$$C(t) = \begin{bmatrix} c_1(t) & 0 & 0 \\ 0 & c_2(t) & 0 \\ 0 & 0 & c_3(t) \end{bmatrix},\; K_Q(t) = \begin{bmatrix} k_{q_1}(t) & 0 & 0 \\ 0 & k_{q_2}(t) & 0 \\ 0 & 0 & k_{q_3}(t) \end{bmatrix}\;, \tag{74}$$

where $c_1(t), c_2(t), c_3(t), k_{q_1}(t), k_{q_2}(t)$, and $k_{q_3}(t)$ are time-variant coefficients which are function of the change of virtual control inputs and the states vector (see eq. (63)). Figure 33 shows a block diagram for the architecture of the adaptive NLDI+AIS controller, where the delays z^{-1} and z^{-d} belong to the logic of the controller measuring changes from initial to current behavior of the system. Their definition is developed by Perez and Moncayo [37] and Perez Rocha [78].

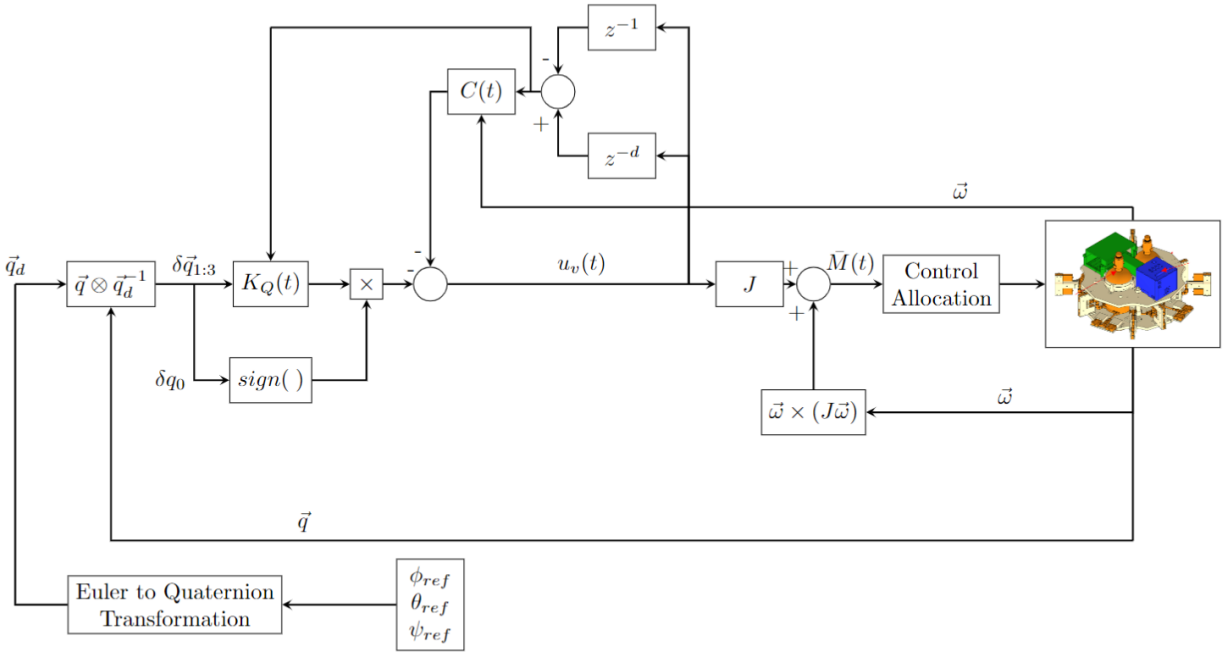


Fig. 33. Adaptive NLDI+AIS controller block diagram.

VIII. RESULTS

A. Fault-Tolerant Controllers Implementation

Table VI shows the maneuver performed by the Spacecraft using each controller in the test bench. For the implementation, the control algorithms were deployed into PC-104 PCM-3355 flight computer through xPC Target. Total thruster failure at a given time were injected while doing the tracking maneuver.

TABLE VI
SPACECRAFT MANEUVER.

Time [s]	0	20
$\boldsymbol{\phi}$ **[deg]**	0	45
$\boldsymbol{\theta}$ **[deg]**	15	0
$\boldsymbol{\psi}$ **[deg]**	0	0

The maneuver scenarios were conducted three times for each controller for 40 seconds so that a correct verification of the operation of each controller can be made. Table VII shows the ID assignment for every flight carried out. Thus, having a total of 27 flights performed for the controllers performance comparison analysis.

Three maneuver scenarios were considered having either nominal and different types of failures. These scenarios are described as follow:

- **Nominal Scenario:** No thruster failures injected.
- **Failure Scenario 1:** Thrusters 1 and 2 stop working after 21 seconds from the beginning of maneuver. This scenario was chosen as thrusters 1 and 2 highly affect pitch maneuver and also have an influence in roll.
- **Failure Scenario 2:** Thrusters 7 and 8 stop working after 21 seconds from the beginning of maneuver.

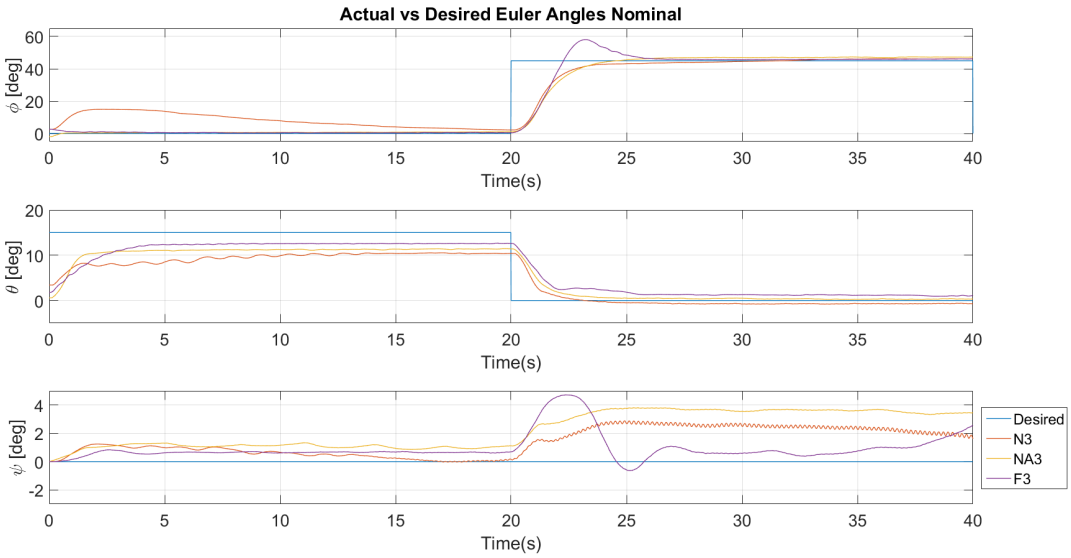
Since at 21 seconds from the beginning of maneuver the spacecraft is performing roll and pitch tracking maneuvers, the failures were injected at this time to see their performance.

TABLE VII
TEST FLIGHT IDS FOR CONTROLLERS PERFORMANCE COMPARISON.

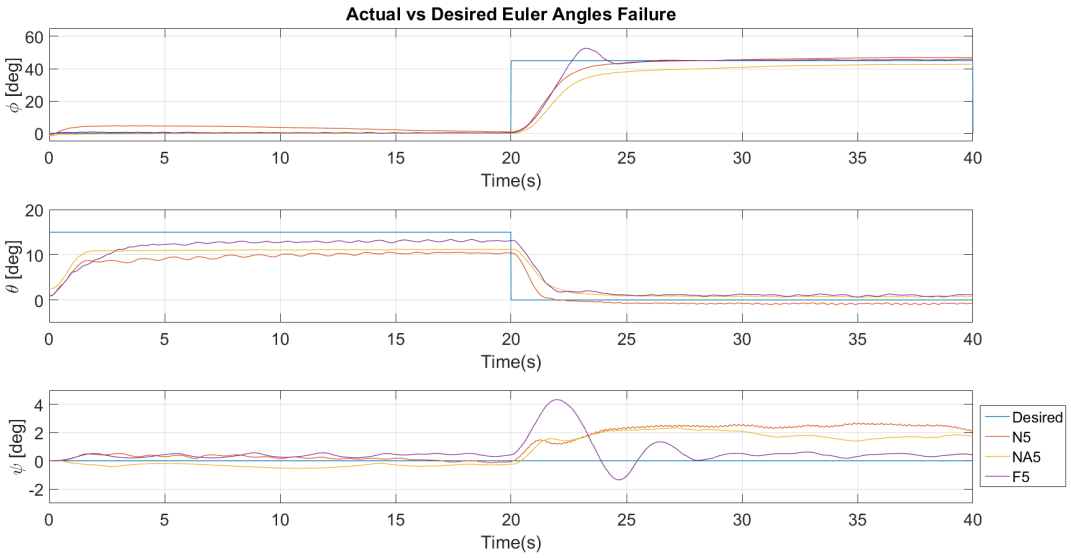
Maneuver Scenario	**NLDI**	**NLDI+AIS**	**Fuzzy**
Nominal	N1	NA1	F1
	N2	NA2	F2
	N3	NA3	F3
Failure 1	N4	NA4	F4
	N5	NA5	F5
	N6	NA6	F6
Failure 2	N7	NA7	F7
	N8	NA8	F8
	N9	NA9	F9

Other types of failures were injected to see the simulation fitting performance results, as will be shown in the next section. Yaw axis tracking maneuvers remained zero due to test bench mechanical limitations on its design, since the gimbaled platform described in Chapter VI has a lot of inertia in this specific axis.

Figure 34 shows the Euler angles tracking for a flight test with each controller, respectively, displaying the nominal scenario on its third repetition and failure scenario 1 on its second repetition. Figure 35 presents the thrusters activity for each test shown previously. Likewise, Figures 36 to 39 show the behavior of each performance metric over flight time, respectively. For these performance metrics, a lower value means a better performance, since the definitions shown in Chapter VI Section C penalize variables of interest for each flight. Then, Figure 40 presents the final value of each performance metric shown before. Finally, Figure 44 shows the average global performance metric for each controller in both nominal and failure scenarios.

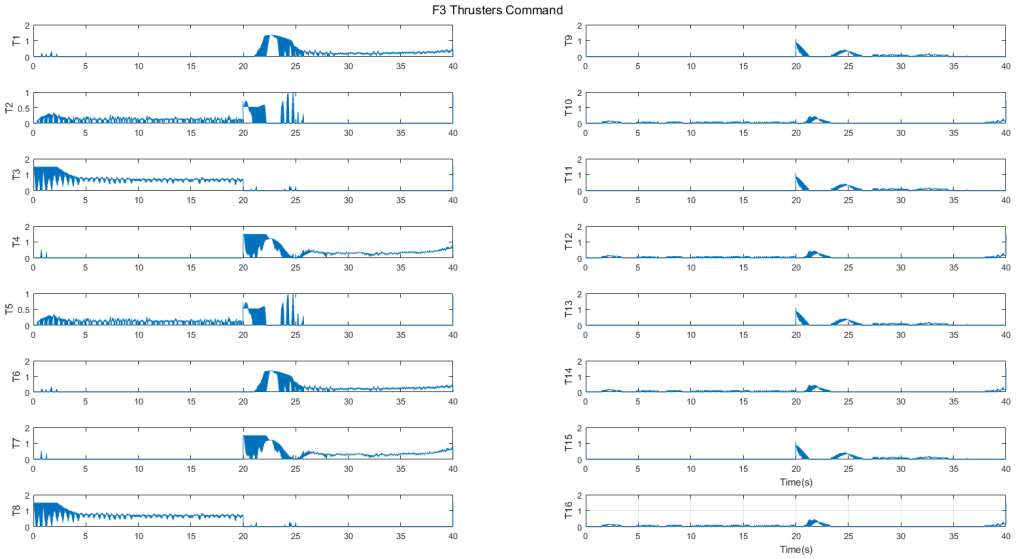


(a) Euler angles tracking nominal test.

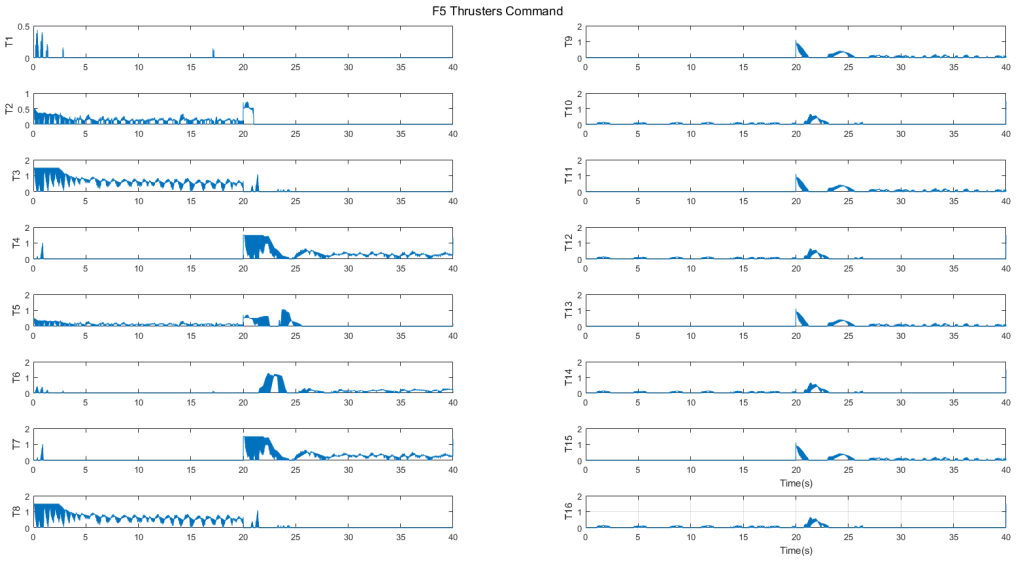


(b) Euler angles tracking failure test.

Fig. 34. Euler angles tracking for flight tests 3 and 5.

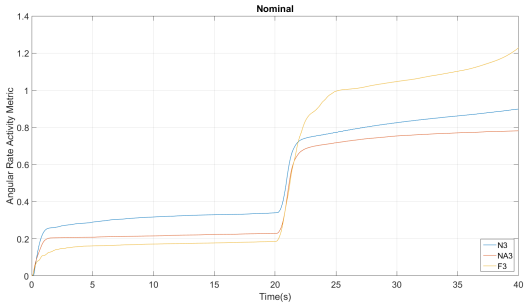


(a) Thrusters activity nominal test.

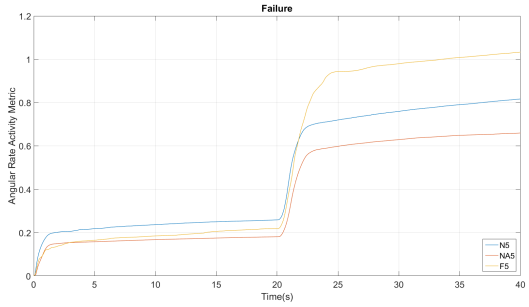


(b) Thrusters activity failure test.

Fig. 35. Thrusters activity over flight time for flight tests 3 and 5.

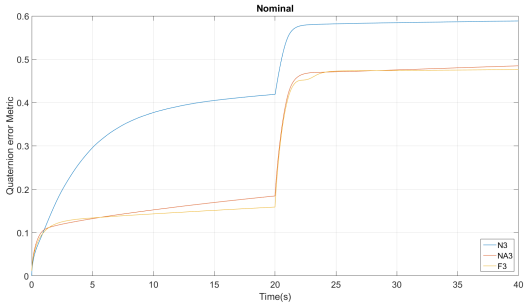


(a) Angular rate activity metric nominal test.

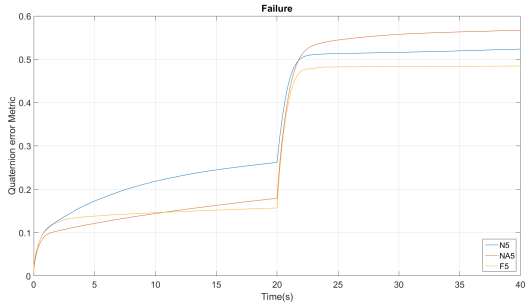


(b) Angular rate activity metric failure test.

Fig. 36. Angular rate activity metric over flight time for flight tests 3 and 5.

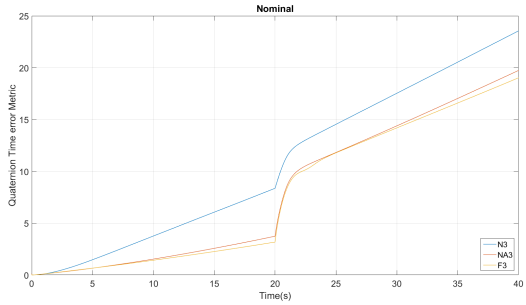


(a) Quaternions error metric nominal test.

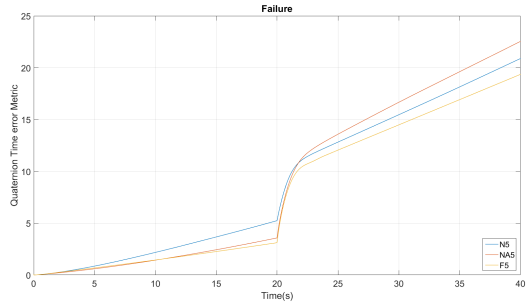


(b) Quaternions error metric failure test.

Fig. 37. Quaternions error metric over flight time for flight tests 3 and 5.

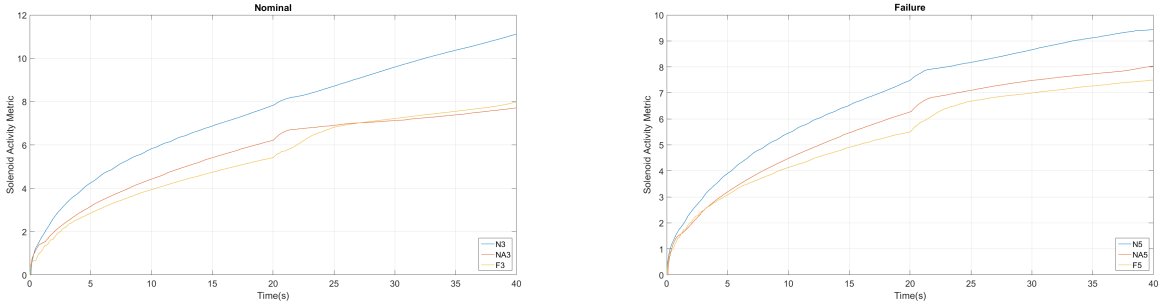


(a) Quaternions time error metric nominal test.



(b) Quaternions time error metric failure test.

Fig. 38. Quaternions time error metric over flight time for flight tests 3 and 5.



(a) Solenoid activity metric nominal test.

(b) Solenoid activity metric failure test.

Fig. 39. Solenoid activity metric over flight time for flight tests 3 and 5.

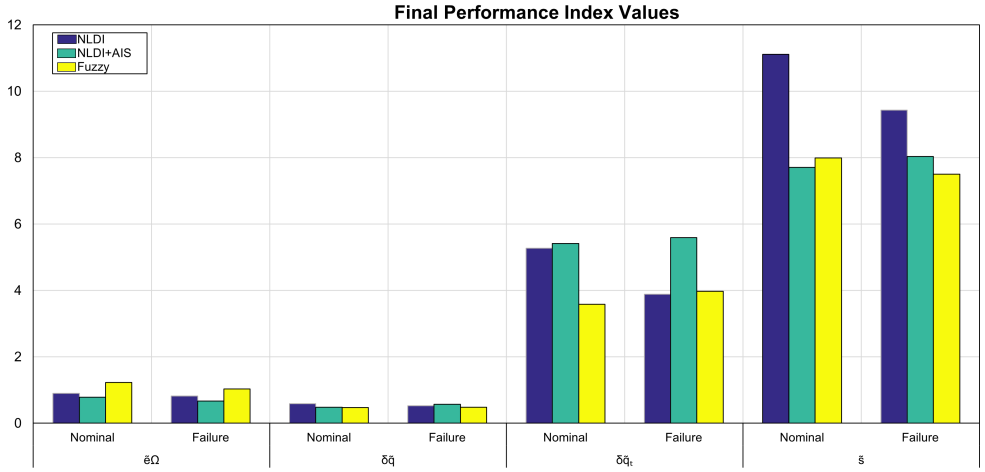


Fig. 40. Final values of performance metrics for flight tests 3 and 5. Better for lower values.

The FLC, as can be seen in Figure 34, has a good performance for steady state error for both nominal and failure scenarios, keeping the tracking error below 5 degrees. However, it shows an overshooting for roll axis with an amplitude of around 15 degrees. Also, after 20 seconds of test time, when roll and pitch commanded angles change at the same time, there is a yaw influence. This behavior can be attributed to the ACS thrusters misalignment, in which a thruster can influence other axes depending on the current attitude, as can be seen in 35, where yaw thrusters 9, 11, 13, and 15 turn on after the change of angle at 20 seconds, and thrusters 10, 12, 14, and 16 slightly turn on while the spacecraft is performing the first roll maneuver, where some thrusters produce torques in yaw axis.

Additionally, the FLC demonstrates superior performance in most metrics, sometimes even better than NLDI+AIS controller. It can be noticed that for $\delta\tilde{\boldsymbol{q}}$ and $\delta\tilde{\boldsymbol{q}}_{\boldsymbol{t}}$ metrics, the values are lower for the failure scenario with NLDI+AIS and Fuzzy controllers, as opposed to the NLDI controller. As can be seen in Figures 36 to 38, performance metrics can improve or worsen through the flight, suggesting that a controller can be better or worse than others performing certain maneuvers.

In order to see the performance of all the 27 flight tests carried out, Figures 41a, 41b, and 41c show the results for each scenario with each controller. Also, Figure 42 displays a bar chart of all flight metrics final values. A minimum-maximum normalization process was applied for comparison purposes, and Figure 43 presents the average of these final metric values. It is important to recall that for these performance metrics, a lower value means a better performance. Finally, Figure 44 shows the global performance metric for each controller. This plot is the main relevance as it gives an overall result taking into account each performance metric.

As can be seen in Figure 41, FLC had the worse performance in some flights, having high steady state errors in all axes. However, for some other flights, it shows the best performance for steady state error, being able to control roll axis with some overshooting to almost zero, and yaw axis with some oscillations. NLDI and NLDI+AIS controllers were able to control all three axes with some steady state error. It can be noticed that NLDI+AIS prioritizes steady state error over time to reach steady state.

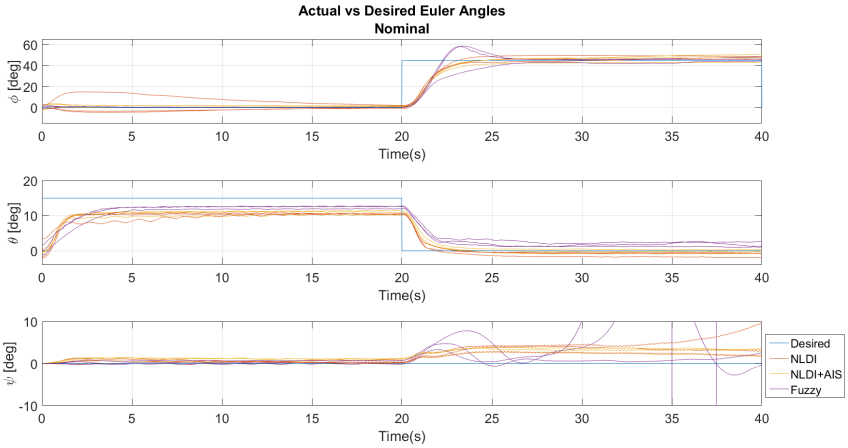
Additionally, for nominal scenario, the FLC had divergent behavior in the yaw axis for one flight test, starting to rotate away from the commanded set point. However, in another flight test it was able to control better than NLDI and NLDI+AIS.

The final values of performance metrics for all tests shows that NLDI+AIS had the best performance with angular rate activity $\tilde{e}_{\Omega}$ for all tests, followed by NLDI and FLC, which the last showed a lot of angular activity compared to the others. For the quaternion error metric $\delta\tilde{\boldsymbol{q}}$, it can be seen that for the nominal scenarios, the FLC behaved better in one out of three tests than the other controllers, and for failure scenarios, the FLC showed a better performance than NLDI and NLDI+AIS in some flights, having the best result in the

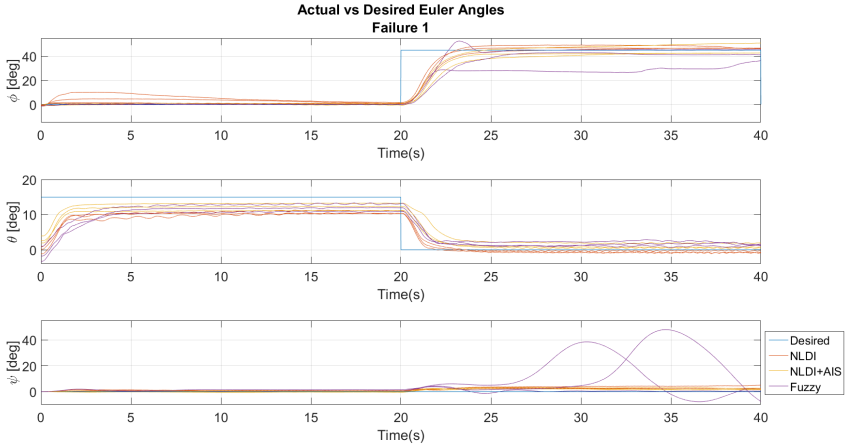
tests flight F3. For the time-dependent quaternion error metric $\delta\tilde{\boldsymbol{q}}_{\boldsymbol{t}}$, the FLC had the worse result on flight F7, this due to the divergence on yaw axis described before, so a comparison of this metric is difficult to accomplish in this situation, however, NLDI and FLC showed to have similar behaviors. Finally, the solenoid activity metric $\tilde{s}$, NLDI and FLC controllers behaved similar in most of the failure scenarios, and NLDI+AIS had the best results overall.

Extending the previous insights, Figure 43 shows the average of the performances metrics for each flight scenario. These results suggest that for nominal scenario, NLDI and FLC behaved similar for all performance metrics. The FLC shows a better performance in the solenoid activity metric on nominal scenario. In general, the FLC showed to have worse results than NLDI and NLDI+AIS, however, it can be comparable to NLDI results for most of the performance metrics.

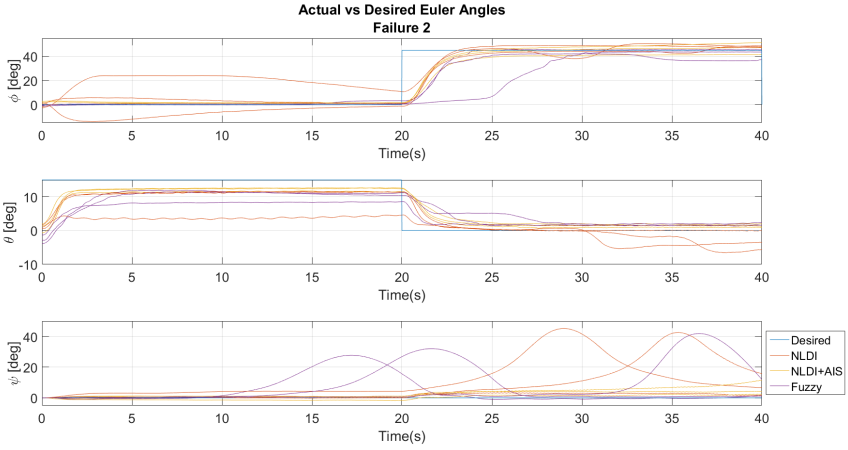
Finally, the average global performance index (Figure 44) presents insights about the overall performance of the three controllers. The values are normalized, as described in Chapter VI Section C, so they serve as a reference for comparison purposes. For this performance index, a higher value means a better performance. It shows that the FLC performed better than NLDI for nominal and failure scenario 2, and worse for failure scenario 1. The NLDI shows better results in failure scenario 1 than in the other scenarios, in contrast with FLC, which has the opposite behavior. At last, the NLDI+AIS performed the best of all, showing better performance for failure scenario 1 than for the other scenarios.



(a) Euler angles tracking all nominal scenario tests.



(b) Euler angles tracking all failure scenario 1 tests.



(c) Euler angles tracking all failure scenario 2 tests.

Fig. 41. Euler angles tracking for all flight scenarios.

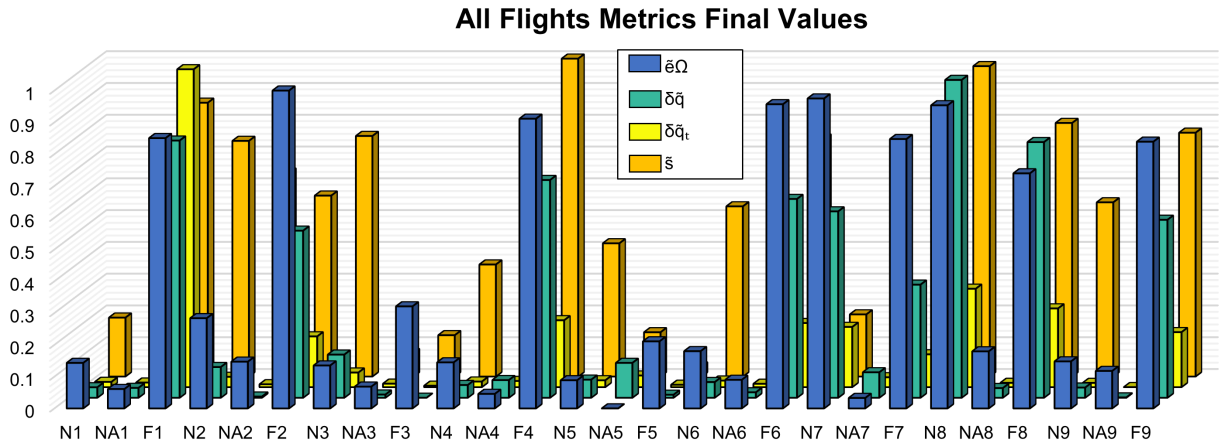


Fig. 42. Final values of performance metrics for all tests. Better for lower values.

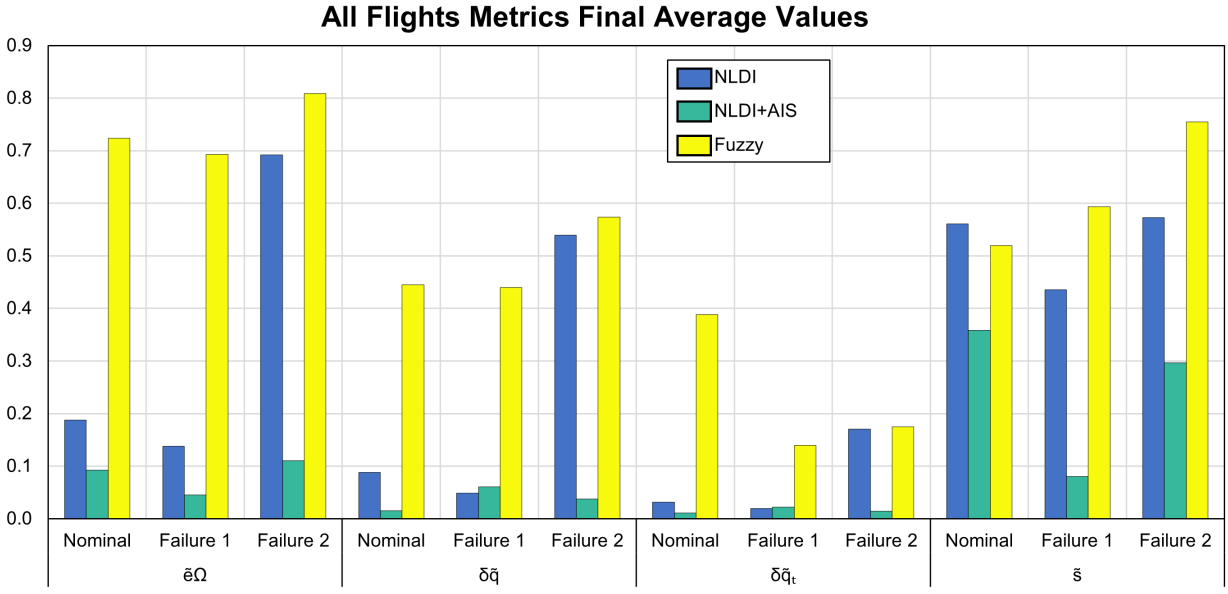


Fig. 43. Final average values of performance metrics for all tests. Better for lower values.

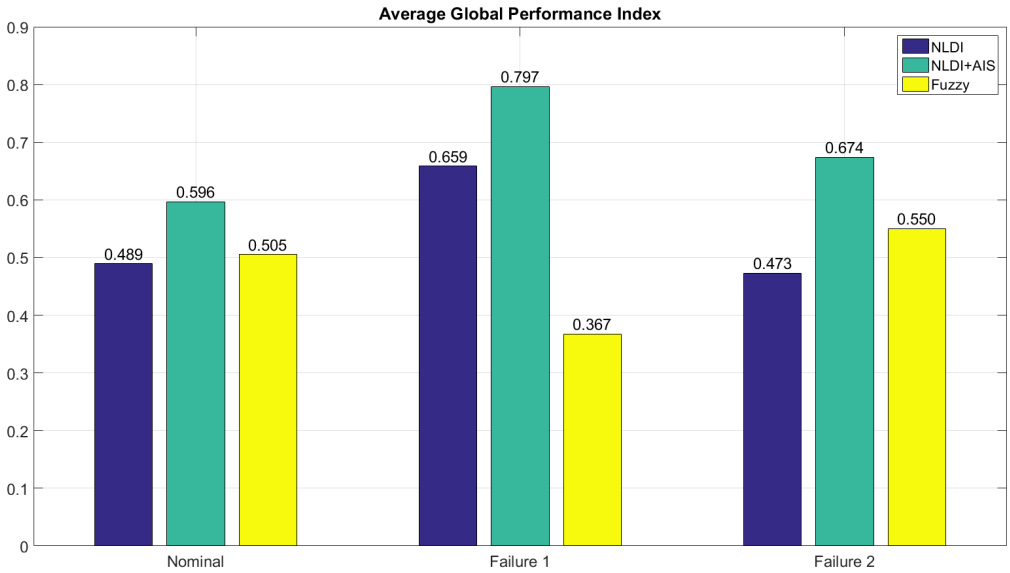


Fig. 44. Average global performance metric. Better for higher values.

B. Simulation Model Performance

For the simulation framework performance evaluation, RMSE values between simulation models and implementation data were calculated. For this purpose, the flights performed for the controllers assessment were taken, and also, other 82 flight tests were performed with different controllers, thus having a total of 109 flight tests, which can be seen in Appendix A. Here, an undisturbed model, referred to as the "Old model", and a disturbed model, referred to as the "New model", are compared.

Figure 45 shows a comparison between each model, considering the total maneuver and the transient responses of each test, and also showing the mean of all RMSE values for each case. These values are also shown in Figure 46 for a better visualization.

It can be noticed that for some test flights, the New model performed both better and worse than Old model. However, the overall results show an improvement of the simulation model when disturbances are considered, having around 0.66° and 0.28° mean RMSE less for transient response and total maneuver, respectively. Also, for both models, the total mean RMSE values were better than the transient response, having the highest difference on the

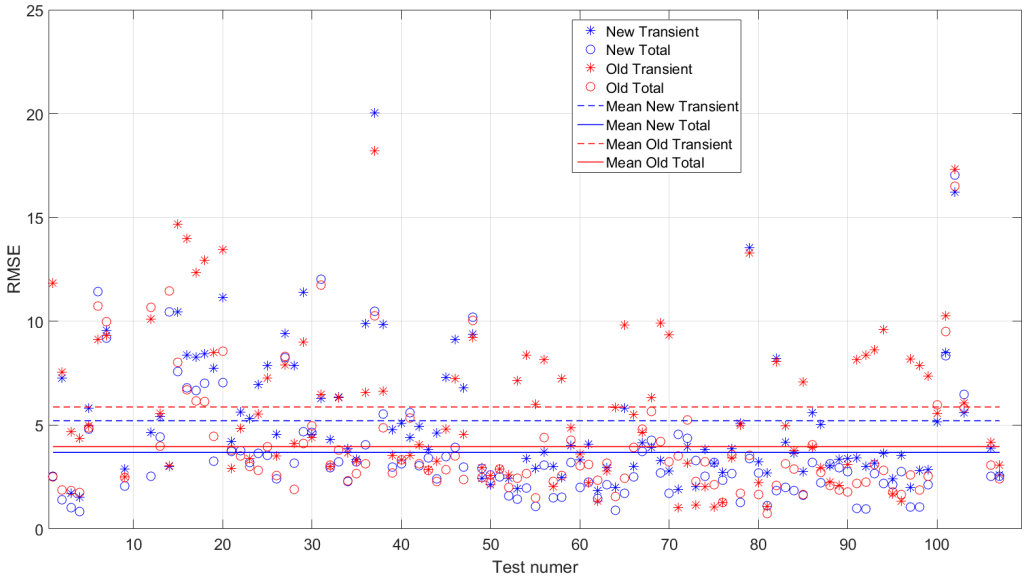


Fig. 45. RMSE of every test flights with each simulation model.

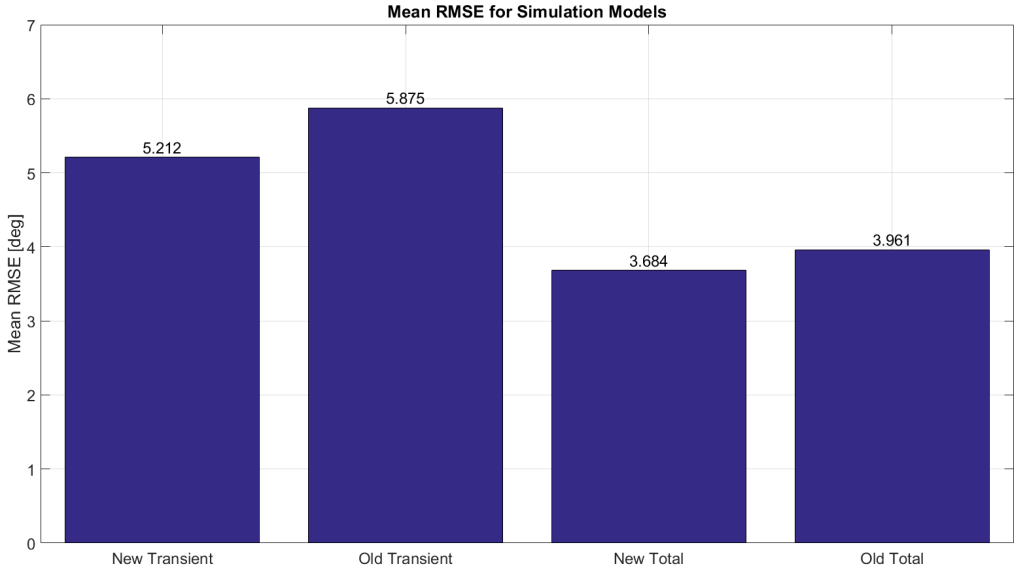


Fig. 46. Mean RMSE values of each simulation model.

Old model with around 1.9° less.

Additionally, a correlation analysis was performed in order to get a value to compare the agreement between the two simulation models data and actual data. In order to do this

correlation analysis, after noticing that yaw axis was having issues due to implementation difficulties, this axis was not considered for this analysis. Then, the data distributions were found using a probability data distribution fitting algorithm [72]. Figure 47 shows the data distributions for roll and pitch axes and Table VIII presents the characteristics of each data distribution, where μ is the mean, σ is the standard deviation, and ν is the degrees of freedom [79].

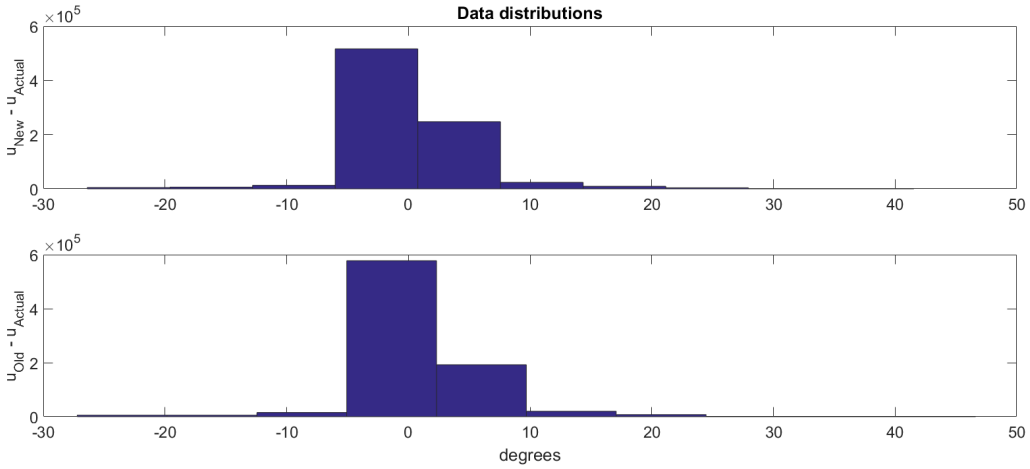


Fig. 47. Data distribution of New and Old model errors.

TABLE VIII
CHARACTERISTICS OF EACH DATA DISTRIBUTION FOR ROLL AND PITCH AXES.

	Disturbed	**Undisturbed**
Distribution Name	T-Distribution	T-Distribution
μ	-0.2223	-0.2707
σ	1.3587	1.2206
ν	1.2571	1.0933

In order to show the performance of the simulations for a flight test, Figure 48 shows the results of actual and simulations for the test flight number 25 (see Appendix A). Finally, the correlation coefficients were calculated for both flight test and all the flight tests carried out. The results are shown in Figure 49, where all p-values were below 0.05, sufficient to say that the coefficients are statistically significant.

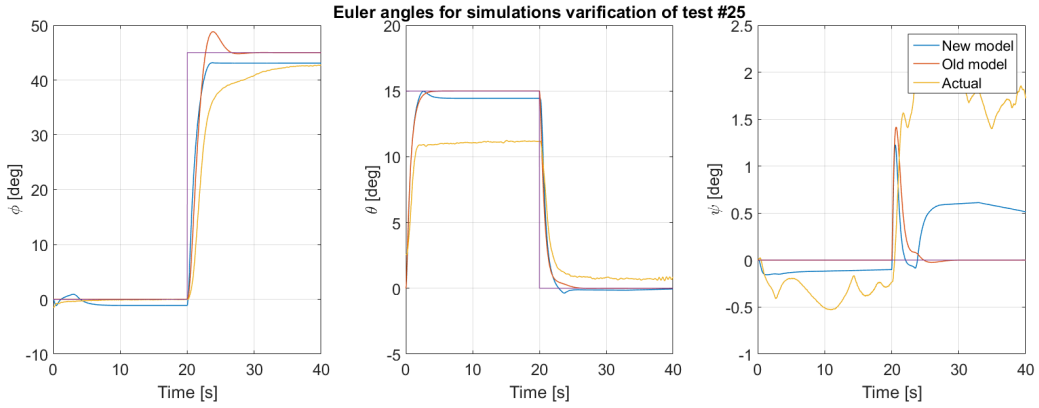


Fig. 48. Actual and Simulation data for flight test #25.

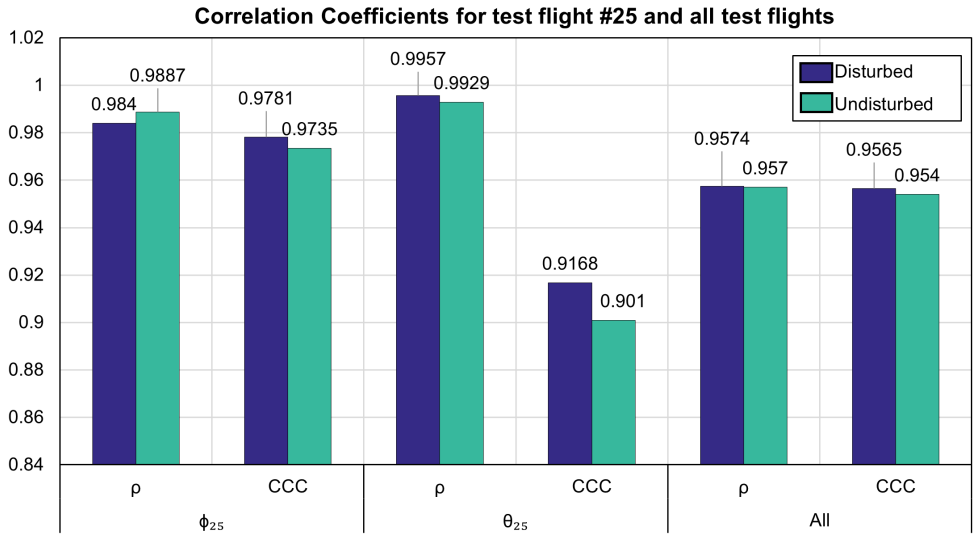


Fig. 49. Correlation coefficients results for rest flight #25 and all test flights.

As mentioned previously, the yaw axis is showing difficulties in implementation data. However, for test flight #25, simulations showed to have a similar behavior when the maneuver was performed. Also, even though the steady state error in pitch angle is different for both simulation models, the behavior of the data shows to be almost the same, having transient responses very close to each other. On the other hand, roll axis simulations (especially in the New model), show to have a similar steady state error, even though the transient response is not very accurate.

Finally, the correlation coefficients show that the undisturbed model performed better

for the roll axis of test flight #25 according to the Pearson's correlation coefficient. However, in all other cases, the coefficients suggest that the disturbed model performs better simulating the actual data. Nevertheless, it is important to note that the coefficients do not vary significantly, all being greater than 0.9.

IX. CONCLUSIONS

The results presented in this thesis demonstrate promising capabilities of the controllers to minimize tracking errors under uncertain conditions. They also show acceptable characteristics that make them suitable for real-time implementations. Specifically, the FLC developed shows performance on a par with NLDI and NLDI+AIS controllers, in some cases even better than NLDI for both nominal and failure scenarios. This demonstrates the capacities that a FLC can have despite that its nature is highly human logic dependent. The design process of the FLC suggests that control action is more critical near to the set-point (or commanded point), that is why MFs are closer to zero and the design process focused in how closer should each MF be to zero. Also, MFs are highly dependent on the physical platform, as disturbances may influence each rotation axis differently. On the other hand, despite good results shown for specific test flights, FLC showed not to be consistent for all test flights, changing the results even for the same flight maneuver. This behavior could indicate that the proposed FLC may be a highly disturbance-dependent controller.

Regardless of good results for the $\delta\tilde{\boldsymbol{q}}_{\boldsymbol{t}}$ performance metric showed better results for the FLC, in space applications the time dependency is not the main driver of spacecraft missions, since smoother maneuvers without overshooting may prevent failures, specially for scientific payload pointing, docking maneuvers, etc. However, this type of controllers provide a powerful tool for initial approach to attitude controllers, making spacecraft physics and control theory easier to understand from a "logical" point of view.

In general, for the comparison of nominal and failure tests individually, the controllers showed an overall good performance on the failure scenario, being able to complete the maneuver even if the thruster failures are injected throughout the change of attitude.

Additionally, the challenges of moving from simulations to implementation include unknown physical disturbances that can be difficult to approach in a simulation environment. Nevertheless, simulations provided the necessary tools to initially test each controller before moving to implementation, saving resources and time needed for EASY operation.

The Old and New model simulations comparison with actual data showed that includ-

ing disturbances can enhance the similarity of simulation data to real data, making control design more accurate before moving to implementation on test bench. In general, both simulations can model the implementation data with an error margin that varies depending on the axis and maneuver performed. However, tools such as ML could be the desired option, since taking into account every disturbance may be very difficult due to high uncertainties in the physical platform. High fidelity simulation flight data can be obtained with ML models, making easier the failure data retrieval which may be difficult to find from real flights and impractical to reproduce.

X. RECOMMENDATIONS

The FLC design of this thesis is considered to be an initial approach, so that it can be given more robustness with advance fault-tolerant characteristics. This can be done by applying model-dependent gains that vary through flight. Some research works, as described in Chapter I Section 6, show control architectures that couple FLC with PID controllers, making a controller that can adapt depending on the output from the plant.

As mentioned before, ML techniques are recommended to obtain a better representation of EASY spacecraft test bench implementation data. Tools such as Support Vector Machine, Autoencoders, etc, could be a good approach for this purpose.

EASY spacecraft showed to be a good platform for spacecraft attitude controllers. However, several improvements can be done in order to generate more accurate data. First, the rotation mechanism for yaw axis varies greatly respect to roll and pitch, as it has more inertia, which makes yaw control more fuel-consuming. Then, mass change rate suggests to be a design problem due to the position of the air tanks, as their position greatly changes the CG of EASY through flights, an undesirable factor on real applications, so horizontal positioning could mitigate this problem. Additionally, IGOR system could not be applied in this work, so translation control action can be developed. Finally, a model for the use of TVC configuration is highly recommended, as this may provide different failure scenarios and more actuation of the propulsion system.

Extensive testing should be performed in order to provide a more comprehensive analysis of the performance of the attitude controllers.

REFERENCES

[1] S. Yin, B. Xiao, S. X. Ding, and D. Zhou, "IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS A Review on Recent Development of Spacecraft Attitude Fault Tolerant Control System," vol. 0046, no. c, 2016. DOI: 10.1109/TIE.2016.2530789.

[2] N. Leveson, "Evaluating Accident Models Using Recent Aerospace Accidents," 2001.

[3] K. A. Weiss, N. Leveson, K. Lundqvist, N. Farid, and M. Strin, "AN ANALYSIS OF CAUSATION IN AEROSPACE ACCIDENTS," pp. 1–12,

[4] C. H. Cheng, S. L. Shu, and P. J. Cheng, "Attitude control of a satellite using fuzzy controllers," *Expert Systems with Applications*, vol. 36, no. 3 PART 2, pp. 6613–6620, 2009, ISSN: 09574174. DOI: 10.1016/j.eswa.2008.08.053.

[5] A. Bello, K. S. Olfe, J. Rodriguez, J. M. Ezquerro, and V. Lapuerta, "Experimental verification and comparison of fuzzy and PID controllers for attitude control of nanosatellites," *Advances in Space Research*, no. xxxx, 2022, ISSN: 18791948. DOI: 10.1016/j.asr.2022.05.055.

[6] B. Lyons, H. Moncayo, A. Noriega, I. Moguel, and M. G. Perhinschi, "Hardware-in-the-Loop simulation of an extended non-linear dynamic inversion augmented with an immunity-based adaptive control system," in *AIAA Modeling and Simulation Technologies (MST) Conference*, American Institute of Aeronautics and Astronautics Inc., 2013. DOI: 10.2514/6.2013-5152.

[7] I. The MathWorks, *Matlab simulink software*, Accessed: 2023-07-13, Massachusetts, US. [Online]. Available: https://www.mathworks.com/products/simulink.html.

[8] N. Bellini, "Magnetic Actuators for Nanosatellite Attitude Control," 2014.

[9] *Cubesat magnetorquer satbus mtq*, Accessed: 2023-06-23. [Online]. Available: https://nanoavionics.com/cubesat-components/cubesat-magnetorquer-satbus-mtq/.

[10] N. Jovanovic, B. Riwanto, P. Niemela, M. R. Mughal, and J. Praks, "Design of Magnetorquer-Based Attitude Control Subsystem for FORESAIL-1 Satellite," vol. 2, no. 4, pp. 220–235, 2021. DOI: 10.1109/JMASS.2021.3093695.

[11] *Reaction/momentum wheel*, Accessed: 2023-06-23. [Online]. Available: https://spinoff.nasa.gov/spinoff1997/t3.html.

[12] *Cubesat reaction wheels control system satbus 4rw0*, Accessed: 2023-06-23. [Online]. Available: https://nanoavionics.com/cubesat-components/cubesat-reaction-wheels-control-system-satbus-4rw/.

[13] Z. Ismail and R. Varatharajoo, "A study of reaction wheel configurations for a 3-axis satellite attitude control," *Advances in Space Research*, vol. 45, no. 6, pp. 750–759, 2010, ISSN: 0273-1177. DOI: 10.1016/j.asr.2009.11.004.

[14] S. Zhaowei, G. Yunhai, X. Guodong, and H. Ping, "The combined control algorithm for large-angle maneuver of HITSAT-1 small satellite," vol. 54, pp. 463–469, 2004. DOI: 10.1016/S0094-5765(03)00223-6.

[15] W.-k. Chu, "HTS Bearings for Space Applications : Reaction Wheel With Low Power Consumption for," no. August, 2015. DOI: 10.1109/TASC.2003.813064.

[16] C. C. Grant and R. E. Zee, "ENABLING REACTION WHEEL TECHNOLOGY FOR HIGH PERFORMANCE," pp. 1–6, 2007.

[17] H. B. Hablani, "Sun-Tracking Commands and Reaction Wheel Sizing with Configuration Optimization," vol. 17, no. 4, 1994.

[18] D. S. Bayard, "An Optimization Result With Application to Optimal Spacecraft Reaction Wheel Orientation Design," pp. 1473–1478, 2001.

[19] H. Paku, R. Takehana, and K. Uchiyama, "Spherical Reaction Wheel System for Satellite Attitude Control," no. January, pp. 1–9, 2016. DOI: 10.2514/6.2016-0693.

[20] M. Schalkowsky S. Harris, "SPACECRAFT MASS EXPULSION TORQUES," no. December, 1969.

[21] NASA, *National air and space museum - smithsonian*, Accessed: 2023-06-22, 2011. [Online]. Available: https://airandspace.si.edu.

[22] P. A. Servidia and R. S. Sanchez Peña, "Thruster Design for Position / Attitude Control of Spacecraft," vol. 38, no. 4, 2002.

[23] M. J. Sidi, *Reaction Thruster Attitude Control*. 2014, ISBN: 9780511815652.

[24] M. Navabi, "Comparing Optimum Operation of Pulse Width- Pulse Frequency and Pseudo-Rate Modulators in Spacecraft Attitude Control Subsystem Employing Thruster," pp. 625–630, 2013.

[25] A. Tavakoli and M. Ashtar, "Feature Article : An Innovative Test Bed for Verification of Attitude Control System," no. 10, 2017.

[26] G. Cervettini and S. Pastorelli, "Development and Experimentation of a CubeSat Magnetic Attitude Control System Testbed," vol. 57, no. 2, pp. 1345–1350, 2021.

[27] D. W. Miller, "Development and Analysis of a Small Satellite Attitude Determination and Control System Testbed LIBRA," no. 2009, 2011.

[28] Caltech, "State and Output Feedback," Accessed: 2023-06-23. [Online]. Available: https://www.cds.caltech.edu/~murray/courses/cds101/fa04/caltech/am04_ch5-24oct04.pdf.

[29] M. I. of Technology, "Understanding Poles and Zeros," pp. 1–13,

[30] I. M. Mehedi, "2548 . Full state-feedback solution for a flywheel based satellite energy and attitude control scheme," pp. 3522–3532, 2017. DOI: 10.21595/jve.2017.18066.

[31] N. e. a. Xuan-mung, "Output Feedback Control for Spacecraft Attitude System with Practical Predefined-Time Stability Based on Anti-Windup Compensator," 2023.

[32] L.-g. Gong, Q. Wang, and C.-y. Dong, "Spacecraft output feedback attitude control based on extended state observer and adaptive dynamic programming," *Journal of the Franklin Institute*, vol. 356, no. 10, pp. 4971–5000, 2019, ISSN: 0016-0032. DOI: 10.1016/j.jfranklin.2019.04.018. [Online]. Available: https://doi.org/10.1016/j.jfranklin.2019.04.018.

[33] D. E. Kirk, *Optimal Control Theory: An Introduction*. New York: Prentice-Hall, Inc., 1970, ISBN: 0-486-43484-2.

[34] R. Tedrake, *Underactuated Robotics, Algorithms for Walking, Running, Swimming, Flying, and Manipulation*. 2023. [Online]. Available: https://underactuated.csail.mit.edu/lqr.html.

[35] A. Guiggiani, I. Kolmanovsky, P. Patrinos, and A. Bemporad, "Constrained Model Predictive Control of Spacecraft Attitude with Reaction Wheels Desaturation," vol. 0, no. 1, pp. 1382–1387, 2015.

[36] H. Bijl, "Nonlinear dynamic inversion," Tech. Rep., 2021. [Online]. Available: https://www.aerostudents.com/courses/advanced-flight-control/advanced-flight-control.php.

[37] A. Perez and H. Moncayo, "Bio-inspired feedback linearized adaptive control for a thrust vectoring free-flyer vehicle," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 2, p. 43, May 2021, ISSN: 1573-0409. DOI: 10.1007/s10846-021-01408-z. [Online]. Available: https://doi.org/10.1007/s10846-021-01408-z.

[38] A. Shekhar, "Review of Model Reference Adaptive Control," pp. 1–5, 2018.

[39] I. D. et al Landau, *Adaptive Control Algorithms, Analysis and Applications*, Second. New York: Springer London Dordrecht Heidelberg New York, 2011, ISBN: 9780857296634. DOI: 10.1007/978-0-85729-664-1.

[40] J. A. Verberne, Y. A. Betancur, K. P. Rivera, N. Coulter, and H. Moncayo, "Comparison of MRAC and L1 adaptive controllers for a gimbaled mini-free flyer," *AIAA Scitech 2019 Forum*, no. January, pp. 1–18, 2019. DOI: 10.2514/6.2019-1290.

[41] S. R. Nekoo, "Tutorial and review on the state-dependent Riccati equation," *Journal of Applied Nonlinear Dynamics*, vol. 8, no. 2, pp. 109–166, 2019, ISSN: 21646473. DOI: 10.5890/JAND.2019.06.001.

[42] S. C. Beeler, "State-Dependent Riccati Equation Regulation of Systems with State and Control Nonlinearities," Tech. Rep., 2004. [Online]. Available: http://www.sti.nasa.gov.

[43] T. A. Gutierrez Martinez, "Health Management and Adaptive Control of Distributed Health Management and Adaptive Control of Distributed Spacecraft Systems Spacecraft Systems," Tech. Rep., 2022. [Online]. Available: https://commons.erau.edu/edt/707.

[44] D. Vrabie, K. Vamvoudakis, and F. Lewis, "Adaptive Optimal Controllers Based on Generalized Policy Iteration in a Continuous-Time Framework," pp. 1402–1409, 2009.

[45] A. K. Pandey, V. Singh, and S. Jain, "Chapter eleven - study and comparative analysis of perturb and observe (p&o) and fuzzy logic based pv-mppt algorithms," in *Applications of AI and IOT in Renewable Energy*, R. N. Shaw, A. Ghosh, S. Mekhilef, and V. E. Balas, Eds., Academic Press, 2022, pp. 193–209, ISBN: 978-0-323-91699-8. DOI: https://doi.org/10.1016/B978-0-323-91699-8.00011-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323916998000115.

[46] Z.-y. Zhao, M. Tomizuka, and S. Isaka, "Fuzzy Gain Scheduling of PID Controllers," vol. 23, no. 5, 1993.

[47] M. Navabi and M. RajabAliFardi, "Quaternion based fuzzy gain scheduled PD law for spacecraft attitude control," *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems, CFIS 2018*, vol. 2018-Janua, no. 3, pp. 149–151, 2018. DOI: 10.1109/CFIS.2018.8336660.

[48] *Mamdani and sugeno fuzzy inference systems*, Accessed: 2023-06-23. [Online]. Available: https://www.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html.

[49] E. H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," pp. 1–13, 1975.

[50] *Defuzzification methods*, Accessed: 2023-06-23. [Online]. Available: https://www.mathworks.com/help/fuzzy/defuzzification-methods.html.

[51] M. Sugeno, *Industrial Applications of Fuzzy Control*. Sole distributors for the U.S.A., 1985.

[52] *Type-2 fuzzy inference systems*, Accessed: 2023-06-23. [Online]. Available: https://www.mathworks.com/help/fuzzy/type-2-fuzzy-inference-systems.html.

[53] J. Yi, N. Yubazaki, and K. Hirota, "A new fuzzy controller for stabilization of parallel-type double inverted pendulum system," vol. 126, pp. 105–119, 2002.

[54] N. Najafizadeh Sari, H. Jahanshahi, and M. Fakoor, "Adaptive Fuzzy PID Control Strategy for Spacecraft Attitude Control," *International Journal of Fuzzy Systems*, vol. 21, no. 3, pp. 769–781, 2019, ISSN: 21993211. DOI: 10.1007/s40815-018-0576-2.

[55] W. G. Dos Santos and E. M. Rocco, "Design of a Fuzzy PID Controller for application in satellite attitude control system," *Workshop on Space Engineering and Technology*, no. June, pp. 1–8, 2012.

[56] M. Navabi, N. S. Hashkavaei, and M. Reyhanoglu, "Satellite attitude control using optimal adaptive and fuzzy controllers," *Acta Astronautica*, vol. 204, no. May 2022, pp. 434–442, 2023, ISSN: 00945765. DOI: 10.1016/j.actaastro.2023.01.005. [Online]. Available: https://doi.org/10.1016/j.actaastro.2023.01.005.

[57] NASA, *Space shuttle roll maneuver*, Accessed: 2023-01-09, 2011. [Online]. Available: www.nasa.gov/pdf/519348main_AP_ST_Phys_RollManeuver.pdf.

[58] M. H. Kaplan, *Modern Spacecraft Dynamics & Control*. Garden City, New York: Dover Publications, 2020.

[59] A. Y. Betancur Vesga, "Development of a Research Spacecraft Test-Bed with Implementation of Control Laws to Compensate Undesired Dynamics," 2019. [Online]. Available: https://commons.erau.edu/edt/482/.

[60] B. Wie, *Space Vehicle Guidance, Control and Astrodynamics*. American Institute of Aeronautics and Astronautics, Inc., Mar. 2015. DOI: 10.2514/4.102752. [Online]. Available: https://doi.org/10.2514/4.102752.

[61] J. Sola, "so3,quaternions,rotation- Quaternion kinematics for the error-state KF," *Laboratoire dAnalyse et dArchitecture des Systemes-Centre national de la recherche scientifique (LAAS-CNRS), Toulouse, France, Tech. Re*, p. 6, 2012. [Online]. Available: http://www.billion.uk.com/downloads/user%20manual/annex_m.pdf.

[62] F. L. Markley and J. L. Crassidis, *Fundamentals of spacecraft attitude determination and control*. 2014, pp. 1–486, ISBN: 9781493908028. DOI: 10.1007/978-1-4939-0802-8.

[63] J. L. Junkins and H. Schaub, *Analytical Mechanics of Space Systems, Second Edition*. American Institute of Aeronautics and Astronautics, Jan. 2009. DOI: 10.2514/4.867231. [Online]. Available: https://doi.org/10.2514/4.867231.

[64] G. P. Sutton and O. Biblarz, *Rocket Propulsion Elements*, 7th ed. New York, 2001, p. 751, ISBN: 0471326429.

[65] I. The MathWorks, *Matlab simulink - calculate inertia tensor*, Accessed: 2023-07-13, Massachusetts, US. [Online]. Available: https://la.mathworks.com/help/aeroblks/estimateinertiatensor.html.

[66] A. Inc., *Autodesk inventor professional*, Accessed: 2023-07-13, California, US. [Online]. Available: https://www.autodesk.com/products/inventor/overview.

[67] N. J. Gogtay and U. M. Thatte, "Principles of correlation analysis," *Journal of Association of Physicians of India*, vol. 65, no. MARCH, pp. 78–81, 2017, ISSN: 00045772.

[68] K. S. University, *Spss tutorials: Pearson correlation*, Accessed: 2023-07-25, Ohio, US. [Online]. Available: https://libguides.library.kent.edu/SPSS/PearsonCorr.

[69] I. The MathWorks, *Correlation coefficients*, Accessed: 2023-07-25, Massachusetts, US. [Online]. Available: https://www.mathworks.com/help/matlab/ref/corrcoef.html.

[70] N. S. Software, *Lin's concordance correlation coefficient*, Accessed: 2023-07-25, Utah, US. [Online]. Available: https://www.ncss.com/wp-content/themes/ncss/pdf/Procedures/PASS/Lins_Concordance_Correlation_Coefficient.pdf.

[71] C. A. E. Nickerson, "A Note On "A Concordance Correlation Coefficient to Evaluate Reproducibility"," *Biometrics*, vol. 53, no. 4, p. 1503, 1997, ISSN: 0006341X. DOI: 10.2307/2533516.

[72] C. Lorenz, *Allfitdist matlab function*, Accessed: 2023-07-25, Massachusetts, US. [Online]. Available: https://gitlab.imk-ifu.kit.edu/c.lorenz/Copula_Bias_Correction/-/blob/580c096a56949d32c59e67e26300aa59e68f9ed5/allfitdist.m.

[73] N. Pandis, "Calculating the P value and carrying out a statistical test," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 148, no. 1, pp. 187–188, 2015, ISSN: 08895406. DOI: 10.1016/j.ajodo.2015.04.009. [Online]. Available: http://dx.doi.org/10.1016/j.ajodo.2015.04.009.

[74] I. The MathWorks, *Xpc target for use with real-time workshop*, English, version Version 12.1, The MathWorks, Inc., 2001, 182 pp., July, 2023.

[75] L. Microstrain, *Imu 3dm-gx3-45®*, Accessed: 2023-07-13, North Carolina, US. [Online]. Available: https://www.microstrain.com/inertial/3dm-gx3-45.

[76] A. B. Younes and D. Mortari, "Derivation of all attitude error governing equations for attitude filtering and control," *Sensors (Switzerland)*, vol. 19, no. 21, pp. 4–6, 2019, ISSN: 14248220. DOI: 10.3390/s19214682.

[77] D. Calvo, T. Avilés, V. Lapuerta, and A. Laverón-Simavilla, "Fuzzy attitude control for a nanosatellite in low Earth orbit," *Expert Systems with Applications*, vol. 58, pp. 102–118, 2016, ISSN: 09574174. DOI: 10.1016/j.eswa.2016.04.004.

[78] A. E. Perez Rocha, "Development of fault tolerant adaptive control laws for aerospace systems," 2016.

[79] J. Bois, *Student-t distribution*, Accessed: 2023-07-25, California, US. [Online]. Available: https://distribution-explorer.github.io/continuous/student_t.html.

APPENDIX

Appendix A. Test flights database

TABLE IX

TEST FLIGHTS DATABASE.

Test	ϕ	θ	ψ	T_{fail}	t_{fail}	ω_{bias}	t_{bias}	$tank_1$	$tank_2$	**Controller**
1	0 15 30 45	0 0 0 0	0 0 0 0	0	0	0 0 0	0	3000	3000	PID
2	0 15 30 45	0 0 0 0	0 0 0 0	0	0	0 0 0	0	3000	3000	PID
3	0 0 15 30 45 45	0 0 0 0 0 0	0 0 0 0 0 0	0	0	0 0 0	0	3000	3000	PID
4	0 0 15 30 45 45	0 0 0 0 0 0	0 0 0 0 0 0	0	0	0 0 0	0	3000	3000	PID
5	0 0 0 30 30	0 15 15 15 15	0 0 0 0 0	1 7	11	0 0 0	0	2100	2500	NLDI+AIS
6	0 0 0 30 30	0 15 15 15 15	0 0 0 0 0	1 8	11	0 0 0	0	1800	2500	NLDI+AIS
7	0 0 20 20	12 12 0 0	0 0 0 0	0	0	15 0 0	11	1800	2800	NLDI+AIS
8*	0 0 20 20	12 12 0 0	0 0 0 0	1 6	11	0 0 0	0	3000	3000	NLDI+AIS
9	0 0 20 20	12 12 0 0	0 0 0 0	2 4	11	0 0 0	0	2700	2000	NLDI+AIS
10*	0 0 20 20	12 12 0 0	0 0 0 0	2 5	11	0 0 0	0	1100	1800	NLDI+AIS
11*	0 0 20 20	12 12 0 0	0 0 0 0	3 8	11	0 0 0	0	800	900	NLDI+AIS
12	0 0 20 20	12 12 0 0	0 0 0 0	4 7	11	0 0 0	0	2600	1800	NLDI+AIS
13	0 30 30 30 30	0 0 0 15 15	0 0 0 0 0	1 7	11	0 0 0	0	2000	2000	NLDI+AIS
14	0 30 30 30 30	0 0 0 15 15	0 0 0 0 0	1 8	11	0 0 0	0	900	1200	NLDI+AIS
15	0 0 45 45 45 75 75 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0	0	0 0 0	0	3000	3000	DRL
16	0 0 45 45 45 75 75 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0	0	0 0 0	0	3000	3000	DRL
17	0 0 45 45 45 75 75 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0	0	0 0 0	0	3000	3000	DRL
18	0 0 45 45 45 75 75 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	1	10	0 0 0	0	3000	3000	DRL
19	0 45 45 45 45	0 0 0 0 0	0 0 0 0 0	1 7	11	0 0 0	0	3000	3000	NLDI+AIS
20	0 45 45 45 45	0 0 0 0 0	0 0 0 0 0	1 8	11	0 0 0	0	3000	3000	NLDI+AIS
21	0 0 45 45	15 15 0 0	0 0 0 0	0	0	0 0 0	0	3000	3000	NLDI
22	0 0 45 45	15 15 0 0	0 0 0 0	0	0	0 0 0	0	2000	2100	NLDI
23	0 0 45 45	15 15 0 0	0 0 0 0	0	0	0 0 0	0	3000	3000	NLDI
24	0 0 45 45	15 15 0 0	0 0 0 0	0	0	0 0 0	0	3000	3000	NLDI+AIS
25	0 0 45 45	15 15 0 0	0 0 0 0	0	0	0 0 0	0	3000	3000	NLDI+AIS
26	0 0 45 45	15 15 0 0	0 0 0 0	0	0	0 0 0	0	2000	2000	NLDI+AIS
27	0 0 45 45	15 15 0 0	0 0 0 0	0	21	0 0 0	0	2200	2200	Fuzzy
28	0 0 45 45	15 15 0 0	0 0 0 0	0	21	0 0 0	0	3000	3000	Fuzzy
29	0 0 45 45	15 15 0 0	0 0 0 0	0	21	0 0 0	0	1500	1800	Fuzzy
30	0 0 45 45	15 15 0 0	0 0 0 0	1 2	21	0 0 0	0	1100	1200	NLDI
31	0 0 45 45	15 15 0 0	0 0 0 0	1 2	21	0 0 0	0	3000	3000	NLDI
32	0 0 45 45	15 15 0 0	0 0 0 0	1 2	21	0 0 0	0	1800	1500	NLDI
33	0 0 45 45	15 15 0 0	0 0 0 0	1 2	21	0 0 0	0	1500	1800	NLDI+AIS
34	0 0 45 45	15 15 0 0	0 0 0 0	1 2	21	0 0 0	0	3000	3000	NLDI+AIS
35	0 0 45 45	15 15 0 0	0 0 0 0	1 2	21	0 0 0	0	2000	2000	NLDI+AIS

36	0 0 45 45	15 15 0 0	0 0 0 0	1 2	21	0 0 0	0	3000	3000	Fuzzy
37	0 0 45 45	15 15 0 0	0 0 0 0	1 2	21	0 0 0	0	1400	1400	Fuzzy
38	0 0 45 45	15 15 0 0	0 0 0 0	1 2	21	0 0 0	0	3000	3000	Fuzzy
39	0 0 45 45	15 15 0 0	0 0 0 0	7 8	21	0 0 0	0	3000	3000	NLDI
40	0 0 45 45	15 15 0 0	0 0 0 0	7 8	21	0 0 0	0	3000	3000	NLDI
41	0 0 45 45	15 15 0 0	0 0 0 0	7 8	21	0 0 0	0	2000	1500	NLDI
42	0 0 45 45	15 15 0 0	0 0 0 0	7 8	21	0 0 0	0	3000	3000	NLDI+AIS
43	0 0 45 45	15 15 0 0	0 0 0 0	7 8	21	0 0 0	0	1800	2100	NLDI+AIS
44	0 0 45 45	15 15 0 0	0 0 0 0	7 8	21	0 0 0	0	3000	3000	NLDI+AIS
45	0 0 45 45	15 15 0 0	0 0 0 0	7 8	21	0 0 0	0	1800	1800	Fuzzy
46	0 0 45 45	15 15 0 0	0 0 0 0	7 8	21	0 0 0	0	3000	3000	Fuzzy
47	0 0 45 45	15 15 0 0	0 0 0 0	7 8	21	0 0 0	0	1800	1800	Fuzzy
48	20 0 0 15	0 5 5 0	0 0 0 0	0	0	15 0 0	11	1000	2000	NLDI+AIS
49	20 0 0 15	0 5 5 0	0 0 0 0	2 4	11	0 0 0	0	1800	1500	NLDI+AIS
50	20 0 0 15	0 5 5 0	0 0 0 0	2 7	11	0 0 0	0	1000	1900	NLDI+AIS
51	20 0 0 15	0 5 5 0	0 0 0 0	2 8	11	0 0 0	0	3000	3000	NLDI+AIS
52	0 25 25 25	0 0 0 0	0 0 0 0	0	0	0 0 0	0	1000	2500	NLDI+AIS$_{v_1}$
53	0 25 25 25	0 0 0 0	0 0 0 0	1	0	0 0 0	0	1000	2400	NLDI+AIS$_{v_1}$
54	0 25 25 25	0 0 0 0	0 0 0 0	1 2	11	0 0 0	0	1900	1500	NLDI+AIS
55	0 25 25 25	0 0 0 0	0 0 0 0	2	0	0 0 0	0	900	2200	NLDI+AIS$_{v_1}$
56	0 25 25 25	0 0 0 0	0 0 0 0	2	11	0 0 0	0	400	2000	NLDI+AIS$_{v_1}$
57	0 25 25 25	0 0 0 0	0 0 0 0	2	11	0 0 0	0	3000	3000	NLDI+AIS$_{v_1}$
58	0 25 25 25	0 0 0 0	0 0 0 0	5	25	0 0 0	0	2400	2000	NLDI+AIS
59	0 28 28 28	0 0 0 0	0 0 0 0	0	0	0 0 0	0	3000	3000	PID
60	0 28 28 28	0 0 0 0	0 0 0 0	1	0	0 0 0	0	2500	2100	NLDI+AIS
61	0 28 28 28	0 0 0 0	0 0 0 0	1 2	11	0 0 0	0	1900	2800	NLDI+AIS
62	0 28 28 28	0 0 0 0	0 0 0 0	2	0	0 0 0	0	2500	1800	NLDI+AIS
63	0 28 28 28	0 0 0 0	0 0 0 0	2	11	0 0 0	0	2500	1500	NLDI+AIS
64	0 28 28 28	0 0 0 0	0 0 0 0	5	25	0 0 0	0	3000	3000	NLDI+AIS
65	0 30 30 30	0 0 0 0	0 0 0 0	0	0	0 0 0	0	1000	2000	NLDI+AIS
66	0 30 30 30	0 0 0 0	0 0 0 0	0	0	0 0 0	0	1000	1800	NLDI+AIS
67	0 30 30 30	0 0 0 0	0 0 0 0	0	0	0 0 0	0	3000	3000	NLDI+AIS
68	0 30 30 30	0 0 0 0	0 0 0 0	1	0	0 0 0	0	3000	3000	NLDI+AIS$_{v_1}$
69	0 30 30 30	0 0 0 0	0 0 0 0	1	0	0 0 0	0	2200	2500	NLDI+AIS$_{v_1}$
70	0 30 30 30	0 0 0 0	0 0 0 0	1 2	11	0 0 0	0	3000	3000	NLDI+AIS$_{v_1}$
71	0 30 30 30	0 0 0 0	0 0 0 0	1 3	11	0 0 0	0	3000	3000	NLDI+AIS
72	0 30 30 30	0 0 0 0	0 0 0 0	1 7	11	0 0 0	0	2500	2100	NLDI+AIS
73	0 30 30 30	0 0 0 0	0 0 0 0	1 8	11	0 0 0	0	3000	3000	NLDI+AIS
74	0 30 30 30	0 0 0 0	0 0 0 0	2	0	0 0 0	0	1500	2000	NLDI+AIS
75	0 30 30 30	0 0 0 0	0 0 0 0	2	0	0 0 0	0	1000	1400	NLDI+AIS
76	0 30 30 30	0 0 0 0	0 0 0 0	2	11	0 0 0	0	1000	1200	NLDI+AIS
77	0 30 30 30	0 0 0 0	0 0 0 0	2	11	0 0 0	0	800	1000	NLDI+AIS
78	0 30 30 30	0 0 0 0	0 0 0 0	2	11	0 0 0	0	3000	3000	NLDI+AIS
79	0 30 30 30	0 0 0 0	0 0 0 0	2 4	11	0 0 0	0	1000	1000	NLDI+AIS

80	0 30 30 30	0 0 0 0	0 0 0 0	2 7	11	0 0 0	0	2100	2000	NLDI+AIS
81	0 30 30 30	0 0 0 0	0 0 0 0	2 8	11	0 0 0	0	1500	1000	NLDI+AIS
82	0 30 30 30	0 0 0 0	0 0 0 0	5	25	0 0 0	0	3000	3000	NLDI+AIS
83	0 32 32 32	0 0 0 0	0 0 0 0	0	0	0 0 0	0	2000	2000	NLDI+AIS$_{v_1}$
84	0 32 32 32	0 0 0 0	0 0 0 0	0	0	0 0 0	0	2000	2000	NLDI+AIS$_{v_1}$
85	0 32 32 32	0 0 0 0	0 0 0 0	1	0	0 0 0	0	3000	3000	NLDI+AIS
86	0 32 32 32	0 0 0 0	0 0 0 0	1	0	0 0 0	0	2500	2500	NLDI+AIS
87	0 32 32 32	0 0 0 0	0 0 0 0	1	11	0 0 0	0	2000	2000	NLDI+AIS$_{v_1}$
88	0 32 32 32	0 0 0 0	0 0 0 0	2	0	0 0 0	0	2200	2400	NLDI+AIS
89	0 32 32 32	0 0 0 0	0 0 0 0	2	0	0 0 0	0	1500	2300	NLDI+AIS
90	0 32 32 32	0 0 0 0	0 0 0 0	2	11	0 0 0	0	1300	2000	NLDI+AIS
91	0 32 32 32	0 0 0 0	0 0 0 0	2	11	0 0 0	0	1000	2000	NLDI+AIS
92	0 32 32 32	0 0 0 0	0 0 0 0	5	25	0 0 0	0	1100	2000	NLDI+AIS
93	0 35 35 35	0 0 0 0	0 0 0 0	0	0	0 0 0	0	3000	3000	PID
94	0 35 35 35	0 0 0 0	0 0 0 0	0	0	0 0 0	0	2000	2000	PID
95	0 35 35 35	0 0 0 0	0 0 0 0	1	0	0 0 0	0	2000	2000	PID
96	0 35 35 35	0 0 0 0	0 0 0 0	1	0	0 0 0	0	2000	2000	PID
97	0 35 35 35	0 0 0 0	0 0 0 0	2	0	0 0 0	0	1800	1500	NLDI+AIS
98	0 35 35 35	0 0 0 0	0 0 0 0	2	0	0 0 0	0	1500	1500	NLDI+AIS
99	0 35 35 35	0 0 0 0	0 0 0 0	2	11	0 0 0	0	3000	3000	NLDI+AIS$_{v_1}$
100	0 35 35 35	0 0 0 0	0 0 0 0	2	11	0 0 0	0	1500	1300	NLDI+AIS$_{v_1}$
101	40 40 0 0	0 0 10 10	0 0 0 0	0	0	15 0 0	11	3000	3000	NLDI+AIS
102	40 40 0 0	0 0 10 10	0 0 0 0	0	0	20 0 0	0	2000	600	NLDI+AIS
103	40 40 0 0	0 0 10 10	0 0 0 0	0	0	7 0 0	0	2100	1000	NLDI+AIS
104*	40 40 0 0	0 0 10 10	0 0 0 0	1 6	11	0 0 0	0	1000	1100	NLDI+AIS
105*	40 40 0 0	0 0 10 10	0 0 0 0	2 5	11	0 0 0	0	1700	2100	NLDI+AIS
106	40 40 0 0	0 0 10 10	0 0 0 0	2 7	11	0 0 0	0	3000	3000	NLDI+AIS
107	40 40 0 0	0 0 10 10	0 0 0 0	2 8	11	0 0 0	0	700	1700	NLDI+AIS
108*	40 40 0 0	0 0 10 10	0 0 0 0	3 8	11	0 0 0	0	900	1100	NLDI+AIS
109*	40 40 0 0	0 0 10 10	0 0 0 0	4 7	11	0 0 0	0	3000	3000	NLDI+AIS

*: Taken as an outlier due to high uncertainties in flight test or simulation divergence.