POLITECNICO DI TORINO
Master Degree course in Mechatronic Engineering

Master Degree Thesis

Multi-spacecraft sliding mode control with hybrid dynamics using Dual-Gimbal Variable-Speed CMGs with integrated Fault Detection and Isolation systems

Supervisors
Prof. Mauro Mancini
Prof. Fausto Francesco Lizzio

Candidate
Chiara Cristofaro

Academic Year 2024-2025

Abstract

This thesis presents the modeling, control and fault-resilient coordination of a network of spacecraft with hybrid dynamics using a Multi-Sliding Mode Control (MSMC) strategy. Each spacecraft is described by a nonlinear attitude dynamic model that accounts for both rigid-body motion and flexible structural dynamics, allowing a more realistic representation of appendages such as antennas and flexible panels.

A distributed consensus protocol is embedded within the control architecture through a Laplacian-based sliding surface, enabling cooperative attitude regulation and synchronization among multiple agents under a predefined communication topology.

The actuation system of each spacecraft consists of parallel-mounted Dual-Gimbal Variable-Speed Control Moment Gyros (DGVSCMGs), whose electromechanical dynamics are modeled from first principles using a first-order approximation. Each motor and gimbal axis is controlled via dedicated PID controllers, while a nonlinear steering law determines the actuator configuration and the corresponding reference commands. In particular, the steering strategy allows the system to operate either in a Control Moment Gyro-based configuration or in a Reaction Wheel-based configuration, depending on the proximity to singular actuator conditions.

To enhance the reliability and autonomy of the overall system, an onboard Fault Detection and Isolation (FDI) module is integrated at actuator level. The proposed architecture implements a Single Failure Tolerant strategy: when a failure is detected and isolated in a nominal actuator, a redundant unit is activated to preserve control performance. If the redundant actuator also fails, the communication topology is automatically reconfigured and the faulty spacecraft is excluded from the network, ensuring the stability and robustness of the remaining formation.

Simulation results demonstrate the effectiveness of the proposed framework in terms of attitude regulation, consensus achievement and fault tolerance, highlighting the capabilities of the MSMC architecture in managing actuator faults and network reconfiguration while maintaining satisfactory performance.

Contents

1 Introduction 1

2 Satellite model 5
- 2.1 Rigid dynamic model . . . 6
- 2.2 Hybrid dynamics . . . 6
- 2.3 kinematics . . . 8

3 Consensus protocol 11
- 3.1 Graph teory . . . 11
  - 3.1.1 Directed Graph . . . 13
  - 3.1.2 Undirected graph . . . 14
- 3.2 Type of consensus . . . 15
  - 3.2.1 Single-integrator consensus . . . 16
  - 3.2.2 Double-integrator consensus . . . 16
  - 3.2.3 Non-linear consensus . . . 17

4 Dual Gimbal Variable Speed Control Moment Gyros 19
- 4.1 Control Moment Gyroscopes . . . 19
- 4.2 CMG classification . . . 20
- 4.3 Electromechanical system . . . 21
  - 4.3.1 Electrical subsystem . . . 22
  - 4.3.2 Mechanical subsystem . . . 22
  - 4.3.3 State-space representation and transfer function . . . 22
- 4.4 PID controller . . . 24
- 4.5 Steering Law . . . 28
  - 4.5.1 Singularities identification . . . 31
  - 4.5.2 Gimbal commands . . . 32
  - 4.5.3 Flywheel commands . . . 34
  - 4.5.4 Steering law testing . . . 35

5 Actuator Fault Detection and Isolation 45
- 5.1 Fault causes of momentum exchange devices . . . 46
- 5.2 Analysis of Fault Management Requirements . . . 47
- 5.3 Extension of Nominal Equipment Set . . . 48

5.4 Definition and Implementation of FDI Concept 49
5.4.1 Residuals' computation 50

6 Multi Sliding Mode Control 53
6.1 Sliding Mode Control 53
6.1.1 Sliding Mode Control Law 53
6.1.2 Design of the sliding surface 54
6.1.3 SMC design 54
6.1.4 Chattering and mitigation techniques 55
6.1.5 Types of SMC 56
6.2 Multi-Spacecraft Sliding Mode Control 56
6.2.1 Multispacecraft Sliding Surface Design 57
6.2.2 Pure Consensus 58
6.2.3 Reference Tracking 58
6.2.4 Control Law 59
6.2.5 Simulation Results 60

7 Conclusions 83

Bibliography 85

Chapter 1

Introduction

This chapter introduces the main topics addressed throughout this thesis. Since the core objective of this work is the development of a cooperative attitude control system for a multi-spacecraft formation, the role and importance of attitude control in modern space missions are first discussed. Subsequently, the fundamental concepts related to distributed control, consensus protocols and fault-tolerant actuation are presented, highlighting the challenges faced by control designers in multi-agent spacecraft systems. Finally, the structure of the thesis is outlined.

The increasing demand for autonomous, cooperative and scalable space missions has led to a growing interest in multi-spacecraft systems, such as satellite constellations and formation flying missions. These architectures are widely employed in Earth observation, deep-space exploration and on-orbit servicing, with recent industrial examples including large-scale satellite constellations such as Starlink and OneWeb. The latter constellation is presented in [14].

Figure 1.1. Starlink network seen from Earth.

Unlike traditional single-spacecraft missions, these scenarios require advanced control strategies capable of handling network topology variability, communication constraints and dynamic interactions among multiple platforms.

A notable historical example of a multi-spacecraft scientific mission is ESA's *Cluster* mission, presented in [12], consisting of four identical spacecraft flying in a tetrahedral formation around Earth. The Cluster satellites (Rumba, Salsa, Samba, and Tango) were launched in 2000 and designed to study the Earth's magnetosphere through distributed measurements, enabling the separation of spatial and temporal variations of plasma phenomena.

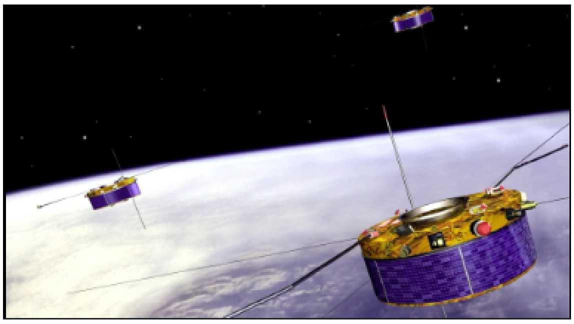

Figure 1.2. ESA Cluster mission.

More recently, cooperative missions such as NASA's Magnetospheric Multiscale (MMS) mission, which formation and objective are presented in [2], have further demonstrated the potential of distributed space systems.

Figure 1.3. NASA's Magnetospheric Multiscale (MMS) mission.

Within this context, this thesis proposes a *Multi-Spacecraft Sliding Mode Control* strategy for a network of four spacecrafts, each modeled as a hybrid-dynamics system. The overall attitude control system is realized within a distributed *Attitude and Orbit Control System* (AOCS) framework and actuated through *Dual Gimbal Variable Speed Control Moment Gyros* (DGVSCMGs), supervised by integrated *Fault Detection and Isolation* (FDI) subsystems.

Modern spacecraft rely on sophisticated *Attitude and Orbit Control Systems* (AOCS) to stabilize the vehicle, perform reorientation maneuvers, reject disturbances, and guarantee precise pointing for payload and communication requirements. At its core, an AOCS is built around the *Guidance, Navigation and Control* (GNC) architecture, which is decomposed into three main blocks:

- *Guidance*: generates reference commands such as desired attitude and angular velocity, either off-line or online (i.e., as function of the current state of the system);
- *Navigation*: reconstructs the spacecraft state from onboard sensor measurements using state estimation algorithms (e.g., Kalman filtering);
- *Control*: computes the control inputs required to drive the spacecraft dynamics toward the reference behavior.

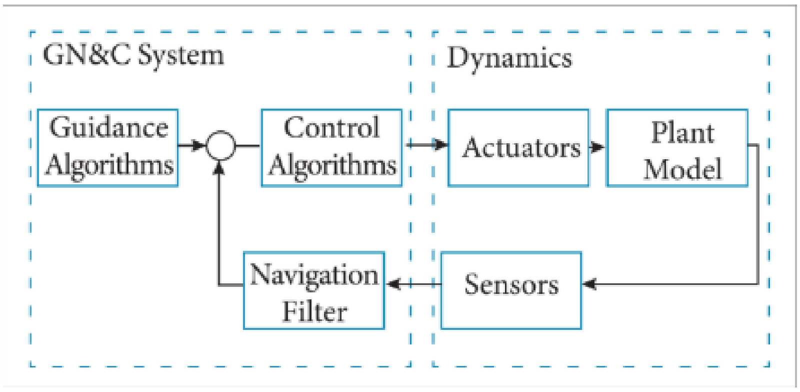

Figure 1.4. GNC system representation.

Beyond the GNC, the AOCS also integrates actuators, sensors, onboard processors and fault management logic.

In cooperative multi-agent systems, *consensus protocols* represent a fundamental tool to achieve coordinated behavior through local information exchange. The theory of consensus was formalized in the early 2000s through seminal works by Olfati-Saber, Fax, Murray and Ren, among others [16]. In such frameworks, the communication structure among agents is modeled using graph theory and represented through the Laplacian matrix, whose properties strongly influence the convergence and stability of the network.

In this work, consensus concepts are embedded into the sliding surface design of the proposed controller by incorporating a Laplacian-based coupling structure into the sliding manifold, according to [22], [23], [21].

Sliding Mode Control (SMC), largely presented in [20], is a nonlinear control technique widely recognized for its robustness against parametric uncertainties and external disturbances. Its suitability for space systems has been demonstrated in several attitude control applications involving spacecraft and formation flying missions.

Each spacecraft in the formation is equipped with two nominal DGVSCMGs and a third redundant unit. DGVSCMGs are commonly adopted in high-agility attitude control systems due to their ability to generate large torques without propellant consumption. Their application can be found in the International Space Station (ISS) and in various Earth observation satellites.

The actuators are controlled via dedicated PID controllers for each gimbal and motor axis. A steering law is implemented to switch between a Control Moment Gyro-based configuration and a Reaction Wheel-based configuration when operating near singular conditions, according to [4].

To enhance reliability and autonomy, each spacecraft integrates an onboard Fault Detection and Isolation (FDI) system. FDI techniques are crucial in modern space systems to enable fault tolerance and system reconfiguration in case of failures. In this work, a Single Failure Tolerant (SFT) strategy is implemented at the actuator level.

Chapter 2 presents the mathematical modeling of the spacecraft dynamics, considering both rigid-body and flexible dynamics. The coupling between the two is described through a dedicated coupling matrix δ, while the flexible dynamics parameters are obtained from [17].

Chapter 3 introduces the concept of consensus in multi-agent systems, presenting both linear and nonlinear consensus strategies and their application to spacecraft networks.

Chapter 4 focuses on the modeling and control of the DGVSCMG actuation system, including the electromechanical first-order modeling, PID design and testing for each axis of the actuators, and the steering law design and testing for singularity management.

Chapter 5 presents the possible fault causes of momentum exchange devices, as reported in [25], describes the integrated Fault Detection and Isolation architecture, discussing fault types affecting momentum exchange devices and the implementation of the actuator-level FDI system, according to [1].

Chapter 6 is devoted to the design of the Multi-Spacecraft Sliding Mode Controller, including theoretical background on SMC and the proposed MSMC formulation. The results of the numerical simulations performed in Matlab/Simulink environment are presented for regulation, tracking and pure consensus scenarios. The regulation problem is simulated with and without the FDI system activation to highlight the effects of this module.

Finally, Chapter 7 summarizes the main conclusions of the work and discusses possible directions for future research.

Chapter 2

Satellite model

The satellite model (in the controlled system, *the plant*), is composed by four subsystems:

1. **hybrid attitude dynamics** of the spacecraft, that is the result of the coupling between rigid and flexible dynamics;
2. **kinematics**, to compute the quaternion as function of the angular velocity;
3. **Euler function**, to compute the Euler angles knowing the quaternion that defines the current orientation;
4. **flexibility dynamics**: modal coordinates' computation block.

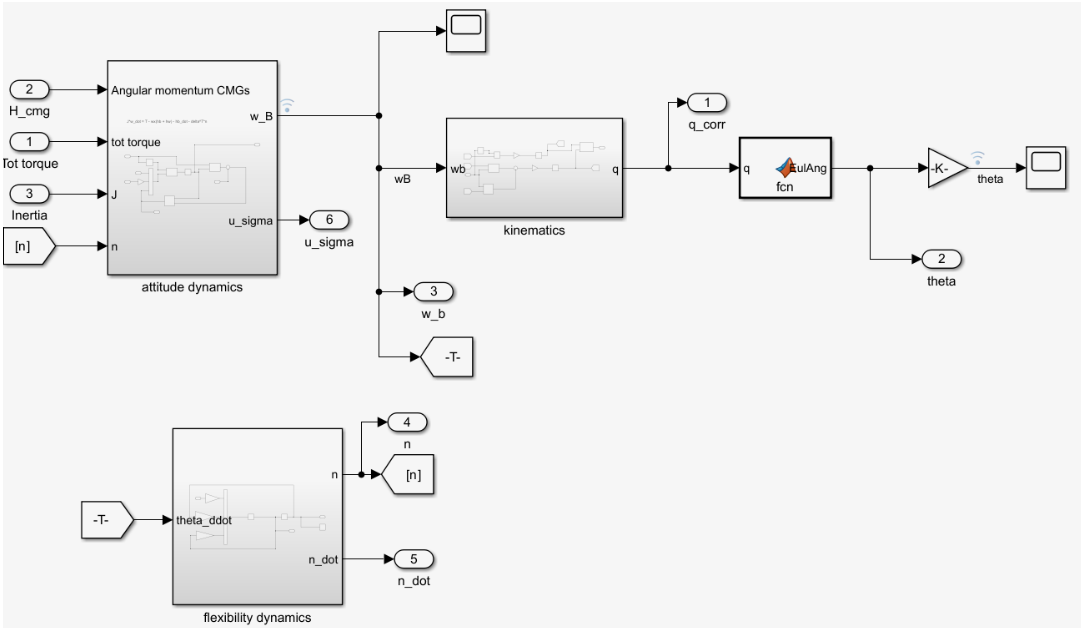

Figure 2.1. Implemented satellite model.

2.1 Rigid dynamic model

Assuming that the spacecraft is rigid, the rotational dynamics of the body is modeled exploiting the Euler equation:

$$\dot{\omega}_B = I^{-1}(\tau - \omega_B \times (I_B w_B + I_{RW} * w_{RW}))$$

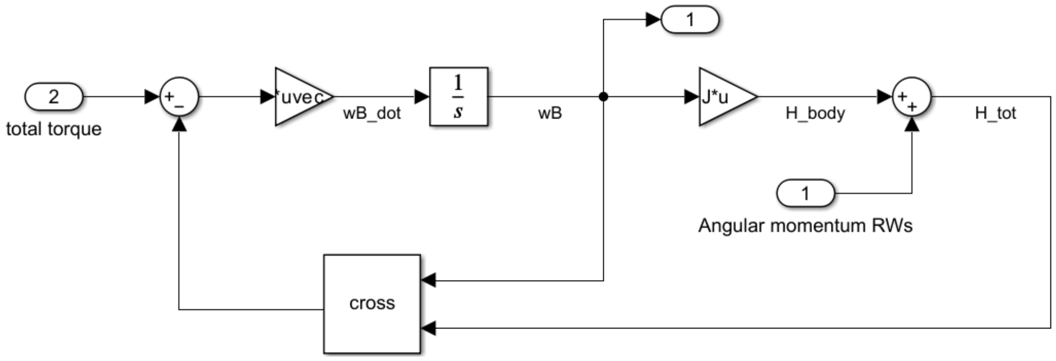

Figure 2.2. Implemented Euler equation.

where τ is the total torque that acts on the satellite, that is composed by the actuators' torques and the external torques (in control theory, the latter are defined as *disturbances* acting on the plant). Moreover, the total angular momentum is given by the sum of the angular momentum of the spacecraft itself and of the actuators: the latter are *momentum exchange devices*, hence, the total angular momentum is constant and they generate internal torques (i.e., when the actuator applies a torque to the satellite, it applies an equal and opposite reaction torque to the actuator, in accordance with Newton's third law).

2.2 Hybrid dynamics

A more realistic assumption for the dynamic analysis of the plant is to consider it as the combination of the parts that can be considered as rigid, and other parts that must be studied and modeled as flexible. To study the latter, one must study and compute the natural frequencies and modes of the flexible parts. Considering a steady spacecraft, the equation to consider to compute the natural frequencies is the dynamic equation of the damped system:

$$\ddot{\eta}(t) + C \cdot \dot{\eta}_i(t) + K \cdot \eta(t) = 0 \tag{2.1}$$

where $\eta \in \mathbb{R}^m$ are the modal coordinates and m is the number of natural modes. The matrices C and $K \in \mathbb{R}^{m \times m}$, are the *mass-normalized* damping and stiffness matrices, respectively. The latter are positive and symmetric matrices computed as:

$$C = diag([2 \cdot \zeta \cdot w_n]) \tag{2.2}$$

$$K = diag([w_n^2]) \tag{2.3}$$

where ζ is a 4x1 vector of the damping coefficient of each considered element of the flexible part, w_n is a 4x1 vector of the natural frequencies. In this work, the natural frequencies and damping ratios of the flexible appendages are not derived analytically. Since the flexible dynamics are included only to capture their effect on the attitude control problem, the modal shapes are not explicitly required, and only the modal parameters w_n and ζ are taken from [17].

Equation (2.1) describes the free response of the damped system. The angular acceleration of the rigid core of the spacecraft imprints a certain torque on the flexible parts: the rigid dynamics and the flexible dynamics are coupled through a coupling matrix δ.

$$\ddot{\eta}(t) + C \cdot \dot{\eta}_i(t) + K \cdot \eta(t) = -\delta \cdot \dot{w}_B(t) \tag{2.4}$$

where $\dot{w}$ is the angular acceleration of the rigid core of the spacecraft.

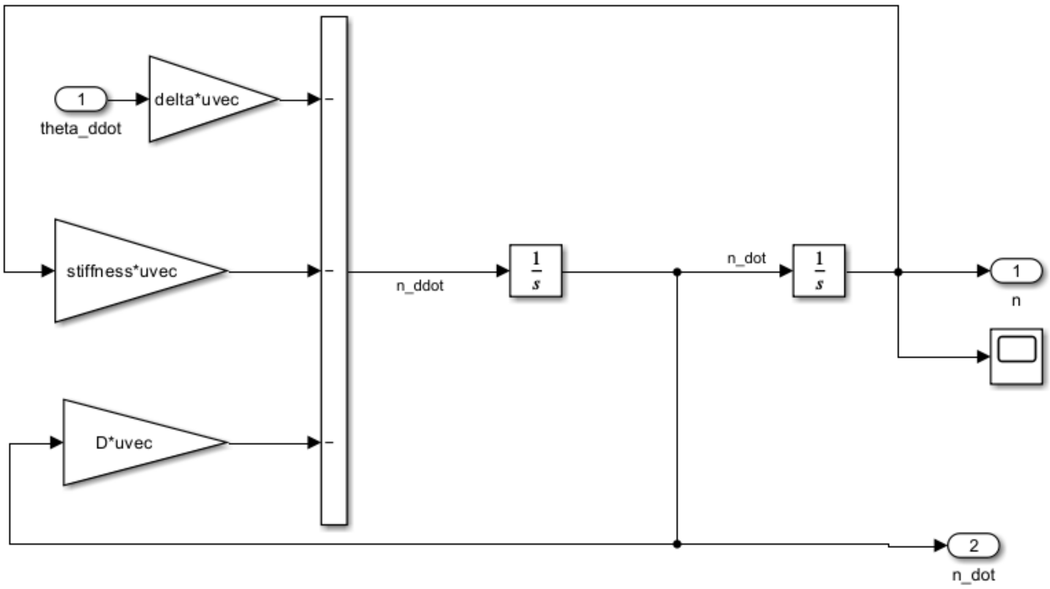

Figure 2.3. Flexibility dynamics model implemented.

Likewise, the rigid dynamics is influenced by the modal coordinates' acceleration:

$$J\dot{w}_B(t) = \tau - w_B(t) \times (H_B + H_{CMG}) - \delta^T \cdot \ddot{\eta}(t) \tag{2.5}$$

The hybrid dynamics model implementation is reported in Figure (2.4):

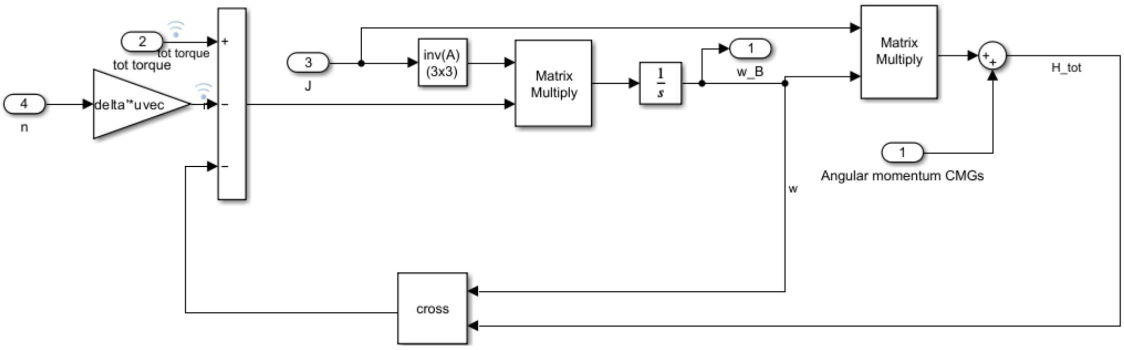

Figure 2.4. Hybrid dynamics model implemented.

Parameter	Symbol	Value	Unit of measure
damping coefficients' vector	ζ	[0.05607 0.08620 0.01283 0.02516]	-
natural frequencies' vector	w_n	[0.7681 1.1038 1.8733 2.5496]	$[rad/s]$
coupling matrix	δ	$\begin{bmatrix} 6.4564 & 1.2781 & 2.1563 \\ -1.2562 & 0.9176 & -1.6726 \\ 1.1169 & 2.4890 & -0.8367 \\ 1.2364 & -2.6581 & -1.1250 \end{bmatrix}$	-
spacecraft inertia	J	$\begin{bmatrix} 350 & 3 & 4 \\ 3 & 280 & 10 \\ 4 & 10 & 190 \end{bmatrix}$	-

Table 2.1. Hybrid dynamics parameters.

2.3 kinematics

The kinematic equations have been implemented. The output of this subsystem is the current quaternion that describes the rotation from the initial body reference frame to the current one. It has been computed as function of the angular velocity of the body. Considering the quaternion kinematic equation:

$$\dot{q} = \frac{1}{2}\Omega(w) \cdot q \tag{2.6}$$

where q is the quaternion, $\dot{q}$ is the time derivative of the quaternion and Ω is the skew-symmetric matrix of the angular velocity's components.

$$\Omega = \begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix} \tag{2.7}$$

Exploiting the inverse relation of Eq. (2.6):

$$q = 2\Omega^{-1}(w) \cdot \dot{q} \tag{2.8}$$

the model has been implemented in Simulink, taking advantage of the environment's capability to handle nonlinear dynamic systems. In particular, since the relation between the quaternion derivative and the angular velocity is nonlinear, the quaternion evolution can be obtained by integrating the corresponding kinematic relation through integrator blocks.

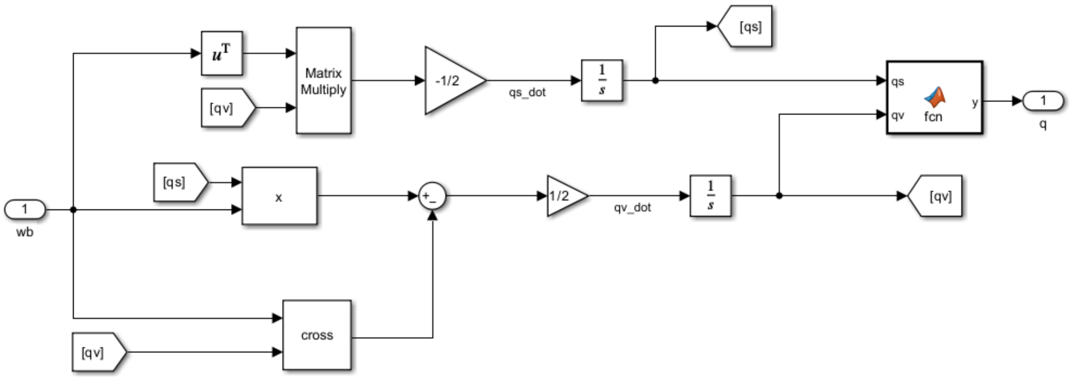

Figure 2.5. kinematic model implementation.

The initial conditions of the integrator blocks are computed as function of the Euler angles that describes the initial attitude of the body frame with respect to the inertial frame, according to the conversion equations of the chosen Euler sequence to describe the rotation. In this implementation, the rotation from the inertial frame to the initial body frame is a 3-2-1 sequence, that is:

- a rotation of an angle ϕ around the inertial frame's Z-axis;
- a rotation of an angle θ around the current frame's Y-axis;
- a rotation of an angle ψ around the current frame's X-axis;

The corresponding initial quaternion is computed as follows:

$$q_0 = \begin{bmatrix} cos(\phi/2) \cdot cos(\theta/2) \cdot cos(\psi/2) - sin(\phi/2) \cdot sin(\theta/2) \cdot sin(\psi/2) \\ sin(\phi/2) \cdot cos(\theta/2) \cdot cos(\psi/2) + cos(\phi/2) \cdot sin(\theta/2) \cdot sin(\psi/2) \\ cos(\phi/2) \cdot sin(\theta/2) \cdot cos(\psi/2) - sin(\phi/2) \cdot cos(\theta/2) \cdot sin(\psi/2) \\ cos(\phi/2) \cdot cos(\theta/2) \cdot sin(\psi/2) + sin(\phi/2) \cdot sin(\theta/2) \cdot cos(\psi/2) \end{bmatrix} \tag{2.9}$$

In Tables (2.2), (2.3), (2.4), (2.5), the kinematic initial conditions of each spacecraft of the communication topology are reported.

Parameter	Symbol	Value	Unit of measure
angle of rotation around Z-axis	ϕ	45	$[deg]$
angle of rotation around Y-axis	θ	15	$[deg]$
angle of rotation around X-axis	ψ	30	$[deg]$
Initial quaternion	q_0	[0.8718;0.3977;0.0183;0.2853]	-

Table 2.2. Kinematic initial conditions of 1st S/C.

Parameter	Symbol	Value	Unit of measure
angle of rotation around Z-axis	ϕ	15	$[deg]$
angle of rotation around Y-axis	θ	5	$[deg]$
angle of rotation around X-axis	ψ	20	$[deg]$
Initial quaternion	q_0	[0.975;0.13593;0.019945;0.17761]	-

Table 2.3. Kinematic initial conditions of 2nd S/C.

Parameter	Symbol	Value	Unit of measure
angle of rotation around Z-axis	ϕ	45	$[deg]$
angle of rotation around Y-axis	θ	15	$[deg]$
angle of rotation around X-axis	ψ	30	$[deg]$
Initial quaternion	q_0	[0.8718;0.3977;0.0183;0.2853]	-

Table 2.4. Kinematic initial conditions of 3rd S/C.

Parameter	Symbol	Value	Unit of measure
angle of rotation around Z-axis	ϕ	10	$[deg]$
angle of rotation around Y-axis	θ	12	$[deg]$
angle of rotation around X-axis	ψ	5	$[deg]$
Initial quaternion	q_0	[0.90814;0.1532;0.030072;0.38848]	-

Table 2.5. Kinematic initial conditions of 4th S/C.

Chapter 3

Consensus protocol

A consensus protocol is a formal plan for cooperation among agents of a network that aims at asymptotic agreement on the value of a variable of interest. It's trivial to notice that a necessary condition for cooperation is to exchange informations. A consensus protocol exploits local information sharing, rather than an all-to-all communication.

The consensus protocol of multi-agent systems provides

- redundancy against a Single Point of Failure: even if one of the agents is in abnormal conditions, the network is still performing its tasks;
- a more versatile deployment of the available resources;
- scalability while handling large-scale problems: the same algorithm can be used for a number N of agents.

Each spacecraft exchanges informations with other spacecrafts according to the network shape, that is the **communication topology** (from Greek *topos*, shape), that is defined through a graph.

3.1 Graph teory

Assuming the network is composed by a number of agents N, a graph G_N of order N is defined as the set $G_N = \{V, E\}$, where

- $V = \{v_1, ..., v_N\}$ is the set of nodes of the network, where each node is an agent;
- $E = \{e_{ij} = (v_i, v_j) | v_i, v_j \in V,\ i \neq j\} \in V \times V$ is an ordered set of edges, that are the links between the agents.

According to the definition, the edge $e_{ij} \in E$ if agent i receives information from agent j. Considering a communication network, an agent that receives information transmitted by another agent is called a **neighbor**. The set of neighbors of agent i is defined as

$$N_i = \{j \mid v_j \in V, e_{ij} \in E\} \tag{3.1}$$

The *interaction parameter* a_{ij} of a network is computed such that:

$$a_{ij} = 1 \text{ if } e_{ij} \in E, a_{ij} = 0 \text{ if } e_{ij} \notin E \tag{3.2}$$

The indegree of a node is computed as:

$$\text{Indegree} = \sum_{j \in N_i} a_{ij} \tag{3.3}$$

The outdegree of a node is computed as:

$$\text{Outdegree} = \sum_{j \in N_i} a_{ji} \tag{3.4}$$

For better understanding, an example is provided: for the topology communication in Figure (3.1)

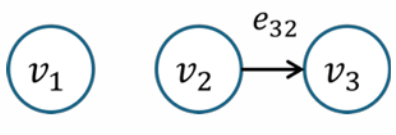

Figure 3.1. Three-agent network.

the set of nodes is $V = \{v_1, v_2, v_3\}$, the set of edges is $E = \{e_{32}\}$, the set of neighbors of each agent is $N_1 = N_2 = 0$, $N_3 = \{v_2\}$. The indegree and outdegree of each agent, computed according to Eq. (3.3) and Eq. (3.4) respectively, are reported in Table (3.1)

Agent	Indegree	Outdegree
1	0	0
2	0	1
3	1	0

Table 3.1. Indegrees and outdegrees of agents of Figure (3.1).

When the outdegree and the indegree of a node i in a graph are equal, it is said to be **balanced**. A mathematical representation of a graph is the Laplacian matrix L $\in \mathbf{R}^{N \times N}$, that is such that

$$l_{ii} = |N_i|, \; l_{ij} = -a_{ij} \tag{3.5}$$

One can observe that, by construction :

- L is a zero-row sum, i.e., $L \cdot 1_N = 0$, where 1_N is the right eigenvector associated to the eigenvalue 0;
- L is diagonally dominant, i.e., $R\{\lambda_i\} \geq 0$, where λ_i is the i-th eigenvalue of the matrix L.

Given the eigenvalues of a matrix A, considering the i-th eigenvalue λ_i, its *algebraic multiplicity* (μ) is defined as the number of times λ_i appears as a root of the characteristic polynomial. The mode corresponding to

$$\lambda_1 = 0$$

represents the agreement dynamics, while the modes corresponding to

$$\lambda_i \neq 0$$

, for $i = 2, ..., N$ represents the disagreement dynamics.
It can be shown that if the graph is either connected or has a spanning tree, it has an eigenvalue equal to 0 with multiplicity equal to 1.

The communication topology is in **strong connectivity condition** if:

- $\lambda_1 = 0$, with algebraic multiplicity equal to 1, $\mu = 1$;
- rank(L) = $N - 1$, N = number of spacecrafts.

This condition is utilized in the implemented system: in particular, the switching topology function of the FDI system (see chapter 5) constructs, if needed, a new undirected topology where each agent is connected with the others, respecting this condition. A graph can be :

- **directed**, if communication among agents can be unidirectional;
- **undirected**, if communication among agents is always bidirectional.

3.1.1 Directed Graph

By definition, a directed graph is such that

$$\forall\ v_i, v_j \in V \mid a_{ij} \neq a_{ji}. \tag{3.6}$$

The outdegree and the indegree of a node i in a directed graph may not be equal. Furthermore, a directed graph can be divided in two categories:

- **strongly connected**, if there exists a directed path between any pair of nodes $v_i, v_j \in V$;
- **directed spanning tree**, if there exists one node, called the root, from which every other node is reachable through a directed path.

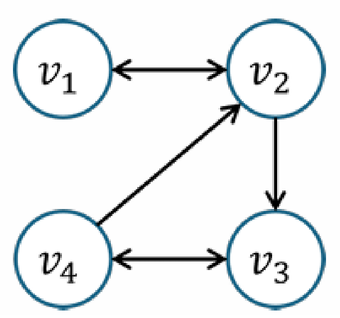

Figure 3.2. Strongly connected directed graph.

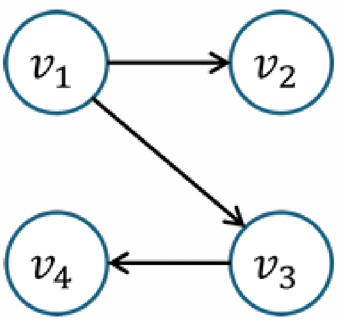

Figure 3.3. Directed spanning tree graph with root v1.

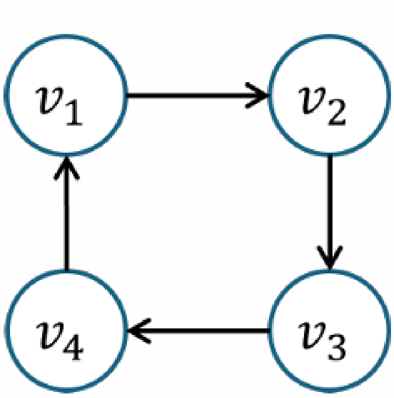

Figure 3.4. Balanced directed graph.

3.1.2 Undirected graph

By definition, an undirected graph is such that

$$\exists \ v_i, v_j \in V \mid a_{ij} = a_{ji}. \tag{3.7}$$

The outdegree and the indegree of a node i in an undirected graph are equal: it is always balanced. The arcs are represented through either double-ended arrows or lines.

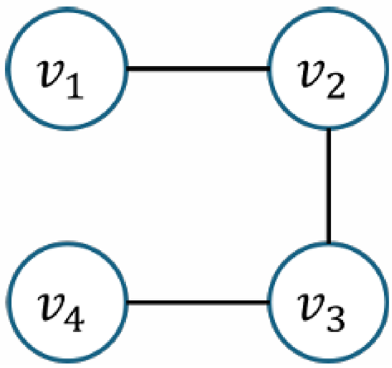

Figure 3.5. Undirected graph.

Furthermore, a directed graph can be divided in two categories:

- **fully connected**, if any two vertices $v_i, v_j \in V$ are neighbors;
- **simply connected**, if there exists a path between any two pair of nodes.

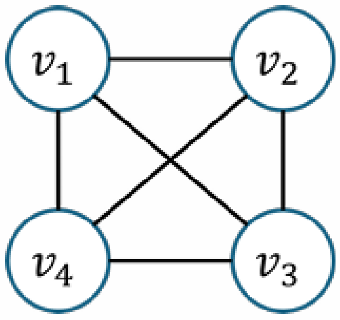

Figure 3.6. Undirected fully connected graph.

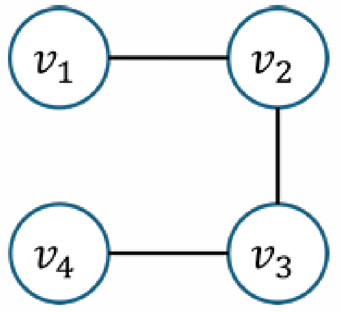

Figure 3.7. Undirected connected graph.

First order protocols achieve a consensus if and only if one of the following condition is valid: the directed graph corresponding to L has a directed spanning tree; the undirected graph corresponding to L is connected.

3.2 Type of consensus

Different types of consensus can be defined based on the agents' dynamics:

- **Single-integrator agents consensus**;
- **Double-integrator agents consensus**;
- **Higher-order agents consensus**;

- **Non-linear agents consensus**.

The first two categories are based on linear consensus algorithms and will be discussed in a general form.

3.2.1 Single-integrator consensus

Consider a network of N single-integrator agents as

$$\dot{x}_i(t) = u_i(t) \quad \text{for i} = 1, ..., \text{N} \tag{3.8}$$

with N number of agents of the communication topology. Consensus is achieved when the states of the agents tends to be the same at steady-state:

$$\lim_{t\to\infty}(x_i(t) - x_j(t)), \quad \text{for i} = 1, ..., \text{N} \tag{3.9}$$

The basic form of consensus protocol for a network of N first-order linear agents is

$$\dot{x}_i(t) = -\sum_{j\in\mathcal{N}_i} a_{ij}\,(x_i(t) - x_j(t)), \tag{3.10}$$

where $x_i(t) \in R$ is a variable of interest. Rewrite the previous equation as

$$\dot{x}_i(t) = -(|N_i|x_i(t) - \sum_{j\in\mathcal{N}_i} a_{ij}\,x_j(t)), \tag{3.11}$$

since the terms a_{ii} are equal to zero by definition. Through matrix L, it is possible to rewrite the system in a compact form

$$\dot{x}(t) = -Lx(t) \tag{3.12}$$

where $x(t) = col\{x_i(t)\} \in R^N$.

3.2.2 Double-integrator consensus

Consider a network of N double-integrator agents as

$$\ddot{x}_i(t) = u_i(t) \quad \text{for i} = 1, ..., \text{N} \tag{3.13}$$

Consensus is achieved when

$$\begin{cases} \lim_{t\to\infty}(x_i(t) - x_j(t)), & \text{for i} = 1, ..., \text{N} \\ \lim_{t\to\infty}(\dot{x}_i(t) - \dot{x}_j(t)) \end{cases} \tag{3.14}$$

The basic form of consensus protocol for a network of N second-order linear agents is

$$\ddot{x}_i(t) = -\sum_{j\in\mathcal{N}_i} a_{ij}\,(x_i(t) - x_j(t) + \gamma(\dot{x}_i(t) - \dot{x}_j(t))), \tag{3.15}$$

where γ is a control parameter. The system can be written in state space form exploiting the matrix L, as:

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} = \begin{bmatrix} 0_N & I_N \\ -L & -\gamma L \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} \tag{3.16}$$

The last two categories are based on Laplacian-based feedback controllers.

3.2.3 Non-linear consensus

A non linear system is defined as

$$\dot{x} = f(x) + g(x) \cdot u \tag{3.17}$$

where $f(x)$ and $g(x)$ are non-linear function of the state vector.
This category includes spacecraft attitude dynamics, kinematic constraints, other systems affected by gyroscopic effects, robotic manipulators, etc.. For general nonlinear dynamics, classical Laplacian-based linear consensus laws are insufficient to guarantee convergence for nonlinear multi-agent systems: a non-linear controller is required. There exist many types of non-linear controllers as feedback linearization, Non-linear Model Predictive Control (NMPC) and Sliding Mode Control (SMC). In this work, a Multi-agent Sliding Mode Control (MSMC) is designed and implemented: the sliding surface is constructed as function of the Laplacian matrix L.

Chapter 4

Dual Gimbal Variable Speed Control Moment Gyros

4.1 Control Moment Gyroscopes

Control Moment Gyroscopes (CMGs) are actuators widely used in space applications. They are defined as *momentum exchange devices* because of their working principle: they exchange angular momentum with the spacecraft, rather than other actuators that expel mass in order to perform the actuation (like thrusters).

Figure 4.1. Control Moment Gyro.

They exert internal torques on the spacecraft, that is, they generate torques that are equal and opposite on the satellite. This property is modeled in the implemented system through two gain blocks with gain equal to -1: one is placed right before the control input enters the actuators system, the other right after its output torque.

4.2 CMG classification

CMGs are produced in different size classes depending on their mass and the maximum torque that they can provide, although internal design may vary. Typical size classes used in the space industry include:

- **Micro-CMGs**, with masses in the range of 50-300 g and torque capabilities of approximately 1-50 mNm;
- **Mini-CMGs**, with masses between 0.5 and 3 kg and torque levels around 0.05-0.5 Nm;
- **Small CMGs**, with masses in the range of 3-10 kg and capable of generating 0.5-5 Nm;
- **Medium CMGs**, with masses of about 10-30 kg and torques in the 5-50 Nm range;
- **Large CMGs**, which can exceed 30 kg and deliver tens to hundreds of Nm of control torque.

In this implementation, a Medium CMGs is modeled, with mass equal to 28 kg and a torque capability of around 40 Nm.

Another classification of this actuator is based on its mechanical structure. A generic CMG is constituted by a flywheel and, at least, one gimbal axis. The flywheel rotates around is own axis with a certain angular velocity w, producing a proportional angular momentum H. The gimbal axis role is to rotate the system in order to generate a variation of the angular momentum proportional to the variation of the orientation of the angular velocity of the flywheel. Depending on the number of gimbal axes and the velocity of the flywheel, either constant or variable, different types of CMG can be distinguished:

- **CMG**, with one gimbal axis and constant flywheel speed;
- **Dual Gimbal CMG (DGCMG)**, with two gimbal aces, one inner and one outer, and constant flywheel speed;
- **Variable Speed CMG (VSCMG)**, with one gimbal axis and a variable flywheel speed;
- **Dual Gimbal Variable Speed CMG (DGVSCMG)**, with two gimbal aces and a variable flywheel speed.

The specific actuators considered in this work are the latter: they are made up of a variable speed flywheel, that can operate as a reaction wheel (RW), the inner and the outer gimbal axes.

Figure 4.2. Dual Gimbal Variable Speed Control Moment Gyro.

Considering the time-varying orientations of the inner and outer gimbals and the variation of the flywheel speed, a single DGVSCMG has three degrees of freedom, providing a 3-D torque. However, for high torque request, it is needed to exploit 2 actuators. Hence, two parallel-mounted DGVSCMGs have been implemented. These actuators have been modeled considering the interaction between the electrical and mechanical subsystems. A PID controller for each rotation axis has been designed and implemented. In the following subsections, the model is described in details.

4.3 Electromechanical system

In general, a gyro is a multi-domain component that couples two energy domains, such that the effort variable in one domain is proportional to the flow variable in the other, and vice versa. The electric motor is a gyro component, which connects the electrical and mechanical domains of the system. In particular, the effort of the electrical domain, the voltage V, is proportional to the mechanical domain flow, the angular velocity w. The effort of the mechanical domain, the torque T, is proportional to the flow of the electrical domain I. This property has been exploited to model the system:

$$\begin{cases} \mathbf{V} = \mathbf{k_e} \cdot \mathbf{w} \\ \mathbf{T} = \mathbf{k_t} \cdot \mathbf{I} \end{cases} \tag{4.1}$$

where ke and kt are the electrical and mechanical constant, assumed to be equal in this work.

For simplicity, here are presented the computations to build the transfer function that describe the relation between the angular velocity of the flywheel and the voltage of the voltage generator, which corresponds to the input of the system and the output of the PID controller (which design will be discussed later on). The same procedure has been used to model the mechanical subsystem of the inner and outer aces.

4.3.1 Electrical subsystem

The electric motor has been modeled, with a first approximation, as a series connection of a voltage source, a resistor, an inductor and a load, across which the voltage drop represents the back electromotive force (back-EMF) of the motor, named V in the following description.

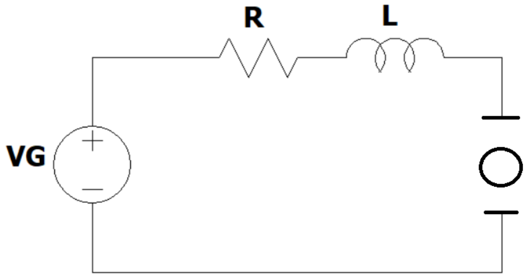

Figure 4.3. Model of the electrical subsystem.

According to the voltage Kirchhoff law:

$$\mathbf{V_g} = \mathbf{R} \cdot \mathbf{I} + \mathbf{L} \cdot \frac{\mathbf{dI}}{\mathbf{dt}} + \mathbf{V} \tag{4.2}$$

4.3.2 Mechanical subsystem

The mechanical subsystem has been modeled neglecting any viscous friction and stiffness component. The subsystem for each axis is constituted by either the flywheel or the inertia of the gimbal aces. Considering the inertia along is axis, the flywheel has been modeled as:

$$\mathbf{T} = \mathbf{J_z} \cdot \dot{\omega} \tag{4.3}$$

4.3.3 State-space representation and transfer function

The state-space representation of the electromechanical system is derived combining the equations that characterize it. The state vector is

$$x = \begin{bmatrix} I \\ w \end{bmatrix} \tag{4.4}$$

where I is the current of the electric motor and w is the angular velocity of the flywheel. The input is the voltage generator, V_G. The state-space representation is:

$$\begin{cases} \dot{x} = A \cdot x + B \cdot V_G \\ y = w \end{cases}$$

$$A = \begin{pmatrix} -\frac{R}{L} & -\frac{k_e}{L} \\ \frac{1}{L} & 0 \end{pmatrix} \qquad B = \begin{pmatrix} \frac{1}{L} \\ 0 \end{pmatrix} \qquad C = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad D = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{4.5}$$

The output is the angular velocity of the flywheel, w.

The transfer function of the latter state-space is computed as explained below. Substituting (1) in (2):

$$\mathbf{V_G} = \mathbf{R} \cdot \frac{\mathbf{J_z \cdot w}}{\mathbf{kt}} + \mathbf{L} \cdot \frac{\mathbf{dI}}{\mathbf{dt}} + \mathbf{k_e \cdot w} \tag{4.6}$$

Passing from time domain to Laplace domain, and collecting the angular velocity:

$$\mathbf{V_G} = \mathbf{w}\left(\frac{\mathbf{J_z \cdot s \cdot (R + Ls)}}{\mathbf{k_t}} + \mathbf{k_e}\right) \tag{4.7}$$

The transfer function that define the relation between the angular velocity of the flywheel and the voltage of the generator is:

$$\mathbf{H} = \frac{\mathbf{w}}{\mathbf{V_G}} = \frac{\frac{\mathbf{1}}{\mathbf{k_e}}}{\mathbf{1} + \left(\frac{\mathbf{J_z \cdot s^2}}{\mathbf{k_e k_t}}\left(\frac{\mathbf{R}}{\mathbf{s}} + \mathbf{L}\right)\right)} \tag{4.8}$$

Providing a voltage such that:

$$\mathbf{V_G} = \mathbf{kp \cdot e_w} = \mathbf{kp} \cdot (\mathbf{w_{des}} - \mathbf{w}) \tag{4.9}$$

The transfer function that relates the current angular velocity to the desired one is:

$$\frac{\mathbf{w}}{\mathbf{w_{des}}} = \frac{\mathbf{H \cdot kp}}{\mathbf{(1 + H \cdot kp)}} \tag{4.10}$$

The parameters considered to model the dynamics of each electromechanical subsystems are:

Parameter	Symbol	Value	Unit of measure
Flywheel axis inertia	I_{zi}	0.7	$[kg/m^2]$
Inner gimbal axis inertia	I_y	0.1	$[kg/m^2]$
Outer gimbal axis inertia	I_z	0.1	$[kg/m^2]$
Motor resistance	R	10	$[\Omega]$
Motor inductance	L	10.9	$[mH]$
Actuator mass	m	28	$[kg]$

Table 4.1. Parameter of the electromechanical subsystems.

The bode diagram that describes the frequency response of the uncontrolled system is shown in Figure(4.4).

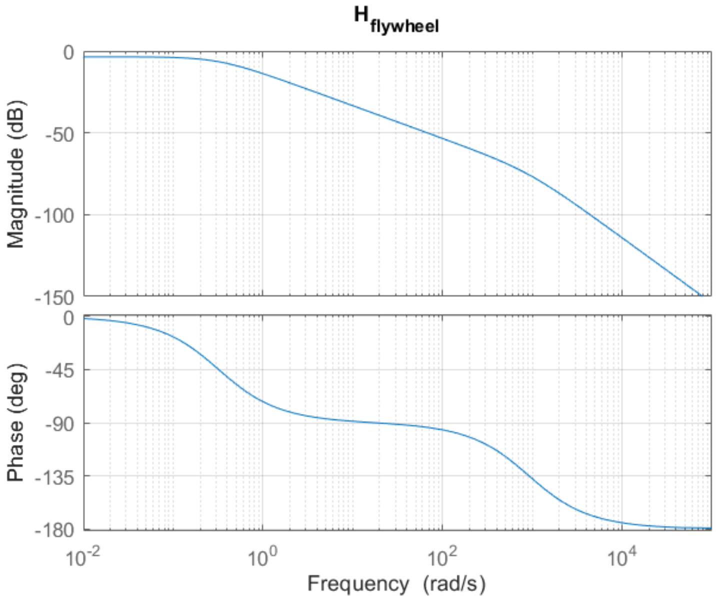

Figure 4.4. Bode diagram of the frequency response of the system.

As observable, the flywheel responds as a second order low pass filter with poles: $p_1 = -0.3215\text{rad/s}, p_2 = -917.1097\text{rad/s}$.

4.4 PID controller

The PID controller was designed to ensure that the angular velocities of each actuators' axis accurately follow the desired velocity reference provided by the employed steering law (described in the following section).

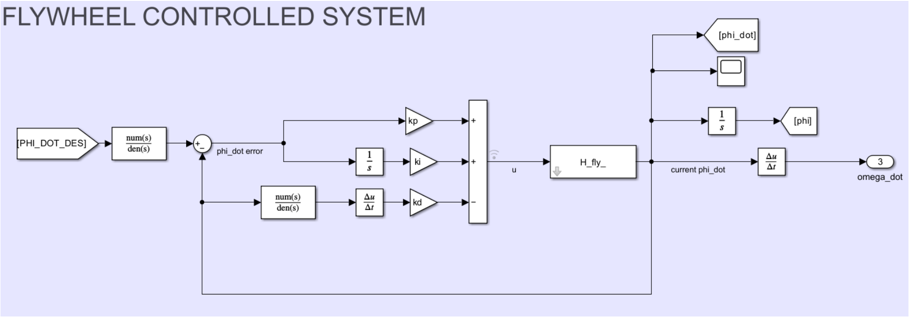

Figure 4.5. Flywheel controlled system.

A Proportional-Integral-Derivative (PID) controller generates a control input that is the sum of three terms: a proportional term, which reacts to the instantaneous error; an integral term, which compensates steady-state errors; and a derivative term, which anticipates future behavior based on the error rate.

In this implementation, two low-pass filters are employed. The first low-pass filter is applied to the input reference signal to model the real response of the system's dynamics. A time constant of $\tau = 0.01s$ was chosen, since higher-frequency components (with smaller time periods) could not be tracked by the real dynamic system without attenuation. The second low-pass filter is placed in the derivative branch to limit high-frequency noise and avoid numerical errors due to rapid signal variations. Its time constant was set to $\tau = 0.05s$.

The proportional, integral, and derivative gains (K_P, K_D, K_I) were tuned through a trial-and-error procedure, based on step and ramp input tests, until system stability and satisfactory tracking performance were achieved. In control theory, two types of problems can be faced:

- regulation problem, where the desired output should reach a constant setpoint;
- tracking problem, where the reference is a dynamic signal to track.

By analyzing the step response of the controlled system, it can be observed that the maximum overshoot corresponds to approximately 5.5% of the steady-state response, while the settling time with a 5% tolerance band ($t_{s,5\%}$) is about 1.5 seconds.

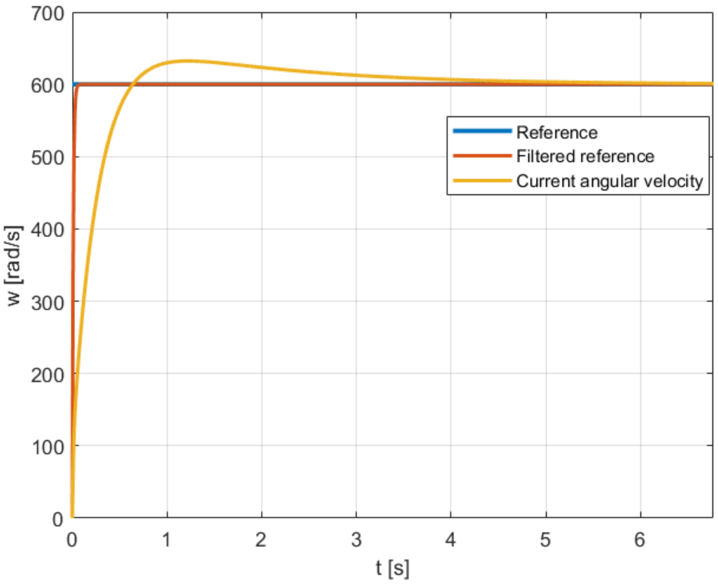

Figure 4.6. Response of the controlled system with a step reference of amplitude 600 rad/s.

The steady-state error is equal to 0.

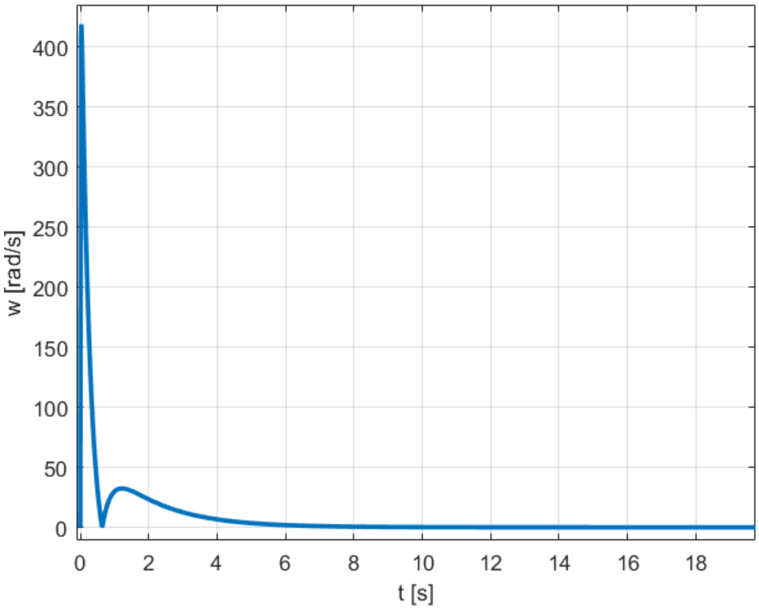

Figure 4.7. Steady-state error for a step reference of amplitude 600 rad/s.

Furthermore, analyzing the response to a ramp with slope equal to 10 (i.e. the reference angular velocity has an acceleration of 10 rad/s^2, that is, two complete turns more of the flywheel at each second), the settling time with a 5% tolerance band ($t_{s,5\%}$) is 2 seconds.

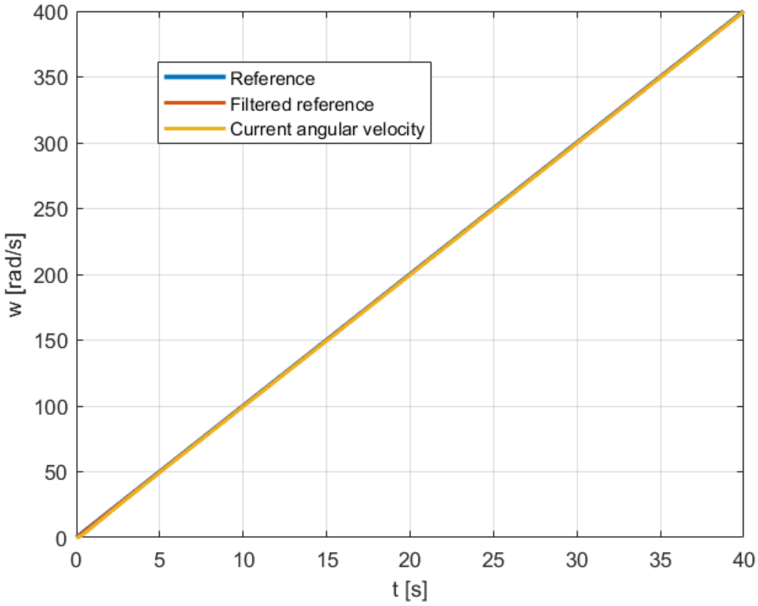

Figure 4.8. Response of the controlled system with a ramp reference with slope equal to 10.

The tracking error at steady-state is equal to 0.375 rad/s, that is 1.87% of the steady-state response.

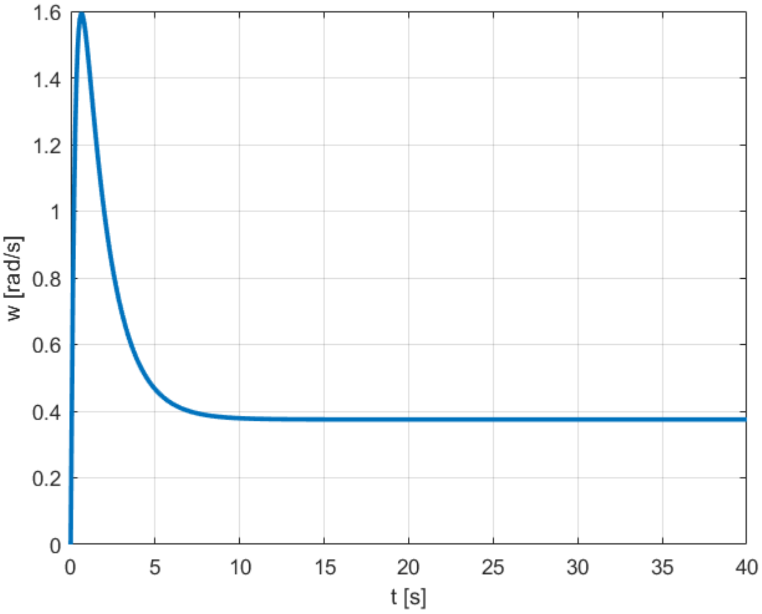

Figure 4.9. Tracking error for a ramp reference with slope equal to 10.

The chosen parameters for the PID controller design are listed in Table(4.2).

Parameter	Symbol	Value	Unit of measure
Proportional constant	k_p	70	-
Integrative constant	k_I	40	-
Derivative constant	k_D	13	-
1st LPF time constant	τ	1	[ms]
2nd LPF time constant	τ_D	5	[ms]

Table 4.2. Parameters of the PID controller.

Performance Parameter	Symbol	Value	Unit of measure
Regulation settling time	$t_{s,5\%}$	1.5	s
Tracking settling time	$t_{s,5\%}$	2	s
Steady-state tracking error	-	0.4	rad/s
Steady-state regulation error	-	0	rad/s
Regulation - max overshoot	$\hat{s}$	5.5	%

Table 4.3. Regulation and tracking performances of the PID controlled system.

4.5 Steering Law

In this section, the steering law implementation is discussed.

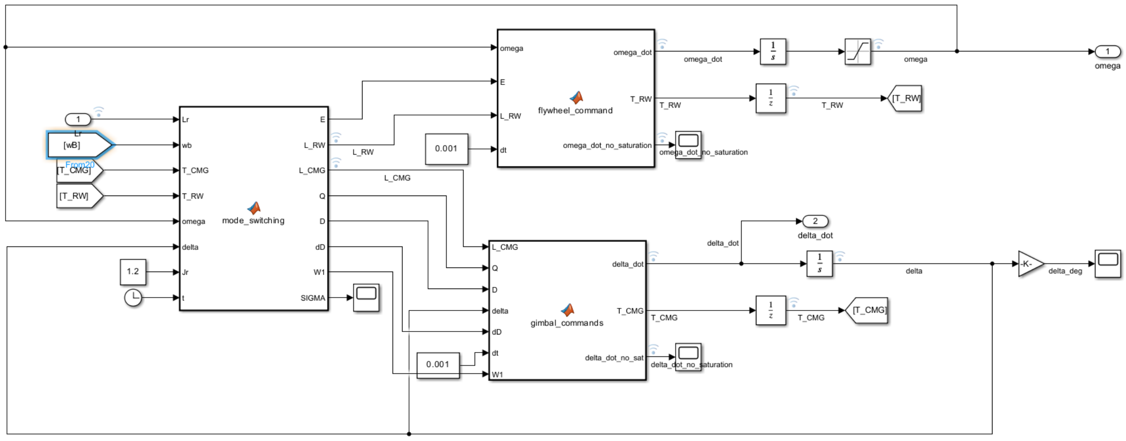

Figure 4.10. Implemented steering law.

Three Matlab functions are employed and connected as shown:

1. the **switching-mode** function, which divide the torque components. Two possible configurations are defined for each actuator. The current configuration type is function of how far the gimbal aces are to a singular configuration. If the DGVSCMG is far from singular conditions, it operates mostly as a CMG, where the gimbal velocities are the torque generators. If a certain amount of torque can not be provided by the CMG-configuration, the equivalent torque is generated by the flywheel acceleration (Reaction Wheels (RW) Mode). Vice versa for the opposite case. If the DGVSCMG is in proximity to a singular configuration, it operates mostly as a Reaction Wheel (RW), where the variable-speed of the flywheel is the main torque generator, until the actuator exits from singular configurations.

2. the **flywheel commands** function, which compute the angular acceleration of the flywheel as function of the requested control torque;

3. the **gimbal commands** function, which compute the angular velocity of the inner and outer gimbal aces as function of the requested control torque.

The steering law implemented in this work aims to decompose the commanded torque among the two DGVSCMGs in such a way that each gimbal rate and flywheel acceleration is the result of the sum of a contribution in the *range space* and a contribution in the *null space* of the actuators' torque. The range space represents the subspace in which the gimbal velocities and wheel accelerations effectively generate a control torque on the spacecraft. The null space identifies those motions that do not produce torque but are exploited either to escape or to move far away singular configurations.

First, the dynamic formulation of the two parallel-mounted DGVSCMGs is presented. To define the gimbal and flywheel dynamics, it is first necessary to establish the unit vectors associated with the outer and inner gimbals and with the flywheel rotation axis.

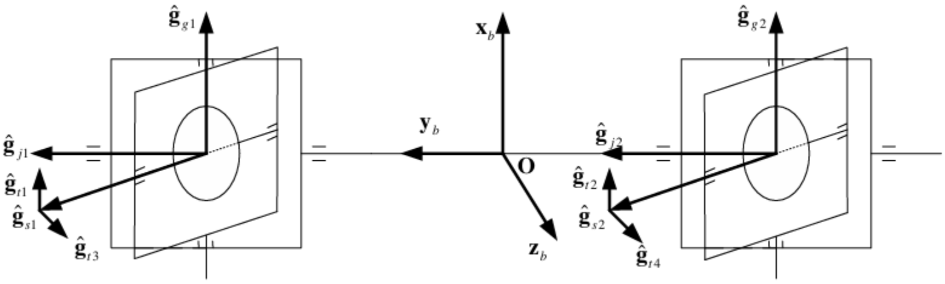

Figure 4.11. Representation of parallel mounted DGVSCMGs.

In the initial configuration, the two DGVSCMGs are mounted such that the flywheel spin axis is aligned with the body X-axis (verser $\boldsymbol{g_{s_i}}$), the inner gimbal axis with the body Y-axis (verser $\boldsymbol{g_{g_i}}$), and the outer gimbal axis with the body Z-axis (verser $\boldsymbol{g_{j_i}}$). Additionally, an offset is introduced in the outer gimbal angle of the first gyro, which corresponds to an initial rotation of 30° about the Z-axis.

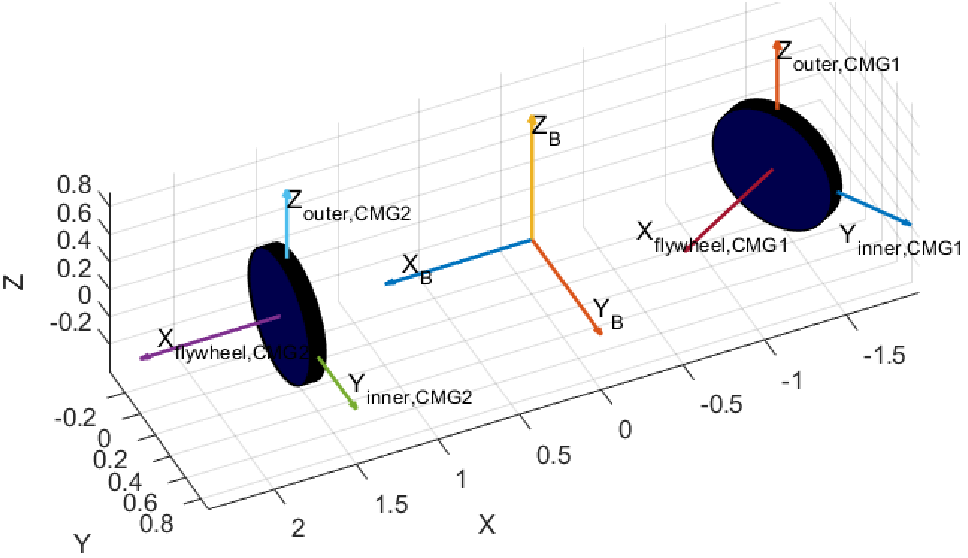

Figure 4.12. DGVSCMGs initial configuration.

The versers that define the direction of the output torque are computed as the vectorial product between the gimbal aces and the flywheel axis.

$$\hat{\boldsymbol{g_{t1}}} = \hat{\boldsymbol{g_{j1}}} \times \hat{\boldsymbol{g_{s1}}} \tag{4.11}$$

$$\hat{\boldsymbol{g_{t2}}} = \hat{\boldsymbol{g_{j2}}} \times \hat{\boldsymbol{g_{s2}}} \tag{4.12}$$

$$\hat{\boldsymbol{g_{t3}}} = \hat{\boldsymbol{g_{g1}}} \times \hat{\boldsymbol{g_{s1}}} \tag{4.13}$$

$$\hat{\boldsymbol{g_{t4}}} = \hat{\boldsymbol{g_{g2}}} \times \hat{\boldsymbol{g_{s2}}} \tag{4.14}$$

The rotation convention adopted to describe the relative orientations is a 3-2-1 Euler sequence, in which the first rotation is associated with the outer gimbal angle α_i about the Z-axis, the second rotation is associated with the inner gimbal angle β_i about the Y-axis, and the third rotation (about X) is null. The coordinate transformations from the outer and inner gimbal frames to the spacecraft body frame and vice versa are described through rotation matrices. Considering the elementary rotation matrices:

$$R_{z_i} = \begin{bmatrix} cos(\alpha_i) & -sin(\alpha_i) & 0 \\ sin(\alpha_i) & cos((\alpha_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.15}$$

$$R_{y_i} = \begin{bmatrix} cos(\beta_i) & 0 & sin(\beta_i) \\ 0 & 1 & 0 \\ -sin(\beta_i) & 0 & cos((\beta_i) \end{bmatrix} \tag{4.16}$$

The rotation matrix the describes the rotation from the body frame to the outer gimbal frame coincide with R_z. The rotation matrix that describes the rotation from the body frame to the inner gimbal frame is given by the cascade product $R_{b2i} = R_z \cdot R_y$.

The total angular momentum of the two parallel-mounted DGVSCMGs can be expressed as:

$$\mathbf{H} = \mathbf{H}_w + \mathbf{H}_g \tag{4.17}$$

where $\mathbf{H}_w$ and $\mathbf{H}_g$ represent the wheel and gimbal angular momenta, respectively.

Assuming that J_o and J_i denote the inertia of the outer and inner gimbals, and $\dot{\alpha}_i$, $\dot{\beta}_i$ their angular rates, the gimbal angular momentum of the i-th DGVSCMG is given by:

$$\mathbf{H}_g = \sum_{i=1}^{2} \Big(J_o(\boldsymbol{\omega}_b + \dot{\alpha}_i \hat{\mathbf{g}}_{j_i}) + J_i(\boldsymbol{\omega}_b + \dot{\beta}_i \hat{\mathbf{g}}_{g_i} + \dot{\alpha}_i \hat{\mathbf{g}}_{j_i}) \Big) \tag{4.18}$$

where $\boldsymbol{\omega}_b$ is the angular velocity of the spacecraft with respect to the inertial frame, and $\hat{\mathbf{g}}_{j_i}$, $\hat{\mathbf{g}}_{g_i}$ are the outer and inner gimbal aces.

The wheel angular momentum of each DGVSCMG can be written as:

$$\mathbf{H}_w = \sum_{i=1}^{2} J_w(\boldsymbol{\omega}_b + \dot{\alpha}_i \hat{\mathbf{g}}_{j_i} + \dot{\beta}_i \hat{\mathbf{g}}_{g_i} + \Omega_i \hat{\mathbf{g}}_{s_i}) \tag{4.19}$$

where J_w is the flywheel inertia and Ω_i the flywheel speed of the i-th DGVSCMG.

Since the spacecraft angular velocity and gimbal rates are much smaller than the wheel speed, the total angular momentum can be approximated as:

$$\mathbf{H} \approx \sum_{i=1}^{2} J_w \Omega_i \hat{\mathbf{g}}_{s_i} \tag{4.20}$$

The time derivative of **H** yields to the dynamics equation:

$$\dot{\boldsymbol{H}} = \boldsymbol{M}_B = \sum_{i=1}^{2} \left(J_w \dot{\Omega}_i \, \hat{\boldsymbol{s}}_i + J_w \Omega_i \, \frac{d}{dt}(\hat{\boldsymbol{s}}_i) \right) \tag{4.21}$$

The gimbal torque contributions can be expressed in compact form as:

$$\mathbf{M}_B = \mathbf{F}\boldsymbol{\Omega} + \mathbf{Q}\dot{\boldsymbol{\delta}} + \mathbf{E}\dot{\boldsymbol{\Omega}} \tag{4.22}$$

$$\dot{\delta} = [\dot{\alpha}; \dot{\beta}] \tag{4.23}$$

where the matrices **F**, **Q**, and **E** are defined as:

$$\mathbf{F} = [\boldsymbol{\omega}_b \times J_{r1} \hat{\mathbf{g}}_{s_1}, \, \boldsymbol{\omega}_b \times J_{r2} \hat{\mathbf{g}}_{s_2}] \tag{4.24}$$

$$\mathbf{Q} = [\hat{\mathbf{g}}_{j_1} \times J_{r_1} \boldsymbol{\Omega}_1 \hat{\mathbf{g}}_{s_1}, \, \hat{\mathbf{g}}_{j_2} \times J_{r_2} \boldsymbol{\Omega}_2 \hat{\mathbf{g}}_{s_2}, \, \hat{\mathbf{g}}_{g_1} \times J_{r_1} \boldsymbol{\Omega}_1 \hat{\mathbf{g}}_{s_1}, \, \hat{\mathbf{g}}_{g_2} \times J_{r_2} \boldsymbol{\Omega}_2 \hat{\mathbf{g}}_{s_2}] \tag{4.25}$$

$$\mathbf{E} = [J_{r_1} \hat{\mathbf{g}}_{s_1}, \, J_{r_2} \hat{\mathbf{g}}_{s_2}] \tag{4.26}$$

Assuming the attitude control torque command is given by $\mathbf{L}_r$, the control equation becomes:

$$\mathbf{L}_r = \mathbf{M}_B - \mathbf{F}\boldsymbol{\Omega} \tag{4.27}$$

and therefore the control command in terms of the DGVSCMG dynamic variables can be written as:

$$\mathbf{Q}\dot{\boldsymbol{\delta}} + \mathbf{E}\dot{\boldsymbol{\Omega}} = \mathbf{L}_r \tag{4.28}$$

This formulation provides a direct link between the commanded spacecraft torque and the gimbal and wheel accelerations. Since the inertia of the aces of rotation of the flywheels of the 2 actuators are considered equal, from now on the symbol used will be $J_{r1} = J_{r2} = J_r$.

4.5.1 Singularities identification

It is of paramount importance to identify all the singular configurations of these actuators setting in order to avoid them:

- when the inner gimbal angle of a DGVSCMG reaches 90°, the outer and flywheel aces become collinear;

- when the flywheels aces of the two actuators are parallel to each other.

The CMG singularity avoidance method relies on the definition of a singularity index matrix, denoted as $\mathbf{D}$, constructed from the outer and inner gimbal rate vectors. The matrix $\mathbf{D}$, associated with singular configurations, is defined as:

$$\mathbf{D} = [\hat{\boldsymbol{g_{t1}}}\ \hat{\boldsymbol{g_{t2}}}\ \hat{\boldsymbol{g_{t3}}}\ \hat{\boldsymbol{g_{t4}}}] \tag{4.29}$$

In the switching mode function, the singularity detection is performed through the evaluation of the minimum singular value σ_{min} of the matrix $\boldsymbol{D}$: when it reaches zero, the rank of the matrix decreases, highlighting a singular configuration has been encountered. To prevent this situation, σ_{min} is compared to a threshold value in order to determine which configuration of the actuator should be chosen. The computation of the singular values of the matrix is done through the Singular Value Decomposition (SVD). Another singularity measurement function is the following:

$$\boldsymbol{V}(\boldsymbol{\delta}) = \frac{1}{2}(\boldsymbol{\sigma_1} - \boldsymbol{\sigma_3})^2 \tag{4.30}$$

where σ_1 and σ_3 are the minimum and the maximum singular values, respectively. The latter formula is exploited for the next computations.

4.5.2 Gimbal commands

The steering law for the two parallel-mounted DGVSCMGs can be formulated by combining the Control Moment Gyroscope (CMG) steering law with the Reaction Wheel (RW) steering law. The objective of the CMG steering component is to keep the system away from singularities while ensuring that the commanded attitude torque $\mathbf{L}_r$ is produced. The overall control command can be expressed as:

$$\mathbf{L}_r = \mathbf{E}\dot{\boldsymbol{\Omega}} + \mathbf{Q}\dot{\boldsymbol{\delta}} \tag{4.31}$$

that is the superposition of the two contributions:

$$\mathbf{Q}\dot{\boldsymbol{\delta}} = \mathbf{L}_{\text{CMG}} \tag{4.32}$$

$$\mathbf{E}\dot{\boldsymbol{\Omega}} = \mathbf{L}_{\text{RW}} \tag{4.33}$$

The CMG cost function is then defined as:

$$J_1(\dot{\boldsymbol{\delta}}) = \mathbf{V}(\delta + \dot{\delta}\mathbf{\Delta t}) + \frac{\mathbf{1}}{\mathbf{2}}\dot{\boldsymbol{\delta}}^{\mathbf{T}}\mathbf{W_1}\dot{\boldsymbol{\delta}} \tag{4.34}$$

where $\mathbf{W}_1$ is a positive definite weighting matrix. A minimum-norm solution of (4.32) is attained by using the Lagrange multiplier vector $\boldsymbol{\lambda_1}$.

$$\boldsymbol{L}_1(\dot{\boldsymbol{\delta}}) = \boldsymbol{J}_1(\dot{\boldsymbol{\delta}}) + \boldsymbol{\lambda}_{\mathbf{1}}^T \left(Q\dot{\boldsymbol{\delta}} - L_{CMG}\right) \tag{4.35}$$

The minimization condition of $\boldsymbol{L_1}(\dot{\boldsymbol{\delta}})$ is

$$\frac{\boldsymbol{\partial L_1}}{\boldsymbol{\partial \dot{\delta}}} = \frac{\boldsymbol{\partial J_1}}{\boldsymbol{\partial \dot{\delta}}} + \boldsymbol{\lambda}_{\boldsymbol{1}}^T \boldsymbol{Q} = \boldsymbol{0} \tag{4.36}$$

The gradient vector and Hessian of the cost function are:

$$\mathbf{g} = \Delta t^2 V'(\delta) \tag{4.37}$$

$$\mathbf{H} = \Delta t^2 V''(\delta) \tag{4.38}$$

where

$$V''(\delta) = \begin{bmatrix} \frac{\partial^2 V}{\partial^2 \dot{\alpha}} & \frac{\partial^2 V}{\partial \dot{\alpha} \partial \dot{\beta}} \\ \frac{\partial^2 V}{\partial \dot{\alpha} \partial \dot{\beta}} & \frac{\partial^2 V}{\partial^2 \dot{\beta}} \end{bmatrix} \tag{4.39}$$

$$V'(\delta) = \begin{bmatrix} \frac{\partial V}{\partial \dot{\alpha}} & \frac{\partial V}{\partial \dot{\beta}} \end{bmatrix} \tag{4.40}$$

Ignoring the high order terms, the Taylor expansion of the singularity measurement function (4.30) is:

$$V(\boldsymbol{\delta} + \dot{\boldsymbol{\delta}} \Delta t) \simeq V(\boldsymbol{\delta}) + \boldsymbol{g}^T \dot{\boldsymbol{\delta}} + \frac{1}{2} \dot{\boldsymbol{\delta}}^T \boldsymbol{H} \dot{\boldsymbol{\delta}} \tag{4.41}$$

By matching Eq. (4.41) to Eq. (4.34) and applying (4.36):

$$\dot{\boldsymbol{\delta}}^d = -(\boldsymbol{H} + \boldsymbol{W}_1)^{-1} (\boldsymbol{g} + \boldsymbol{Q}^T \boldsymbol{\lambda}_1) \tag{4.42}$$

Inserting Eq. (4.42) into Eq. (14), the Lagrange multiplier vector can be expressed as:

$$\boldsymbol{\lambda}_1 = -(\boldsymbol{Q} \boldsymbol{H}_1^{-1} \boldsymbol{Q}^T)^{-1} \boldsymbol{L}_{CMG} - (\boldsymbol{Q} \boldsymbol{H}_1^{-1} \boldsymbol{Q}^T)^{-1} \boldsymbol{Q} \boldsymbol{H}_1^{-1} \boldsymbol{g} \tag{4.43}$$

where $\boldsymbol{H}_1 = \boldsymbol{H} + \boldsymbol{W}_1$.
Therefore, the gimbal rate command is obtained by:

$$\dot{\boldsymbol{\delta}}^d = \boldsymbol{S}_1 \boldsymbol{C}_1^+ \boldsymbol{L}_{CMG} + (\boldsymbol{S}_1 \boldsymbol{C}_1^+ \boldsymbol{S}_1^T - \boldsymbol{H}_1^{-1}) \boldsymbol{g} \tag{4.44}$$

where

$$\boldsymbol{S}_1 = \boldsymbol{H}_1^{-1} \boldsymbol{Q}^T, \qquad \boldsymbol{C}_1^+ = (\boldsymbol{Q} \boldsymbol{H}_1^{-1} \boldsymbol{Q}^T)^{-1}.$$

The *range-space* velocity contribution is given by:

$$\dot{\boldsymbol{\delta}}^d_{RANGE} = \boldsymbol{S}_1 \boldsymbol{C}_1^+ \boldsymbol{L}_{CMG} \tag{4.45}$$

The *null-space* contribution is:

$$\dot{\boldsymbol{\delta}}^d_{NULL} = (\boldsymbol{S}_1 \boldsymbol{C}_1^+ \boldsymbol{S}_1^T - \boldsymbol{H}_1^{-1}) \boldsymbol{g} \tag{4.46}$$

To avoid angular rates to overcome their physical limits, the gimbal rate command is saturated according to the maximum allowable gimbal rate $\dot{\boldsymbol{\beta}}_{\max}$:

$$\dot{\boldsymbol{\delta}} = \begin{cases} \dot{\boldsymbol{\delta}}, & \text{if } \|\dot{\boldsymbol{\delta}}\| \le \dot{\boldsymbol{\delta}}_{\max} \\ \dfrac{\dot{\boldsymbol{\delta}}_{\max}}{\|\dot{\boldsymbol{\delta}}\|}\dot{\boldsymbol{\delta}}, & \text{if } \|\dot{\boldsymbol{\delta}}\| > \dot{\boldsymbol{\delta}}_{\max} \end{cases} \tag{4.47}$$

This ensures that the commanded gimbal rates remain within physically feasible limits while maintaining effective singularity avoidance.

4.5.3 Flywheel commands

A similar procedure has been considered to attain the contributions of the flywheel commands.

Considering nominally constant wheel speeds, the RW cost function yields

$$J_2(\dot{\Omega}) = \frac{1}{2}\dot{\Omega}^T W_2 \dot{\Omega} + \frac{1}{2}(\dot{\Omega} - \dot{\Omega}_d)^T Z (\dot{\Omega} - \dot{\Omega}_d) \tag{4.48}$$

where W_2 and $Z \in \mathbb{R}^{2\times 2}$ are positive definite matrices, $\dot{\Omega}_d$ is the preferred gimbal angular acceleration vector and can be obtained by

$$\dot{\Omega}_d = (\Omega_d - \Omega)/\Delta t \tag{4.49}$$

where Ω_d is the average gimbal speed vector, which is selected as the preferred gimbal speed vector.

A minimum-norm solution of Eq. (4.33) can be attained by using the Lagrange multiplier vector $\boldsymbol{\lambda}_2$:

$$L_2(\dot{\Omega}) = J_2(\dot{\Omega}) + \boldsymbol{\lambda}_2^T (E\dot{\Omega} - L_{CMG}) \tag{4.50}$$

The minimization condition of $L_2(\dot{\Omega})$ is

$$\frac{\partial L_2}{\partial \dot{\Omega}} = \frac{\partial J_2}{\partial \dot{\Omega}} + \boldsymbol{\lambda}_2^T E = 0 \tag{4.51}$$

By matching Eqs. (4.48)–(4.51), $\dot{\Omega}^d$ can be obtained:

$$\dot{\Omega}^d = (Z + W_2)^{-1}(Z^T \dot{\Omega}_d - E^T \boldsymbol{\lambda}_2) \tag{4.52}$$

Inserting Eq. (4.52) into Eq. (4.33), the Lagrange multiplier vector can be expressed as

$$\boldsymbol{\lambda}_2 = (E H_2^{-1} E^T)^+ (E H_2^{-1} Z^T \dot{\Omega}_d) + (E H_2^{-1} E^T)^+ L_{RW} \tag{4.53}$$

where $H_2 = Z + W_2$ and W2 is a weighting matrix.

Therefore, the flywheel angular acceleration command is derived from

$$\dot{\Omega}^d = S_2 C_2^+ L_{RW} - (S_2 C_2^+ S_2^T - H_2^{-1}) Z^T \dot{\Omega}_d \tag{4.54}$$

where

$$S_2 = H_2^{-1}E^T \quad \text{and} \quad C_2^+ = (EH_2^{-1}E^T)^+.$$

To prevent the flywheels to overcome their physical limits, the wheel angular acceleration command is saturated at its maximum allowable value $\dot{\boldsymbol{\Omega}}_{\text{max}}$:

$$\dot{\boldsymbol{\Omega}} = \begin{cases} \dot{\boldsymbol{\Omega}}, & \text{if } \|\dot{\boldsymbol{\Omega}}\| \leq \dot{\boldsymbol{\Omega}}_{\text{max}} \\ \dfrac{\dot{\boldsymbol{\Omega}}_{\text{max}}}{\|\dot{\boldsymbol{\Omega}}\|}\dot{\boldsymbol{\Omega}}, & \text{if } \|\dot{\boldsymbol{\Omega}}\| > \dot{\boldsymbol{\Omega}}_{\text{max}} \end{cases} \tag{4.55}$$

Finally, combining the CMG and RW steering components ensures that the overall system maintains the desired torque while avoiding singularities and staying within the physical limits of gimbal and wheel motion. This hybrid control allocation represents a robust and flexible strategy for DGVSCMG-based spacecraft attitude control.

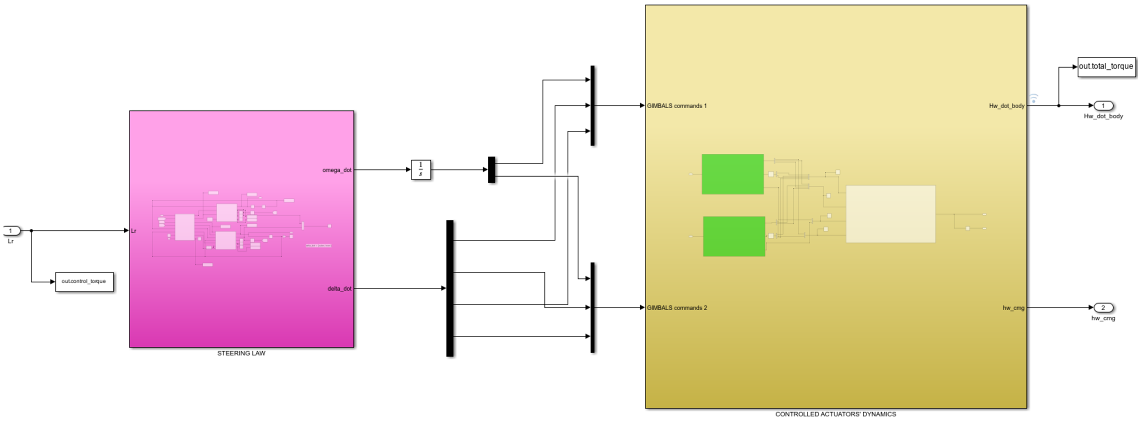

Figure 4.13. Steering law and controlled actuators' systems connected.

4.5.4 Steering law testing

The implemented steering law has been tested with a column vector of steps with amplitude 10 Nm. The combination of the steering law and the controlled DGVSCMGs systems has satisfactory performances.

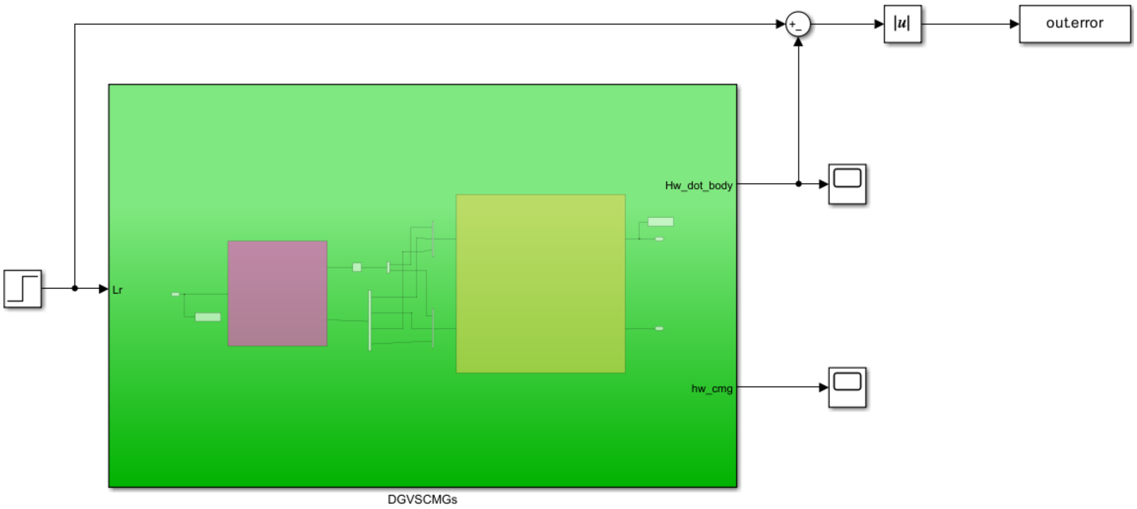

Figure 4.14. Steering law testing.

The response of the entire controlled system is now analyzed:

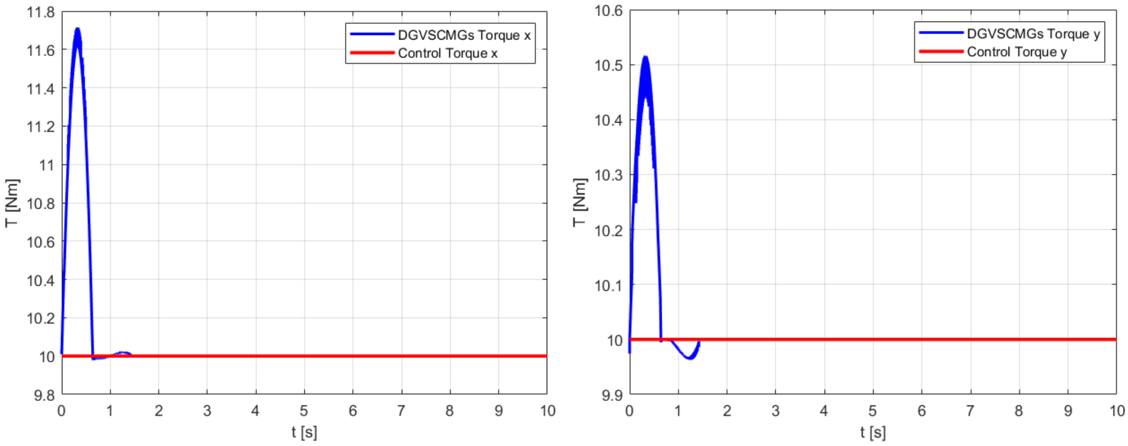

Figure 4.15. Response of the system compared to the reference along x.

Figure 4.16. Response of the system compared to the reference along y.

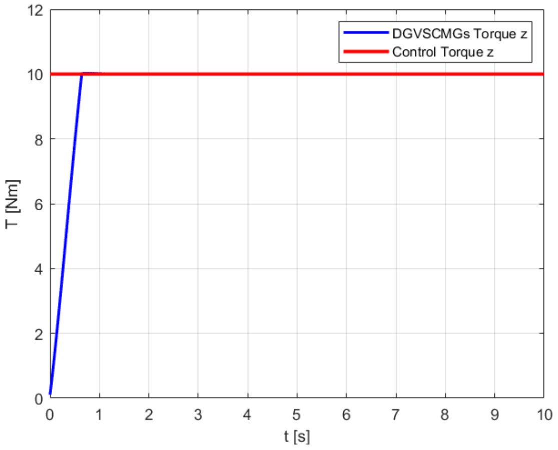

Figure 4.17. Response of the system compared to the reference along z.

- the maximum overshoot is equal to 11.7 Nm;
- the settling time with a 5% of band tolerance is 1.3 seconds;
- the steady-state error reaches zero;

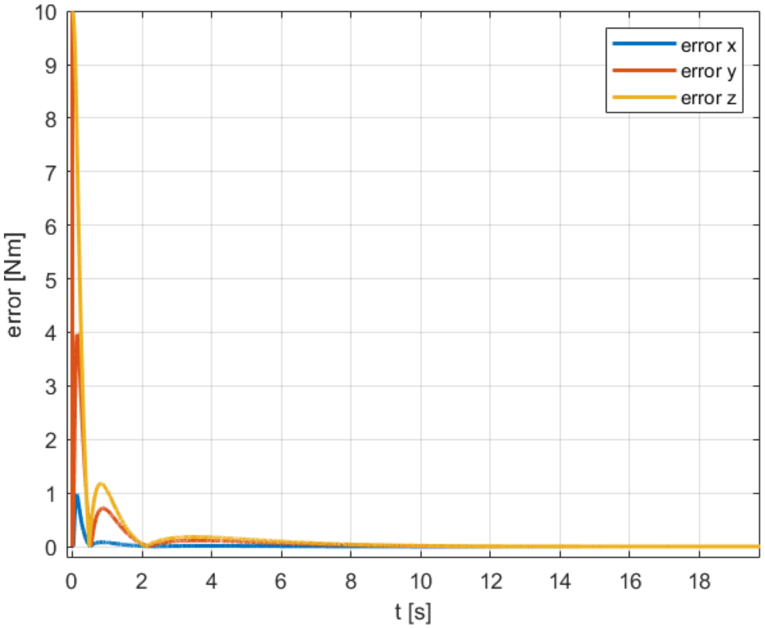

Figure 4.18. Evolution in time of the error.

The minimum singular value of the matrix D is evaluated in time: as one can observe, it overcomes the chosen threshold for singularity identification, that is 0.3, and then it increases oscillating. The oscillations in Figure(4.19) are the result of the null-space gimbal velocities' components, needed to escape from singular configuration vicinity.

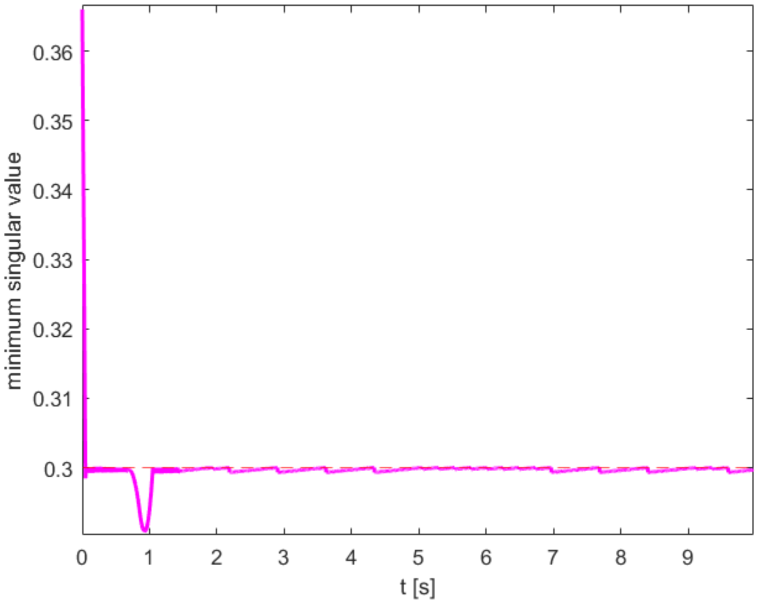

Figure 4.19. Minimum singular value of the singularity identification matrix D.

According to the **switching mode** function, when the system is far from singular configurations, the actuators torque is generated predominantly by the gimbal velocities. The remaining torque to be produced is provided by the flywheel acceleration. In this simulation, at $t = 1\,\mathrm{s}$ the actuator system enters the vicinity of a singular configuration. As a result, the torque is provided in RW-mode, while the CMG actuators steer the system away from the singularity.

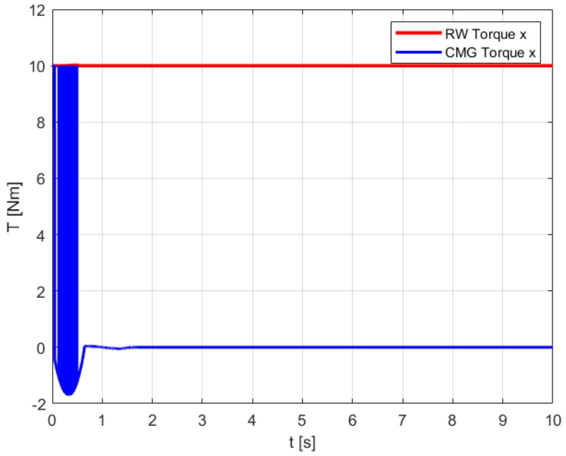

Figure 4.20. RW and CMG configurations' contribution along x.

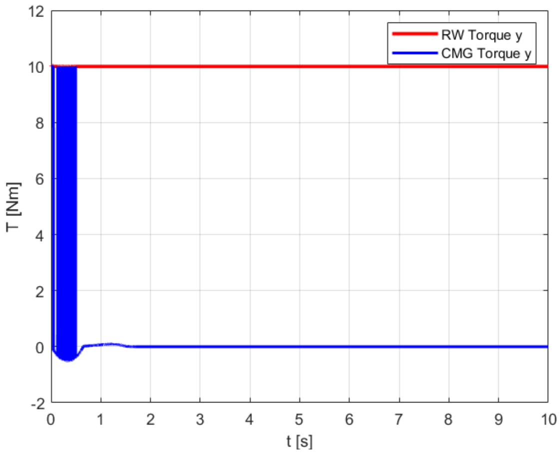

Figure 4.21. RW and CMG configurations' contribution along y.

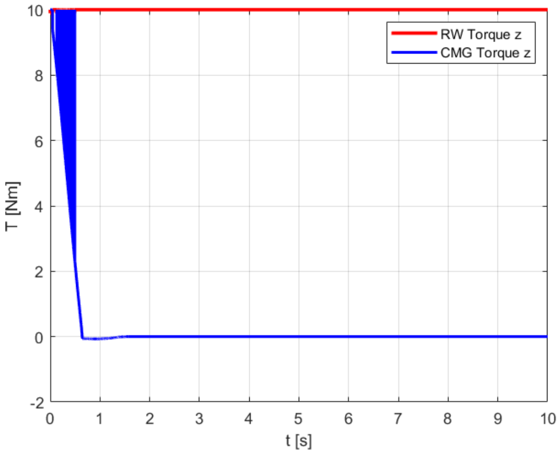

Figure 4.22. RW and CMG configurations' contribution along z.

The effective gimbal commands obtained by the actuators are attenuated and have a slight delay during the transient, as a consequence of the presence the low pass filters. The accelerations and velocity are obtained as the sum of a *range-space* component and a *null-space* component each.

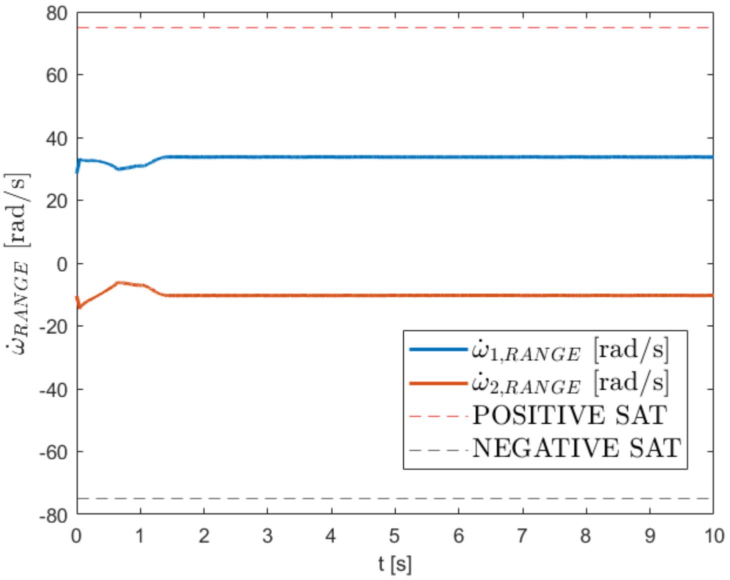

Figure 4.23. *Range-space $\dot{\omega}$ contributions.*

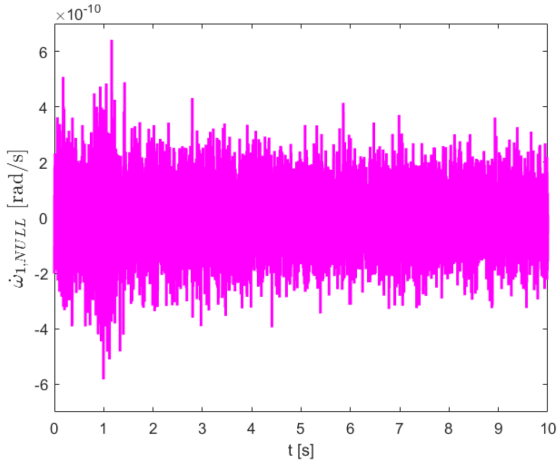

Figure 4.24. *Null-space $\dot{\omega}_1$ contributions.*

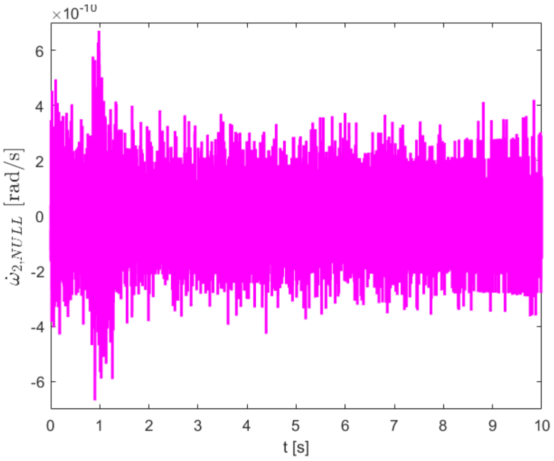

Figure 4.25. *Null-space $\dot{\omega}_2$ contributions.*

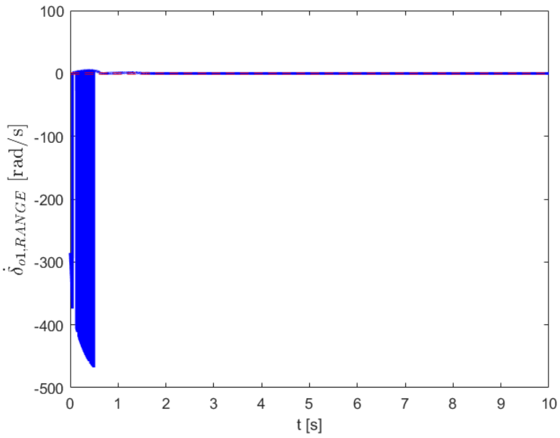

Figure 4.26. Range-space $\dot{\delta}_{o,1}$ contributions.

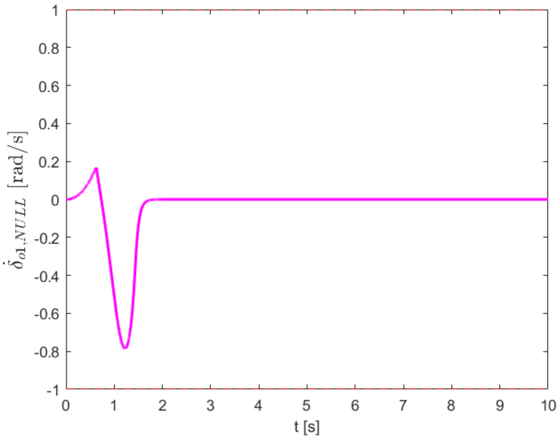

Figure 4.27. Null-space $\dot{\delta}_{o,1}$ contributions.

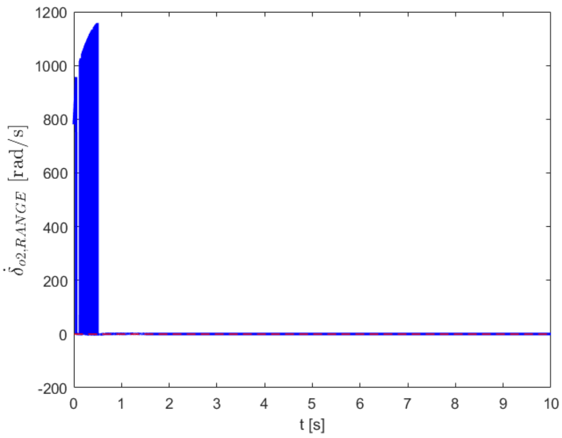

Figure 4.28. Range-space $\dot{\delta}_{o,2}$ contributions.

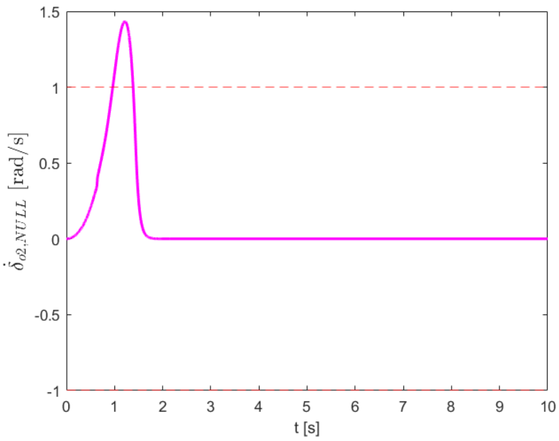

Figure 4.29. Null-space $\dot{\delta}_{o,2}$ contributions.

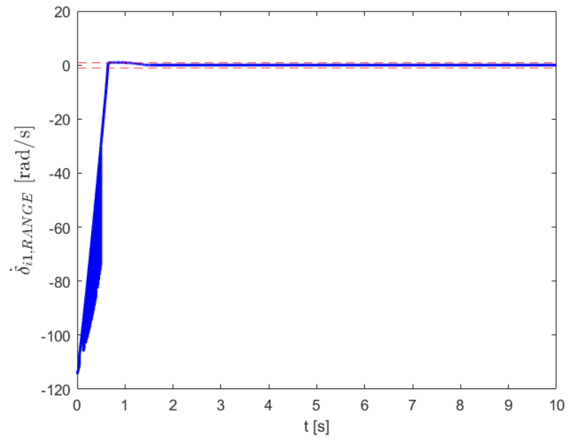

Figure 4.30. Range-space $\dot{\delta}_{i,1}$ contributions.

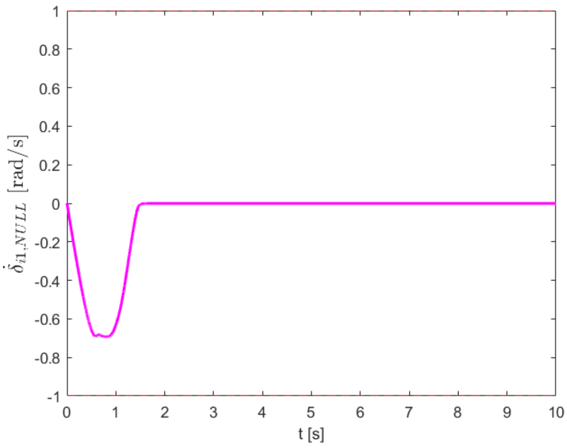

Figure 4.31. Null-space $\dot{\delta}_{i,1}$ contributions.

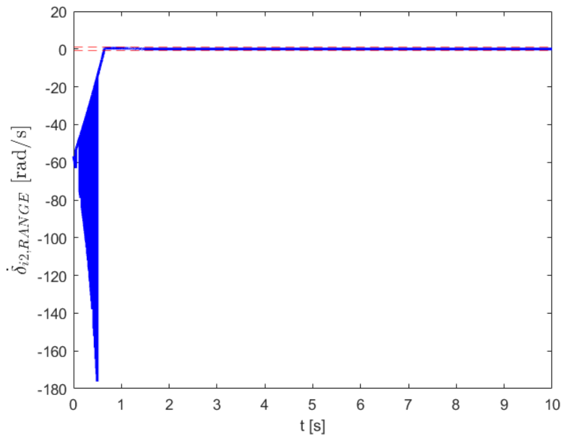

Figure 4.32. Range-space $\dot{\delta}_{i,2}$ contributions.

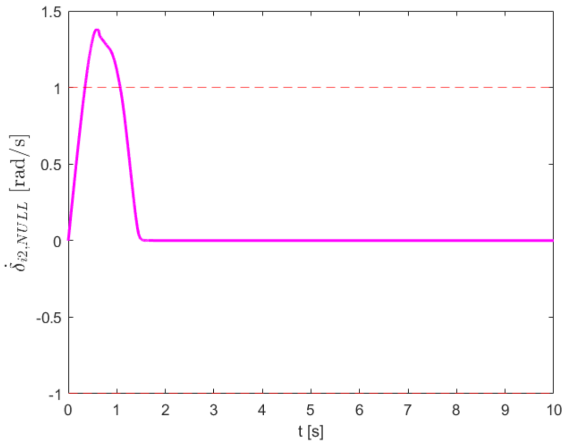

Figure 4.33. Null-space $\dot{\delta}_{i,2}$ contributions.

Then, the commanded signals (accelerations and velocities) pass through a saturator to model the physical limits of the mechanical subsystem of the DGVSCMG. The saturation values are listed in Table(4.4). Finally, the velocity and accelerations requested by the steering law are compared to the ones obtained by the combination of the PID controllers and the dynamics of the DGVSCMGs.

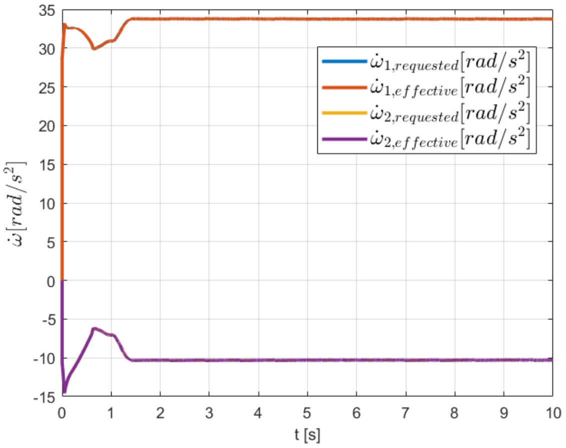

Figure 4.34. Requested and actual flywheels'accelerations.

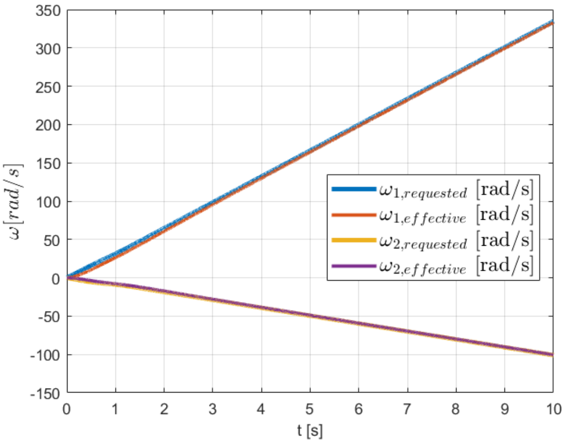

Figure 4.35. Requested and actual flywheels'velocities.

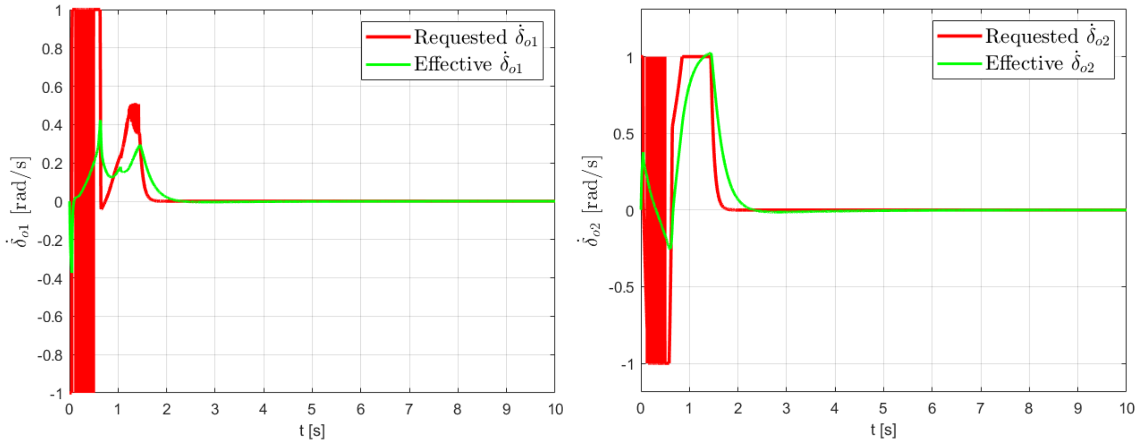

Figure 4.36. Requested and actual outer gimbal velocities.

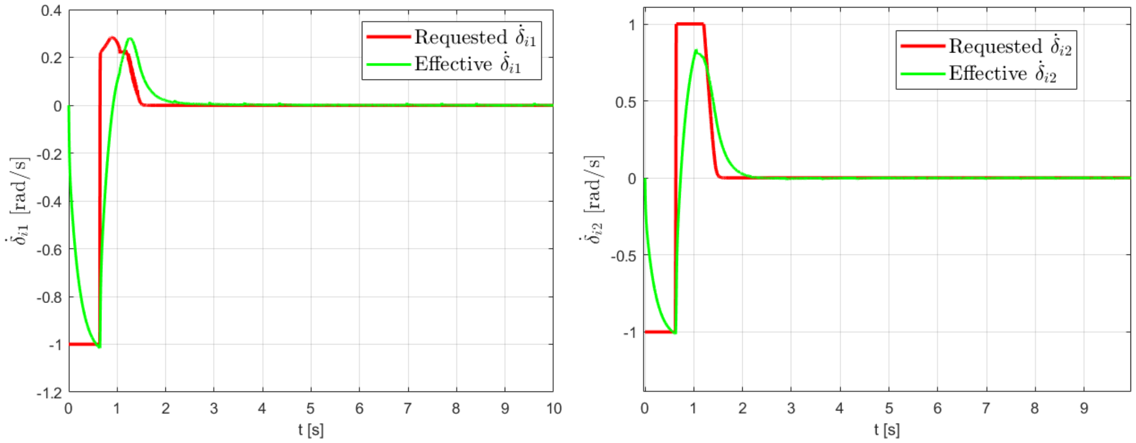

Figure 4.37. Requested and actual inner gimbal velocities.

Parameter	Symbol	Value	Unit of measure
weighting matrix	W1	$diag([10^{-4},10^{-4},10^{-4}])$	-
weighting matrix	Z	diag([0.1,0.1,0.1])	-
weighting matrix	W2	$diag([10^-3,10^-3,10^-3])$	-
flywheel angular velocity saturation	w_{SAT}	600	$[rad/s]$
flywheel angular acceleration saturation	$\dot{w}_{SAT}$	75	$[rad/s^2]$
outer gimbal angular velocity saturation	$\dot{\alpha}_{SAT}$	1	$[rad/s]$
inner gimbal angular velocity saturation	$\dot{\beta}_{SAT}$	1	$[rad/s]$

Table 4.4. Parameters for steering law testing.

Chapter 5

Actuator Fault Detection and Isolation

In this section, the actuators' Fault Detection and Isolation (FDI) subsystems implementation will be discussed. FDI systems are devices able to monitor the system they are designed for, in order to detect and isolate any fault. In order to proceed the discussion, the terms *Fault* and *Failure* must be defined. A **fault** is a defect or an abnormal condition in a system. A **failure** is the permanent interruption of a systems ability to perform a required function under specified operating conditions. It can be distinguished between two types of failures:

- **Fail operational**: the scheduled operation is continued without the need for ground intervention;
- **Fail safe**: the scheduled operation is terminated and the affected subsystem is switched into a safe state (i.e. the major functions are preserved), until ground intervenes.

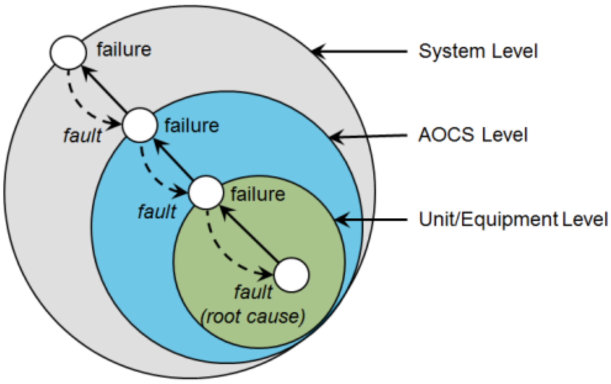

Figure 5.1. Fault and Failure as seen from different levels.

Figure (5.1) graphically explains how fault and failure are seen from different levels. If a fault happens at Unit Level, it can result in a failure for the unit itself but not for the successive level (that is, the AOCS level). E.g., if the FDI subsystem of the first spacecraft of the communication topology detects a fault in one of the two main actuators, hence it detects an abnormal condition in the actuator's dynamics, two possible situations can happen:

- the detected fault do not result in a failure since abnormality depended from external causes (e.g., magnetic-field that interfered with the electronic subsystem of the actuator);
- the actuator is in failure. The FDI switches off the failed actuator and activates the reserve one. It's trivial to notice that the AOCS is not in failure even if the unit level is. Furthermore, if two actuators out of three are in failure, the satellite is expelled from the communication: the spacecraft is considered in failure but the entire system sees it as a fault and continue to perform its actions.

5.1 Fault causes of momentum exchange devices

In an AOCS, the impact of a failure strongly depends on the subsystem in which it occurs. Failures affecting thrusters, gyroscopes, or faults of unknown origin generally have the most severe consequences, as they compromise the primary actuation and attitude sensing capabilities of the spacecraft.

Conversely, failures in momentum exchange devices tend to have a comparatively lower impact on overall mission performance. Reported statistics indicate an impact of approximately 10% for momentum wheels, 6% for reaction wheels, and 4% for CMGs.

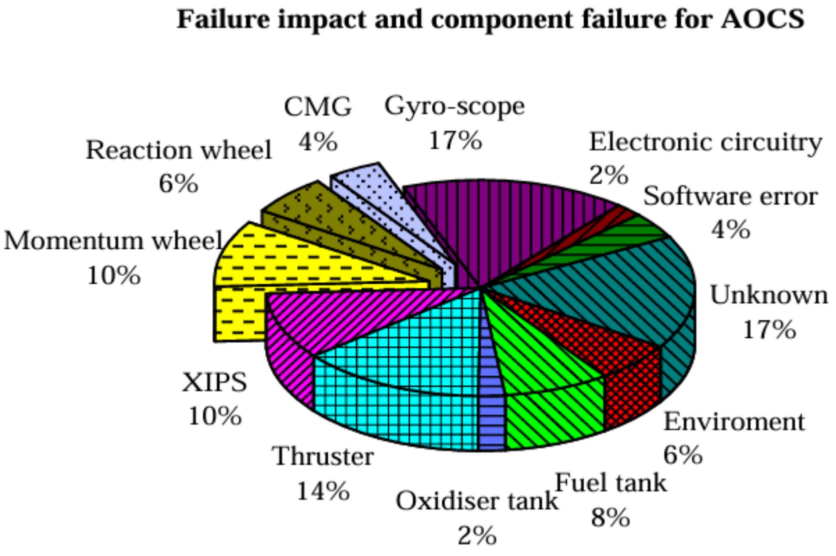

Figure 5.2. Failure impact and component failure for AOCS.

Here, a list of all possible faults that can lead to the failure of a momentum exchange device is presented:

- **Stator faults**: breakdown of the winding insulation in the region where the end-windings enter the stator slots;
- **Rotor faults**: damaged rotor magnet; partial demagnetization of the magnets; broken rotor bar; cracked rotor end-rings;
- **Eccentricity-related faults**: eccentricity between rotor and stator; ovality of the stator core; incorrect positioning of the rotor and stator;
- **Bearing and gearbox faults**: caused by continued stress, inherent eccentricity, fatigue and other external causes, such as unbalanced load, improper installation, contamination and corrosion, and improper lubrication;
- **Inverter faults**: loss of phase of one or more switches; short circuit of a switch; opening of one of the lines to the machine.

The GAFE methodology (Generic AOCS/GNC FDIR Engineering) developed by Airbus Defence & Space in collaboration with ESA provides a systematic framework for the design and verification of fault detection, isolation and recovery (FDIR) strategies in spacecraft AOCS/GNC systems. This methodology has been adopted as the reference model for this work and adapted to the specific case. FDI methodology's tasks are presented as follows:

1. **Analysis of Fault Management Requirements**;
2. **Extension of Nominal Equipment Set**;
3. **Definition & Implementation of FDI Concept**;

5.2 Analysis of Fault Management Requirements

The analysis of the fault management (FM) requirements of a space mission is the first step towards a robust FDI concept. Although the specific FM requirements differ from mission to mission, most of them belong to one of the following categories:

- **failure tolerance requirements**, define how many failures the system can withstand; can be either fail-operational or fail-safe requirements;
- **availability requirements**, typically related to fail-operational and are based on the percentage of mission time they should be available;
- **best-practice requirements**;
- **reliability requirements**, related to the probability that the system operates for the entire mission without failing;
- **survivability requirements**, typically related to fail-safe, specifying the capability of the system to overcome critic conditions.

A further discussion about the first type of requirements listed above is presented: the failure tolerance requirements are part of the customer specification of a space mission and define the number and type of failures the system to be designed and developed must be able to cope with.

There are different characterizations of the type of failures. They are often linked to consequences which have to be avoided in case the failure happens. They are usually expressed similar to:

- **Single Failure Tolerance**: *each satellite shall be able to sustain a single failure or operator error without critical or catastrophic consequences*;
- **Two Failure Tolerance**: *no single failure or operator error shall have major or critical or catastrophic consequences.*

In this work, a Single Failure Tolerant FDI subsystem for each spacecraft is modeled.

5.3 Extension of Nominal Equipment Set

The Nominal Equipment Set (NES) comprises the whole set of AOCS equipment required to fulfill the objectives of the mission in the absence of faults. This set is assumed to be minimal: it contains no single element which could be left out without violating any functional and/or performance requirement, i.e., it is not tolerant against any failure. In this work, the extension of the NES consists in a reserve DGVSCMG for each satellite, in order to cope with the failure of one of the main two actuators.

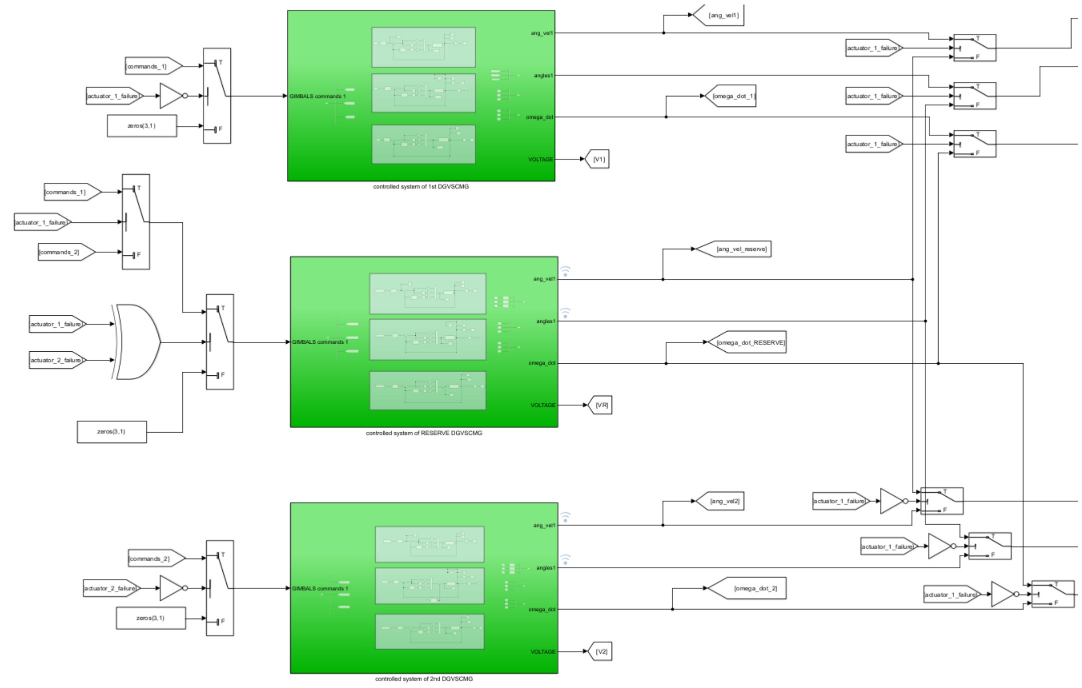

Figure 5.3. Extension of NES: reserve DGVSCMG.

In the figure above, from up to down, the 1st, the reserve and the 2nd DGVSCMGs.

5.4 Definition and Implementation of FDI Concept

There are different method of fault detection:

- investigate a single variable and check for plausibility of the values, through an asymptotic observer that provides an estimate of the variable, through a model that provides the expected evolution of the variable, or through the comparison of measurement of the variable with two or more sensors;
- detect discrepancies in variables connected to each other through constraints, checking the correctness of the variables exploiting the model that relates them.

This leads to the definition of the residual, which is function of the variables analyzed. In a fault free situation (neglecting noise and transitory events) a residual has to be zero. If any discrepancy in one of the variables occurs, it will deviate from zero, which gives the possibility of detecting the failure.

The design of the FDI in this implementation is divided in:

1. **Residuals' computation** block: once the residuals are computed, they are compared to some threshold values, chosen by trial and error procedure;
2. **Data Fusion** block: if the comparison reveals an abnormal behavior, a weighted sum of the residuals is calculated and compared to a threshold value;
3. **Fault Detection** block: if the fusion of the residuals overcomes the limit, the failure of the actuator is detected if the other spacecrafts' FDI systems are not in fault (faults occurring simultaneously are considered as not credible);
4. **Isolation and Reconfiguration**: the actuator in fault is isolated and the reserve one is activated. If two actuators over three are in fault, the spacecraft is expelled from the communication. A new communication topology (that respects the *strong connectivity* condition) is computed and provided to the controller by the *switching topologies* function. The new topology is a connected undirected graph: each spacecraft exchanges informations with the others.

Figure (5.4) shows the FDI implementation for each spacecraft. Each yellow block is a stateflow chart containing the logic that implements the procedure described above: from up to down, the FDI system of the 1st main actuator, of the 2nd main actuator and of the reserve actuator.

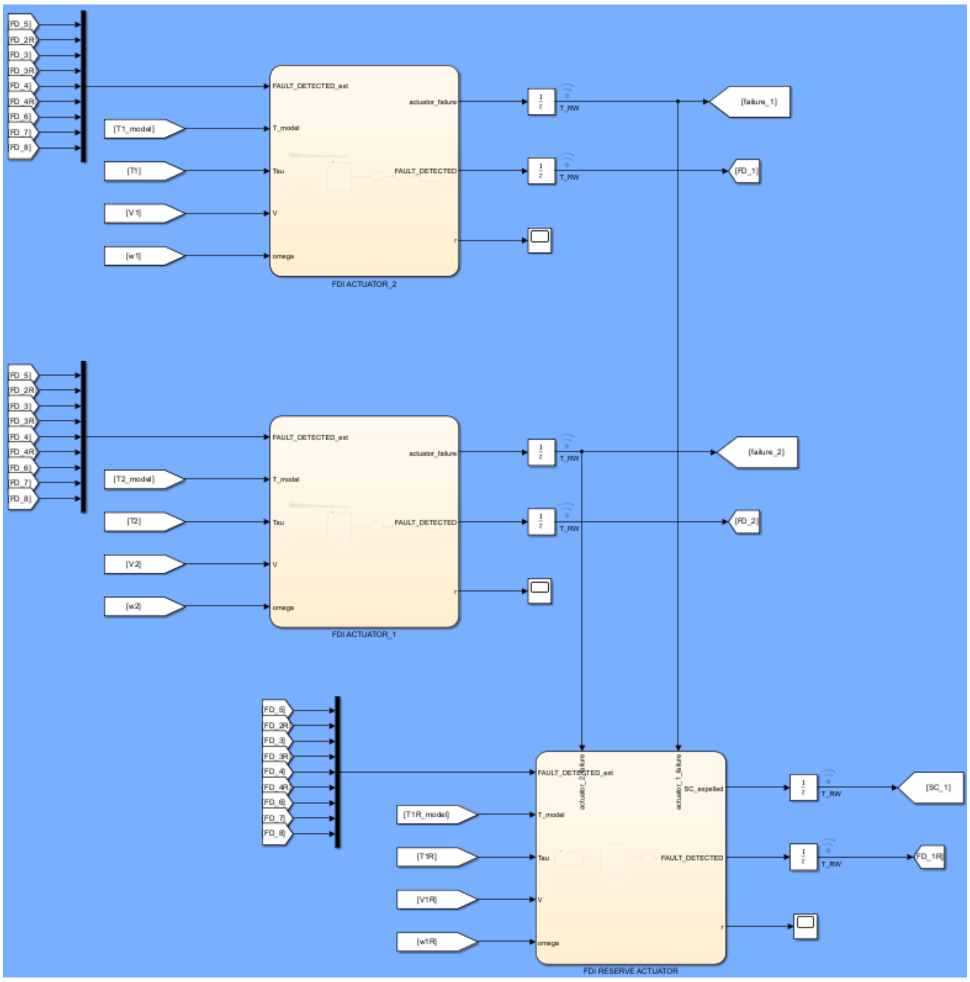

Figure 5.4. FDI implementation of the two main actuators and reserve actuator of one S/C.

5.4.1 Residuals' computation

A residual can be:

- a **cross-check residual**, if it only involves measurement constraints;
- a **model-based residual**, if there are analytical constraints involved between the variables.

In this implementation, two residuals are computed for each FDI subsystem:

- the **torque cross-check residual**, a three-component vector obtained as the absolute difference between the measured actuator torque and the expected torque;
- the **voltage model-based residual**, a three-component vector, defined as

$$R_V = |V - k_e\,\omega|$$

where V is the vector of controller output voltages (one per actuator axis), k_e is electric constant, and ω is the angular velocity vector reached by each axis of rotation of the actuator.

The relation between voltage and angular velocity holds only at steady state. During the transient phase, this relation is temporarily violated, which would produce non-zero residuals even in the absence of a fault. To prevent the FDI logic from incorrectly interpreting these transient deviations as faults, both the voltage and angular velocity signals are passed through a low-pass filter, with time constant $\tau = 0.01s$, before the residual computation, thus attenuating the transient components.

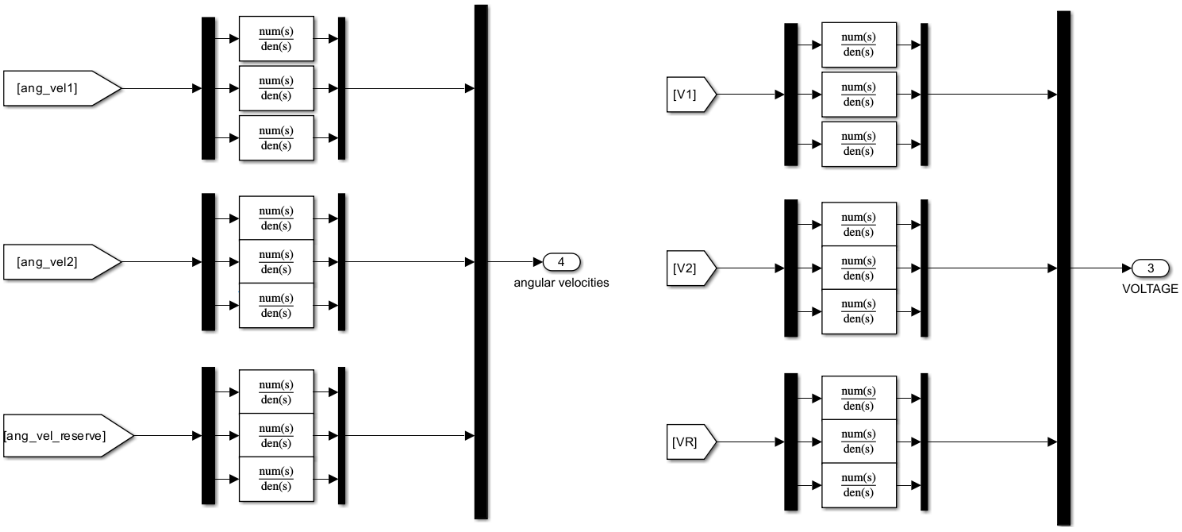

Figure 5.5. Filtered voltage and angular velocity signals before the residuals' computation block.

Furthermore, the residual thresholds are selected through a trial-and-error tuning procedure, ensuring that all residual deviations caused by nominal transient behavior, rather than by actual faults, remain below the detection threshold.

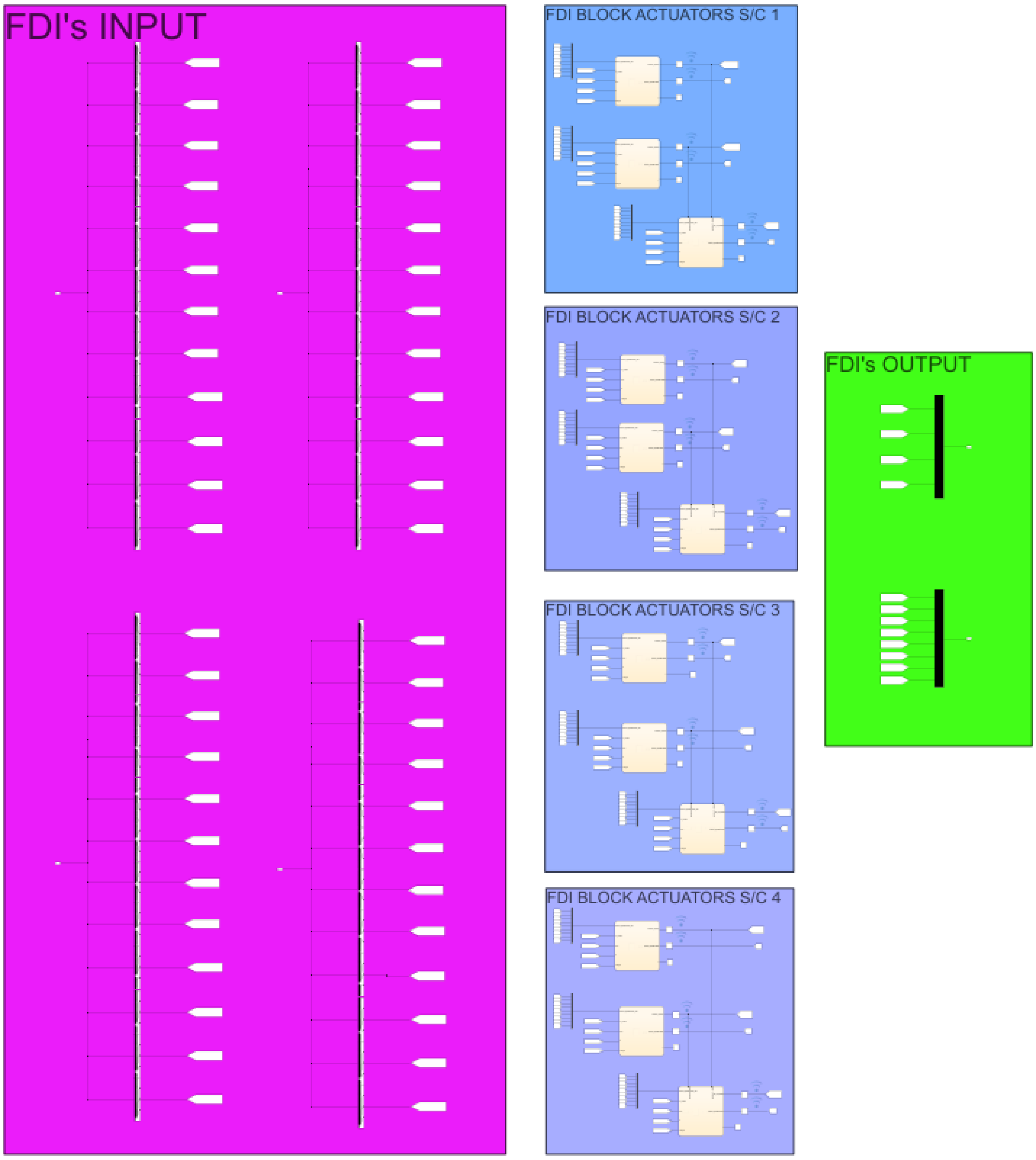

Figure 5.6. Entire FDI system.

The results of the simulation of the FDI system in presence of one or more actuators' failures are reported in subsection 6.2.5.

Chapter 6

Multi Sliding Mode Control

Multi Sliding Mode Control (MSMC) is a multi-agent control strategy based on Sliding Mode Control (SMC). A brief overview of SMC will be presented in the following sections.

6.1 Sliding Mode Control

Sliding Mode Control (SMC) is a nonlinear control technique characterized by robustness against model uncertainties and external disturbances, as well as for its fast convergence properties. The main idea of SMC is to design a control law that drives the system state onto a predefined manifold, called the **sliding surface**, and then forces the system to evolve along it until the desired state is reached.

Consider a nonlinear system of the form

$$\dot{x} = f(x) + g(x)u, \tag{6.1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input, and $f(x)$ and $g(x)$ are nonlinear functions. The sliding surface is defined as a manifold in the state space expressed as

$$\mathcal{S} = \{x \in \mathbb{R}^n : \sigma(x) = 0\}, \tag{6.2}$$

where $\sigma(x) \in \mathbb{R}^m$ is the sliding variable, constructed as a linear combination of the system state. Sliding mode control consists of two main phases:

- the *reaching phase*, to drive the system trajectories onto the surface $\mathcal{S}$;
- the *sliding phase*, to maintain the motion on $\mathcal{S}$, where the system exhibits the desired dynamics.

6.1.1 Sliding Mode Control Law

The full Sliding Mode Control input is usually composed of two terms:

$$u = u_{eq} + u_{sw}, \tag{6.3}$$

where:

- u_{eq} is the equivalent control;

- u_{sw} is the discontinuous term which purpose is to drive the system trajectories towards the sliding surface.

  A common choice for the switching control term is:

$$u_{sw} = -K\,\mathrm{sign}(\sigma), \tag{6.4}$$

where K is a positive definite gain matrix and $\mathrm{sign}(\cdot)$ denotes the *sign* function applied component-wise. This discontinuous term guarantees one of the fundamental properties of the sliding surface: the **attractiveness**, that ensures that system trajectories converge to $\mathcal{S}$ in finite time.
Moreover, since $\dot{\delta} = 0$, the sliding surface is said to be **invariant**.

6.1.2 Design of the sliding surface

The sliding surface is a manifold in the state space chosen such that motion on the manifold yields the desired closed-loop behaviour. Mathematically, the sliding surface is defined as

$$\mathcal{S} \;=\; \{x \in \mathbb{R}^n \;:\; \sigma(x) = 0\},$$

where $\sigma(x) \in \mathbb{R}^r$ is the sliding function. In general, the sliding surface is constructed as a suitable linear combination of the system states. For a second-order system, a common choice is

$$\sigma(x) = \dot{e}(t) + \lambda e(t), \tag{6.5}$$

where $e(t)$ represents the tracking (or consensus) error, $\dot{e}(t)$ its time derivative, and $\lambda > 0$ is a design parameter.

This construction can be interpreted as a state transformation that forces the system motion to evolve along a specific direction in the state space. More precisely, the sliding surface defines a reduced-order dynamics: once $\sigma = 0$ is reached and maintained, the system behavior is governed by the differential equation

$$\dot{e}(t) + \lambda e(t) = 0. \tag{6.6}$$

The solution of this equation is given by

$$e(t) = e(0)e^{-\lambda t}, \tag{6.7}$$

which guarantees exponential convergence of the error to zero if and only if $\lambda > 0$.

Therefore, the choice $\lambda > 0$ ensures the internal stability of the sliding dynamics,that converges to zero, whereas a negative value of λ would lead to exponentially diverging trajectories.

6.1.3 SMC design

To design a Sliding Mode Controller, it is necessary to determine the **relative degree** of the system with respect to the output $\sigma(x)$.

The relative degree γ is defined as the minimum number of time derivatives of $\sigma(x)$ required until the control input u explicitly appears:

$$\frac{d^\gamma}{dt^\gamma}\sigma(x) = \Phi(x) + \Psi(x)u, \tag{6.8}$$

where $\Psi(x) \neq 0$. When the input appears linearly in the system equations, the system is said to be *affine* in u. In many practical applications, including spacecraft attitude dynamics, the sliding variable is chosen such that $\gamma = 1$, i.e.,

$$\dot{\sigma}(x) = \frac{\partial \sigma}{\partial x}\dot{x} = L_f\sigma(x) + L_g\sigma(x)u, \tag{6.9}$$

where L_f and L_g denote the Lie derivatives of σ along $f(x)$ and $g(x)$, respectively, and $L_g\sigma(x) \neq 0$.

The **equivalent control** u_{eq} is defined as the continuous control law that keeps the system on the sliding surface once it is reached, i.e. it satisfies the condition

$$\dot{\sigma}(x) = 0. \tag{6.10}$$

Assuming $L_g\sigma(x)$ is invertible, the equivalent control can be computed as

$$u_{eq} = -[L_g\sigma(x)]^{-1}L_f\sigma(x). \tag{6.11}$$

This represents the control action required to maintain the sliding condition in ideal conditions, without disturbances.

6.1.4 Chattering and mitigation techniques

A classical SMC switching term is

$$u_{sw} = -K\ \mathrm{sign}(\sigma),$$

with $K > 0$. While this discontinuous action guarantees finite-time reachability of $\mathcal{S}$, it induces high-frequency switching of the actuator command, the phenomenon known as *chattering*. In real applications chattering is harmful because it excites unmodelled high-frequency dynamics, causes wear on actuators and may lead to performance degradation.

Common countermeasures include:

- **Boundary layer**: replace the sign function with a saturation function

  $$\mathrm{sat}\Big(\frac{\sigma}{\phi}\Big) = \begin{cases} \mathrm{sign}(\sigma), & |\sigma| > \phi, \\ \dfrac{\sigma}{\phi}, & |\sigma| \leq \phi, \end{cases}$$

  so that

  $$u_{sw} = -K\ \mathrm{sat}\Big(\frac{\sigma}{\phi}\Big),$$

  where $\phi > 0$ is the boundary-layer thickness. This yields continuous control inside the layer and reduces switching frequency at the cost of a finite steady-state error bounded by ϕ.

- **Smooth approximations** such as the hyperbolic tangent:

  $$u_{sw} = -K \tanh\Big(\sigma\eta\Big),$$

  where $\eta > 0$ is a design parameter. Computing the limit of the function $\tanh \cdot$:

  $$\lim_{\eta\to\infty} \tanh\Big(\sigma\eta\Big) \to \operatorname{sign}\Big(\frac{\sigma}{\phi}\Big) \tag{6.12}$$

  one can observe that η should be chosen high. The function $\tanh(\cdot)$ provides a smooth transition and reduces chattering while keeping small tracking error for sufficiently small η.
- **Higher-order sliding modes (HOSM)**: algorithms like the super-twisting controller deliver continuous control signals (no ideal discontinuity) while retaining finite-time convergence of σ and its derivatives. HOSM methods are generally more complex to design but are effective in eliminating chattering caused by ideal switching.

6.1.5 Types of SMC

Other types of SMC that differ from the conventional one are:

- **Higher-order SMC (HOSM)**, e.g. super-twisting, which act on derivatives of σ to obtain continuous control effort;
- **Adaptive SMC**, where gains are adapted online to cope with disturbance bounds;
- **Integral Sliding Mode**, where an integral term is included in the surface design.

6.2 Multi-Spacecraft Sliding Mode Control

After introducing the general principles of Sliding Mode Control (SMC), this section focuses on its extension to multi-agent spacecraft formations, referred to as *Multi-Spacecraft Sliding Mode Control* (MSMC). The implemented GNC system, shown in Figure(6.1), is constituted by a network of four spacecrafts controlled by a MSMC (the yellow block in Figure), that compute the control input as function of the difference between the reference provided by the Guidance block (blue block) and the actual state of the system. The Guidance block is constituted by a kinematic block that computes the desired quaternion as function of the desired angular velocity when it is provided (tracking problems). The green blocks are the actuators, which FDI integrated system is modeled in the magenta block. The matlab functions on the right computes the equivalent input for each spacecraft, according to Eq.(6.22).

The considered communication topology is a spanning tree with the first spacecraft as the root node. The related Laplacian matrix is:

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 2 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix} \tag{6.13}$$

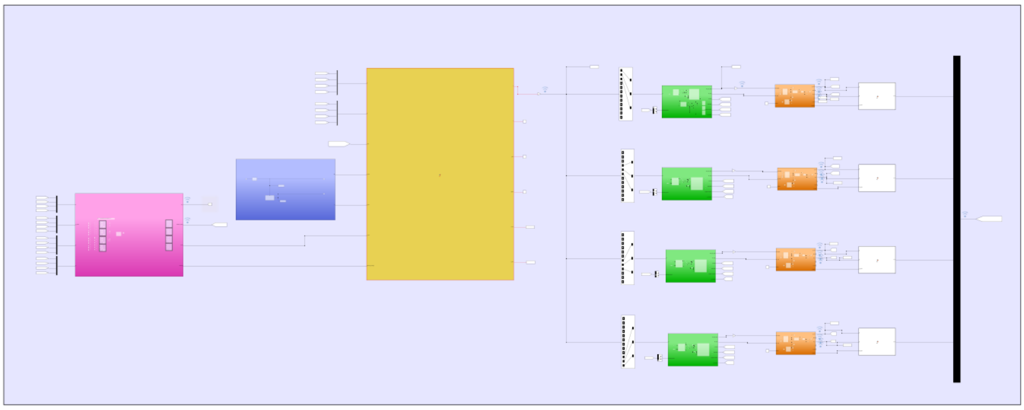

Figure 6.1. GNC system with integrated FDI subsystem.

The corresponding eigenvalues are

$$\lambda(\mathbf{L}) = \{\,2{,}1{,}1{,}0\,\},$$

6.2.1 Multispacecraft Sliding Surface Design

Let us consider a formation of n spacecraft. For the i-th spacecraft, the attitude is described by the quaternion $\tilde{q}_i \in \mathbb{R}^3$ (vector part only) and the body angular velocity $\tilde{\omega}_i \in \mathbb{R}^3$. The multispacecraft sliding mode vector is defined as:

$$S = [s_1, \dots, s_n]^T, \tag{6.14}$$

where each sliding variable $s_i \in \mathbb{R}^3$ is given by

$$s_i = b_i(\tilde{\omega}_i + C\tilde{q}_i) + \sum_{j=1}^{n} a_{ij}\left[(\tilde{\omega}_i - \tilde{\omega}_j) + C(\tilde{q}_i - \tilde{q}_j)\right], \quad j \neq i, \tag{6.15}$$

where:

- $C \in \mathbb{R}^{3\times 3}$ is a positive definite gain matrix,
- $b_i \geq 0$ is the attitude tracking weight for spacecraft i,
- $a_{ij} \geq 0$ is the *interaction parameter* that describes the communication from agent i to agent j.

In vector form, the sliding surface can be rewritten as

$$S = [(L + B) \otimes I_3]\left(\tilde{\Omega} + CQ\right), \tag{6.16}$$

where:

- $L \in \mathbb{R}^{n\times n}$ is the graph Laplacian of the communication topology,

- $B = \text{diag}(b_1, \ldots, b_n)$,
- I_3 is the 3×3 identity matrix,
- $\tilde{\Omega} = [\tilde{\omega}_1, \ldots, \tilde{\omega}_n]^T \in \mathbb{R}^{3n}$,
- $Q = [\tilde{q}_1, \ldots, \tilde{q}_n]^T \in \mathbb{R}^{3n}$,
- $\tilde{C} = \text{diag}(C, \ldots, C) \in \mathbb{R}^{3n \times 3n}$.

The symbol $\otimes$ denotes the Kronecker product, used to extend the network Laplacian matrix L, originally defined for scalar consensus variables, to the vector-valued attitude dynamics. This allows the same communication topology and weights to be consistently applied to all three rotational components of each spacecraft sliding surface.

Two main configurations arise depending on the value of matrix B:

- *pure consensus*, when $B = 0$;
- *reference tracking*, when $B \neq 0$.

6.2.2 Pure Consensus

If $B = 0$, the sliding surface reduces to:

$$S = (L \otimes I_3)\,(\tilde{\Omega} + \tilde{C}Q). \tag{6.17}$$

In this configuration, no external reference is provided to the agents and the network evolves only according to the relative information exchanged through the Laplacian matrix L. As a consequence, the spacecraft converge towards a common attitude and angular velocity, achieving a pure consensus state.

If the communication is represented through a spanning tree graph, the final consensus state coincides with the initial conditions of the *root* agent. More generally, for different graph structures, the consensus value depends on the network topology.

6.2.3 Reference Tracking

If $B \neq 0$, an external reference is introduced in the control law and each spacecraft is directly driven towards a desired attitude and angular velocity.

In this case, the diagonal matrix B defines the contribution of the reference signal to each agent: large values of B increase the influence of the reference with respect to the consensus coupling, while smaller values emphasize the mutual interaction among agents.

The presence of the Laplacian matrix L in the sliding surface still plays a key role: even though each spacecraft tracks the reference independently, the interconnection defined by L enforces coordination among agents, reducing relative errors and improving convergence speed and coherence of the overall formation response.

Therefore, the reference tracking configuration can be interpreted as a compromise between individual tracking and cooperative synchronization. The Laplacian-based coupling acts as an additional mechanism to enhance robustness and transient performance with respect to disturbances and modeling uncertainties.

Hence, matrix B can be interpreted as a tuning parameter that blends consensus and tracking behaviors, allowing the designer to modulate the level of cooperation versus individual reference following within the multi-spacecraft network.

6.2.4 Control Law

According to Eq.(6.3), the control law is composed by two terms. The switching term u_{sw} exploited in this implementation is

$$u_{sw} = -k \cdot \tanh(\eta\sigma) \tag{6.18}$$

where the trigonometric function is chosen to mitigate chattering phenomenon. The equivalent term u_{eq} is computed considering the derivative of the sliding variable for each spacecraft. The sliding variable for each spacecraft is defined in Eq.(6.16). Its derivative is equal to:

$$\dot{\sigma} = [(L + B) \otimes I_3] \left(\dot{\tilde{\Omega}} + C\dot{Q} \right), \tag{6.19}$$

where:

- $\dot{\tilde{\Omega}} = [\dot{\tilde{\omega}}_1, \ldots, \dot{\tilde{\omega}}_n]^T \in \mathbb{R}^{3n}$,
- $\dot{Q} = [\dot{\tilde{q}}_1, \ldots, \dot{\tilde{q}}_n]^T \in \mathbb{R}^{3n}$,

The rotational dynamics of each spacecraft is given by:

$$\dot{\omega}_B = J^{-1}(\tau - \omega_B \times (H_B + H_{CMG}) - \delta^T \ddot{\eta}), \tag{6.20}$$

where:

- J is the inertia matrix,
- τ is the control torque,
- H_B is the body angular momentum,
- H_{CMG} is the control moment gyro angular momentum,
- the term $\delta^T \ddot{\eta}$ represents flexible dynamics, neglected in the equivalent control input as its contribution is assumed to be sufficiently small.

Imposing the sliding condition:

$$\dot{\sigma}_i = 0, \tag{6.21}$$

and using the spacecraft dynamics, the equivalent control torque $\tau_{eq,i}$ can be derived as:

$$\tau_{eq,i} = \omega_i \times (H_{B,i} + H_{CMG,i}) - CJ\dot{\tilde{q}}_i. \tag{6.22}$$

Moreover, using the quaternion kinematics equation:

$$\dot{q} = \frac{1}{2}\Omega(\omega)q, \tag{6.23}$$

where $\Omega(\omega)$ is the skew-symmetric matrix of the angular velocity of the spacecraft.

The global equivalent control input for the formation is then obtained by stacking each individual torque:

$$\tau_{eq} = \begin{bmatrix} \tau_{eq,1} \\ \tau_{eq,2} \\ \vdots \\ \tau_{eq,n} \end{bmatrix} \in \mathbb{R}^{3n}. \tag{6.24}$$

through a multiplexer that collects each input component.

6.2.5 Simulation Results

In this section, the simulation results related to the attitude control performance of the multi-spacecraft system are presented. Four different simulation scenarios are considered to assess the performance and robustness of the proposed multi-spacecraft attitude control framework:

- a *regulation problem*, where the objective is to drive the spacecraft attitude to a constant desired configuration. Although the reference attitude is conceptually defined by an external operator, in the simulation environment it is generated through a kinematic block, which integrates the desired angular velocities and provides the corresponding quaternion trajectory over time;
- a second scenario is devoted to the validation of the *Fault Detection and Isolation (FDI)* strategy. In this case, a regulation maneuver is performed while an actuator failure is artificially introduced during the simulation. The resulting behavior of the system, together with the response of the FDI module, is analyzed to evaluate the capability of detecting, isolating, and coping with actuator malfunctions.
- an *attitude tracking problem*. In this case, unlike the regulation task, the reference is time-varying and defines a dynamic trajectory rather than a fixed equilibrium point. The controller is therefore required to continuously track the desired attitude evolution;
- a *pure consensus problem*. By setting the matrix $\mathbf{B}$ in the sliding surface definition equal to zero, the leader contribution is removed, and the network dynamics are governed solely by the inter-satellite communication topology. Under these conditions, the multi-spacecraft system is expected to asymptotically converge to a common attitude configuration, determined by the consensus protocol;

For robustness assessment, the disturbance was modeled as a sinusoid of amplitude $10^{-2} Nm$ and frequency 10 rad/s. This choice is intentionally conservative: typical environmental torques in LEO are of the order of $10^{-6} - 10^{-4} Nm$, hence the adopted amplitude exceeds realistic values by 2-4 orders of magnitude.

Regulation problem

Considering the controller parameters and references reported in Table(6.1), all network's spacecrafts converge to the desired attitude and angular velocity (as shown in Figures(6.2),(6.3)).

Parameter	Symbol	Value	Unit of measure
control input gain	k	80	-
lambda	λ	0.8	-
sliding weight matrix	B	diag([ones(12,1)]	-
desired angular velocity	w	[0;0;0]	$[deg/s]$
desired quaternion	q	[1;0;0;0]	-
chattering mitigation parameter	η	100	-

Table 6.1. Controller parameters for regulation problem.

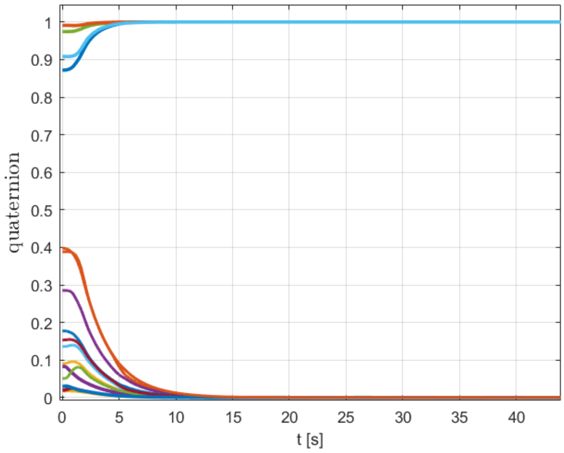

Figure 6.2. Quaternion evolution of all the network's agents.

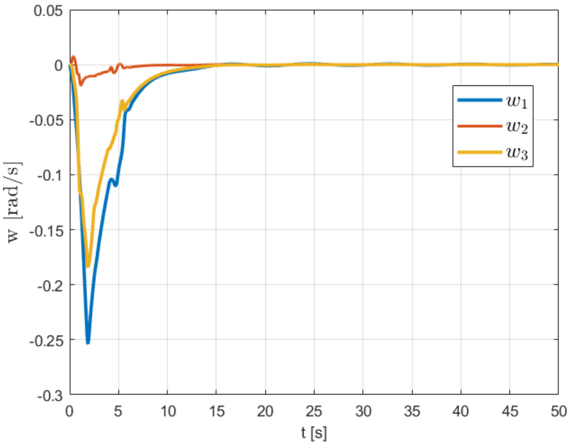

Figure 6.3. Angular velocities of all the network's agents.

It can be observed that a very good response is achieved:

- the steady-state error is equal to zero;
- the settling time with a 5% of tolerance band ($t_{s,5\%}$) is equal to 6.3 s.

As shown in Fig. 6.4, each component of the angular velocity vector converges asymptotically to zero, as expected in the desired equilibrium configuration. At the same time, the attitude quaternion components converge to the desired reference, corresponding to $q_d = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$. In Figure(6.5) q_0 is the scalar component of the current quaternion, while $[q_x, q_y, q_z]^T$ is its vectorial part.

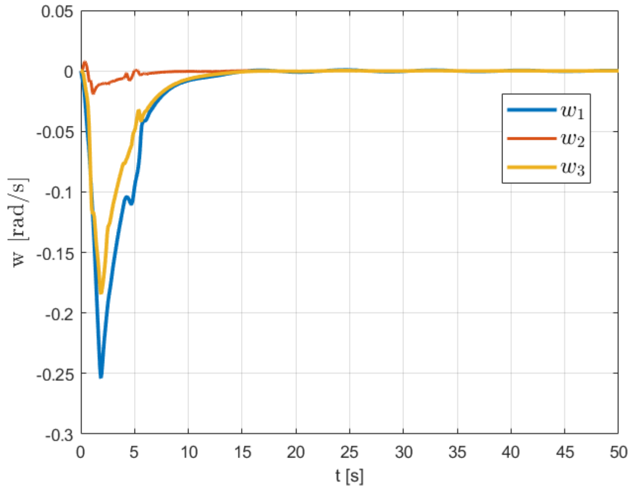

Figure 6.4. Angular velocity components of the spacecraft.

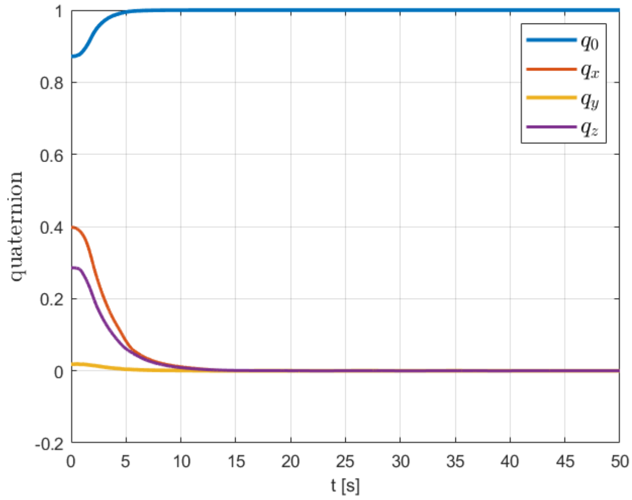

Figure 6.5. Quaternion components evolution of the controlled system.

For brevity, only the details of the simulation of one satellite of the network are shown and explained in the following. Figures(6.6),(6.7),(6.8) show the comparison between the control input computed by the controller and the effective torque delivered by the actuators. A slight attenuation of the control signal can be observed, especially during rapid transients, due to the low-pass filters included in the actuator dynamics, which model the physical inertia of the actuators.

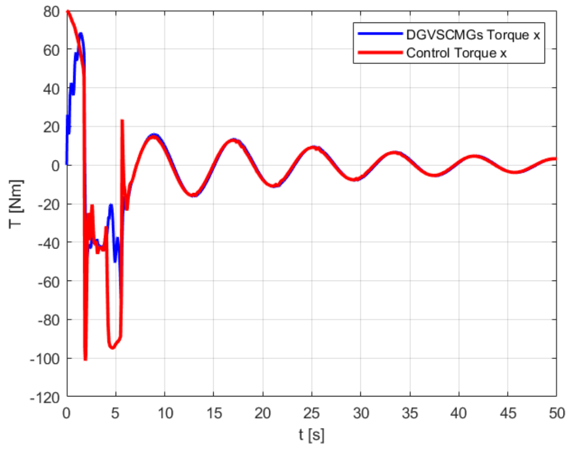

Figure 6.6. Comparison between x-component of commanded control torque and actual actuator output torque of the first spacecraft.

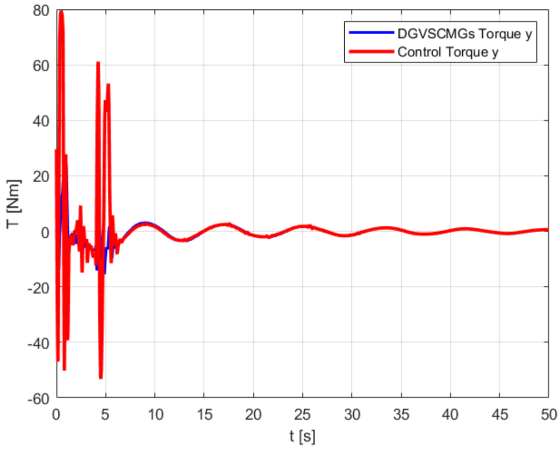

Figure 6.7. Comparison between y-component of commanded control torque and actual actuator output torque of the first spacecraft.

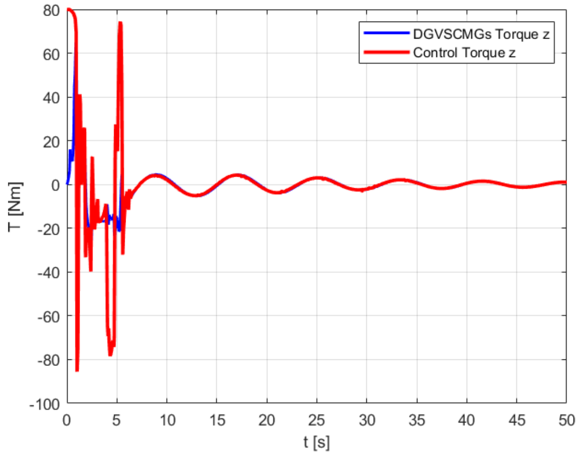

Figure 6.8. Comparison between z-component of commanded control torque and actual actuator output torque of the first spacecraft.

Moreover, the torque components along the body axes x, y, and z are reported for the CMG and the RW configurations in Figures(6.10),(6.11) and (6.12). The required torque is predominantly provided by the CMG-based configuration, except when the actuator is in the vicinity of singular configurations: as one can observe from Figure(6.9), the minimum singular value drops below the chosen singularity threshold, triggering the singularity-avoidance mechanism that involves the RW-based configuration of the actuator. Once the singular configuration is escaped, the singular value rises above the conservative band limit and the system re-enters the safe operational region where it exhibits exponentially decaying oscillations.

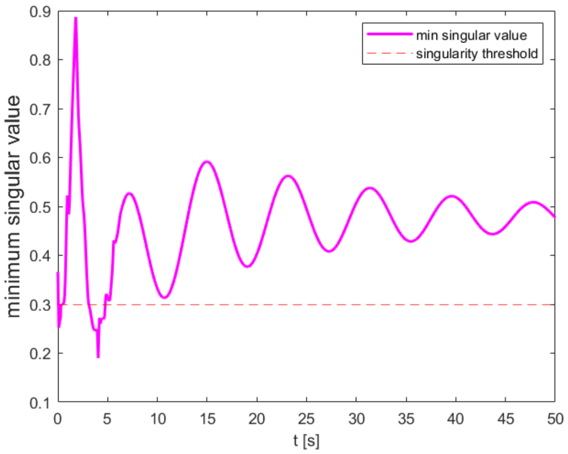

Figure 6.9. Minimum singular value's time evolution.

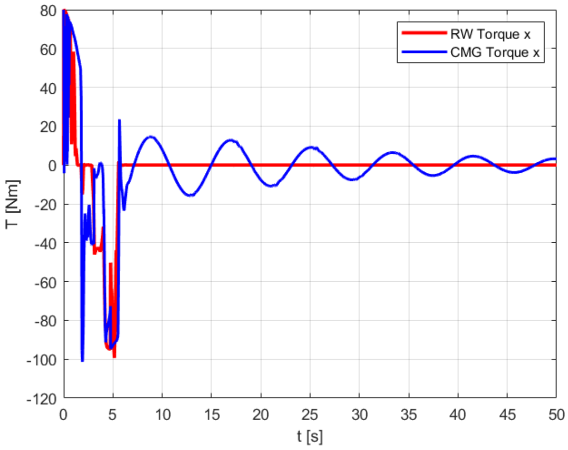

Figure 6.10. Comparison between x-component of the torques generated by CMG and RW configuration of the first spacecraft.

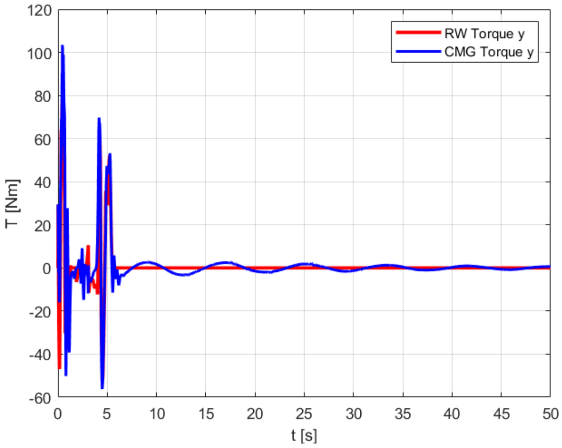

Figure 6.11. Comparison between y-component of the torques generated by CMG and RW configuration of the first spacecraft.

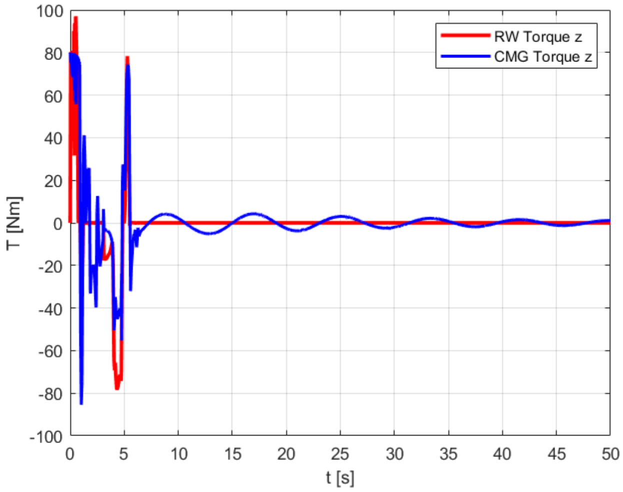

Figure 6.12. Comparison between z-component of the torques generated by CMG and RW configuration of the first spacecraft.

Finally, the desired and actual angular velocities of both the gimbal axes and the flywheels are presented, together with the desired and actual flywheel accelerations. The desired quantities are provided by the steering law, while the actual ones are the outputs of the actuator system, controlled by three separated PID controllers.

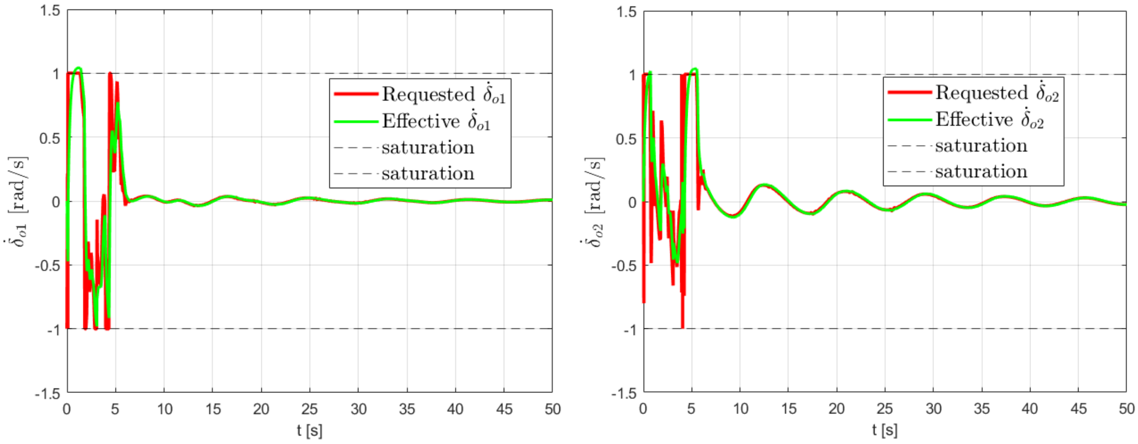

Figure 6.13. Desired and actual angular velocities of outer gimbals of actuators of the first spacecraft.

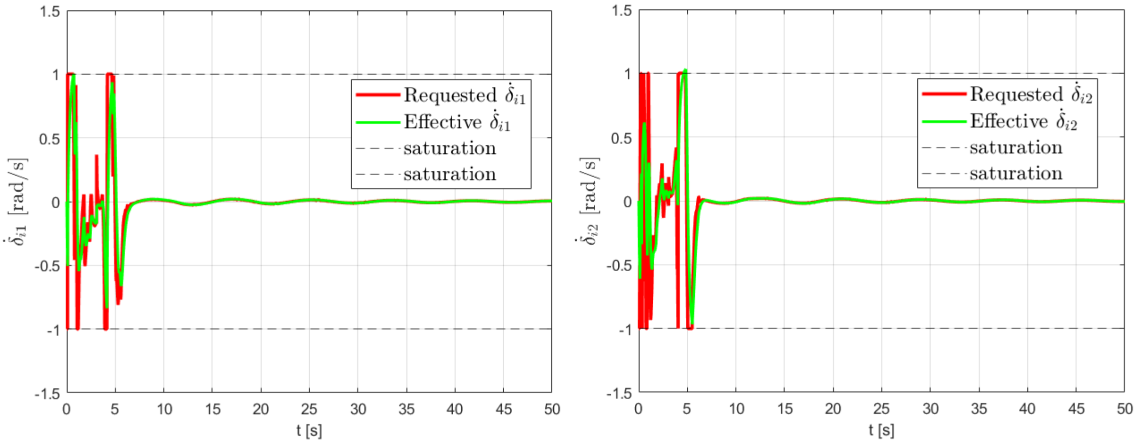

Figure 6.14. Desired and actual angular velocities of inner gimbals of actuators of the first spacecraft.

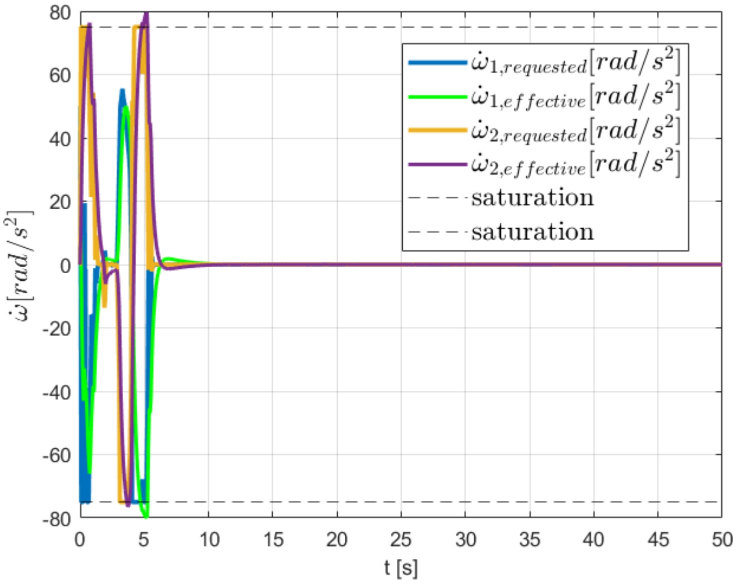

Figure 6.15. Desired and actual flywheel angular accelerations of actuators of the first spacecraft.

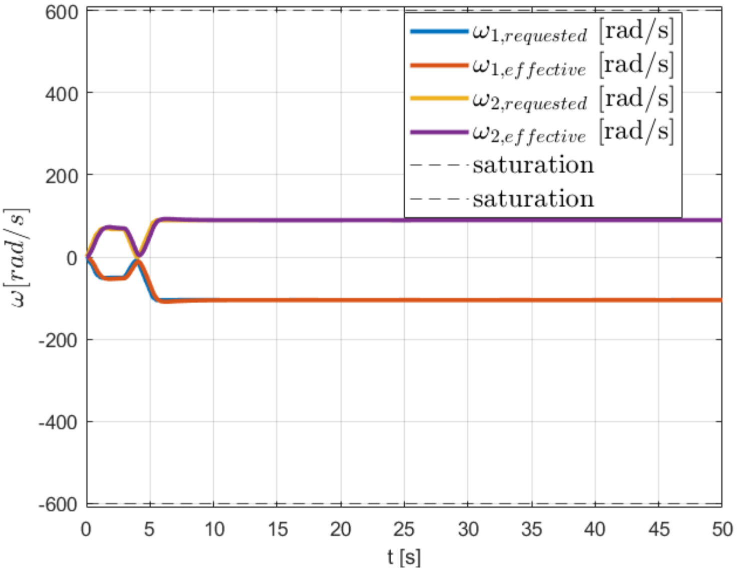

Figure 6.16. Desired and actual flywheel angular velocity of actuators of the first spacecraft.

Despite the presence of low-pass filters leads to small discrepancies and attenuations, it can be observed that the controlled system is able to track the reference signals provided by the steering law with good accuracy.

The flexible dynamics of the spacecraft are modeled using modal coordinates, according to Eq.(2.4). As highlighted before, the external modal torque acting on the flexible dynamics is generated by the rigid-body angular accelerations. However, as the attitude controller drives the spacecraft toward steady-state conditions, the angular acceleration tends asymptotically to zero.

During the transient phase the angular acceleration of the satellite is not negligible and exhibits a non-zero time evolution. Consequently, the flexible dynamics initially experience a *forced response* driven by this transient excitation.

Although the modal coordinates are initialized with zero initial conditions, i.e. $\eta(0) = [0;0;0;0]$ and $\dot{\eta}(0) = [0;0;0;0]$, the non-zero angular acceleration during the transient induces a modal excitation which generates non-zero modal amplitudes. Once the rigid-body acceleration vanishes, the external forcing tends to zero and only the free response of the flexible structure evolves in time.

Therefore, the subsequent evolution of the modal coordinates is governed by the homogeneous equation:

$$\ddot{\eta} + C\dot{\eta} + K\eta = 0$$

and is fully determined by the effective initial conditions reached at the end of the transient phase, consequence of the excitation induced by the rigid-body angular acceleration. Considering one component of the modal coordinate vector, η_i, the equation becomes:

$$\ddot{\eta}_i + 2\zeta_i\omega_i\dot{\eta}_i + \omega_i^2\eta_i = 0$$

where ω_i is the natural frequency of the i-th mode and ζ_i is its damping ratio. Considering the flexible parameters of Table (2.1), this is an underdamped case ($\zeta_i < 1$), the solution assumes the form:

$$\eta_i(t) = A_i\, e^{-\zeta_i \omega_i t} \cos(\omega_{d,i} t) + B_i\, e^{-\zeta_i \omega_i t} \sin(\omega_{d,i} t)$$

where:

$$\omega_{d,i} = \omega_i \sqrt{1 - \zeta_i^2}$$

is the damped natural frequency of the i-th mode of excitation. The constants A_i and B_i depend on the initial conditions $\eta_i(0)$ and $\dot{\eta}_i(0)$ of the free response.

One can observe the response of the flexible dynamics in Figure(6.17):

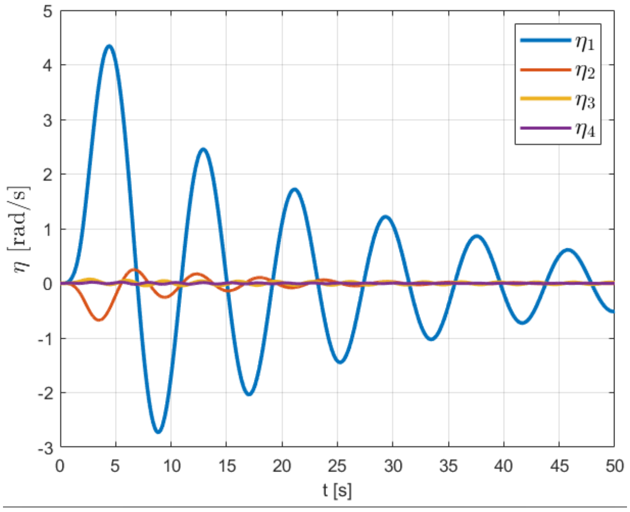

Figure 6.17. Flexible dynamics response of first spacecraft.

once the angular acceleration of the spacecraft vanishes, the free response of the flexible dynamics exponentially decays.

Regulation problem in case of fault detection

The objective is to demonstrate how the proposed AOCS reacts when an actuator in the network's agent fails and how the switching-topology function reconfigures the communication graph to preserve formation control among the remaining healthy satellites.

During the simulation it has been injected:

1. at $t = 1\,\mathrm{s}$, a fault on the control input of the second actuator (nominal actuator #2) of the first satellite (the root node of the spanning tree topology). This fault is modelled as an additive step on the actuator command, that is the voltage of the electric motor of one axis of the actuator.

2. a second fault on the reserve actuator of the same satellite, to simulate a double failure at the same platform and trigger topology reconfiguration.

After the topology reconfiguration triggered by the fault event, the switching topology function computes a new Laplacian matrix for the reduced communication network, given by

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}.$$

The corresponding eigenvalues are

$$\lambda(\mathbf{L}) = \{\, 0{,}3{,}3 \,\},$$

and the rank of the Laplacian is

$$\operatorname{rank}(\mathbf{L}) = N - 1 = 2,$$

where $N = 3$ is the number of agents in the reconfigured network. Hence, the Laplacian of the reduced communication graph remains strongly connected. Therefore, the necessary condition for achieving consensus over the new network topology is still satisfied.

The sequence of event is the following:

1. Immediately after the step fault on the nominal actuator, the residuals computed by the FDI module exceed their thresholds and a fault is flagged (see Figure(6.18));

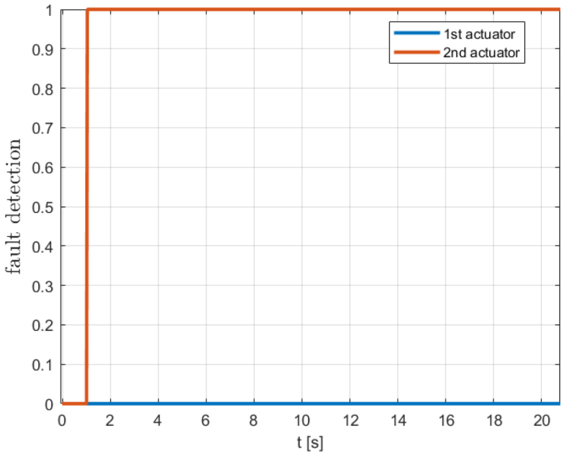

Figure 6.18. Fault detection of the second main actuator of the first spacecraft: fault detected equal to 1.

2. The redundant actuator is activated and replaces the failed actuator;

3. redundant actuator also fails (second injected fault): the switching-topology function then removes the faulty satellite from the communication topology;

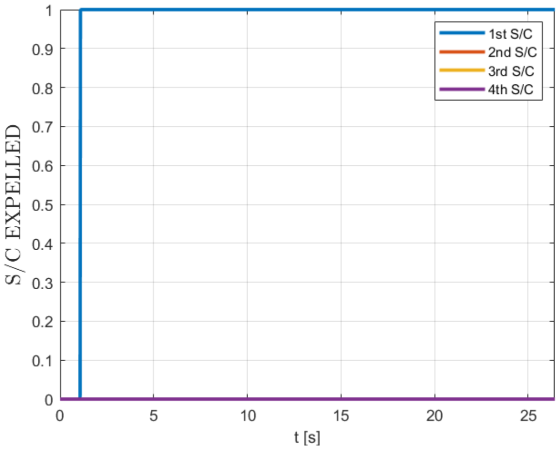

Figure 6.19. Expulsion of the first spacecraft from the network: S/C expelled equal to 1.

4. the remaining satellites resume regulation to the constant reference using the recomputed Laplacian and the MSMC law defined on the reduced network.

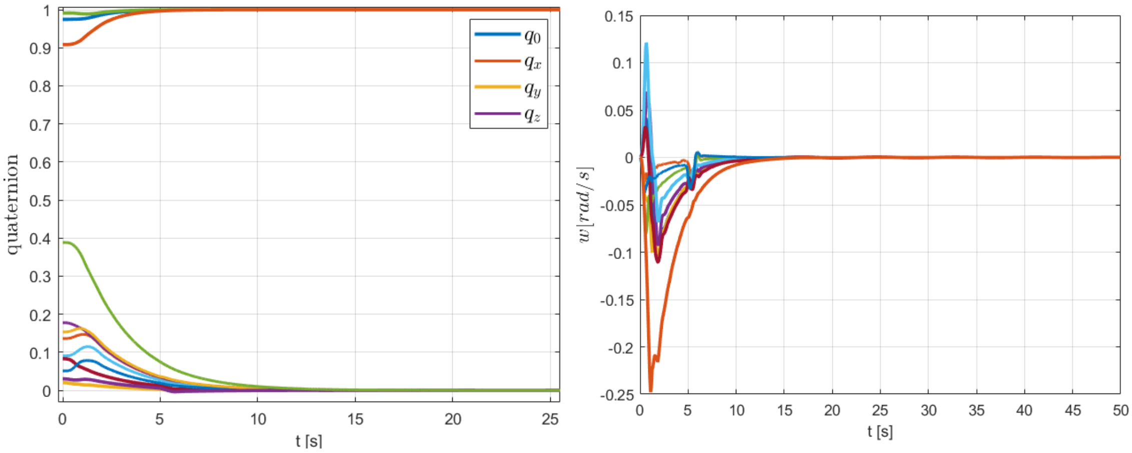

Figure 6.20. Quaternion and angular velocity components of the faulty satellite: after the expulsion it is not controlled by the MSMC.

The other agents, reorganized in a new network follow the reference.

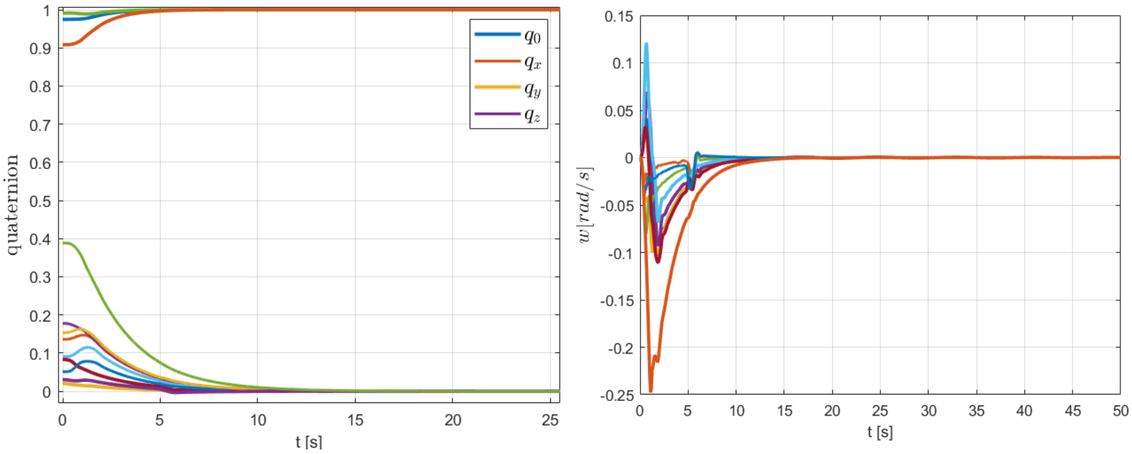

Figure 6.21. Quaternion and angular velocity components of the new network'agents.

Despite a slight degradation of the transient response can be observed, a very good response is achieved:

- the steady-state error is equal to zero;
- the settling time with 5% of band tolerance is equal to 6 s.

Tracking problem

In this scenario, the commanded angular velocity profile (that can be found in Table(6.2)) is fed into the kinematic block, which generates the corresponding desired quaternion trajectory. Figures 6.22 and 6.23 report the attitude evolution of all spacecraft in the formation. It is evident that all satellites successfully track the time-varying reference.

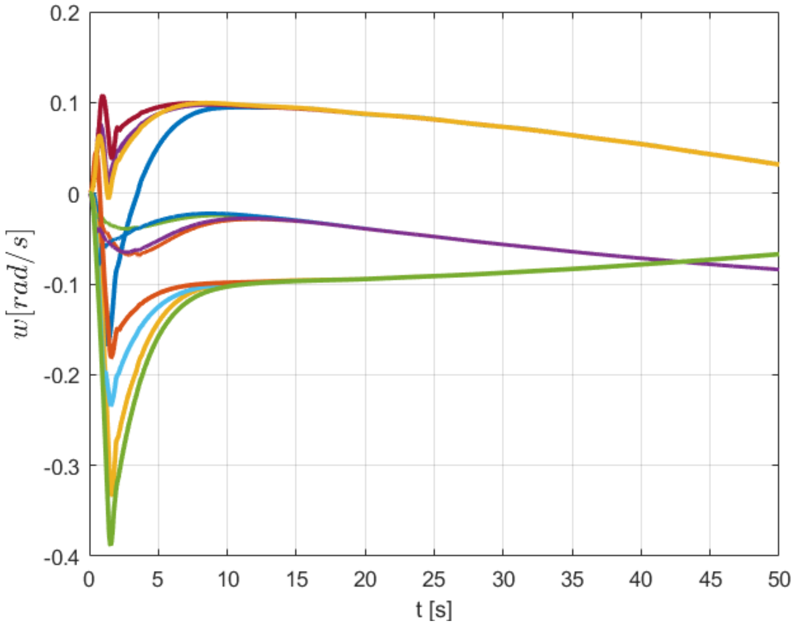

Figure 6.22. Global attitude tracking of the multi-spacecraft formation.

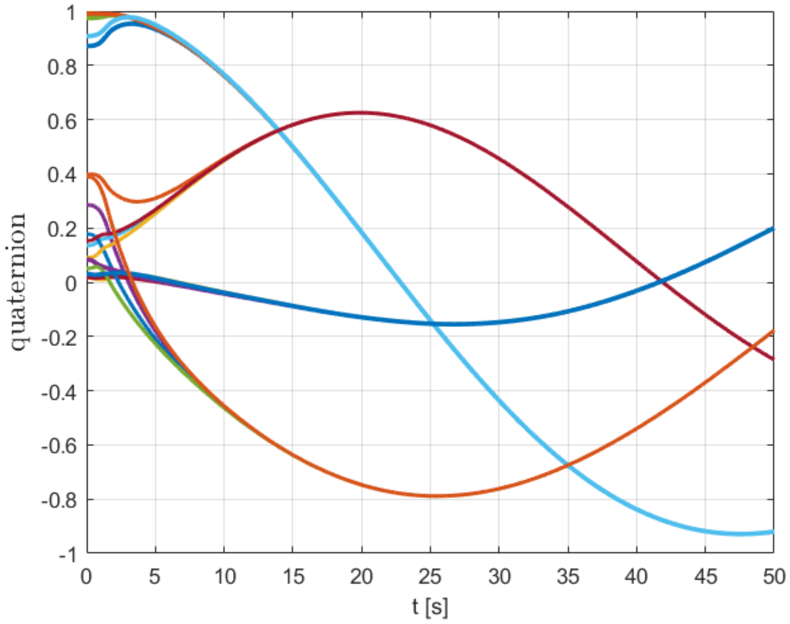

Figure 6.23. Global attitude tracking of the multi-spacecraft formation.

Parameter	Symbol	Value	Unit of measure
control input gain	k	70	-
lambda	λ	0.8	-
sliding weight matrix	B	diag([ones(12,1)]	-
chattering mitigation	η	100	-
desired angular velocity	w	$[0.1 \cdot cos(t/40); -0.1 \cdot sin(t/50); -0.1 \cdot cos(t/60)]$	$[deg/s]$

Table 6.2. Controller parameters for tracking problem.

For brevity, only the details of the simulation of one satellite of the network are shown and explained in the following. Figure 6.24 shows the comparison between the desired angular velocity of the spacecraft, provided by the guidance block, and its actual angular velocity. Figure 6.25 reports the time evolution of the actual and desired quaternion components. It can be observed that a very good tracking performance is achieved:

- the steady-state error is equal to zero;
- the settling time with 5% of band tolerance is equal to 7.5 s.

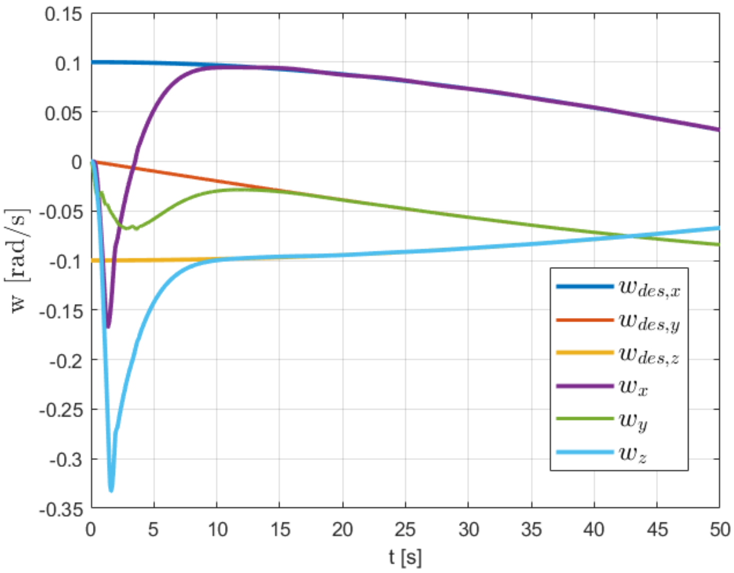

Figure 6.24. Desired vs actual angular velocity components for the tracking case.

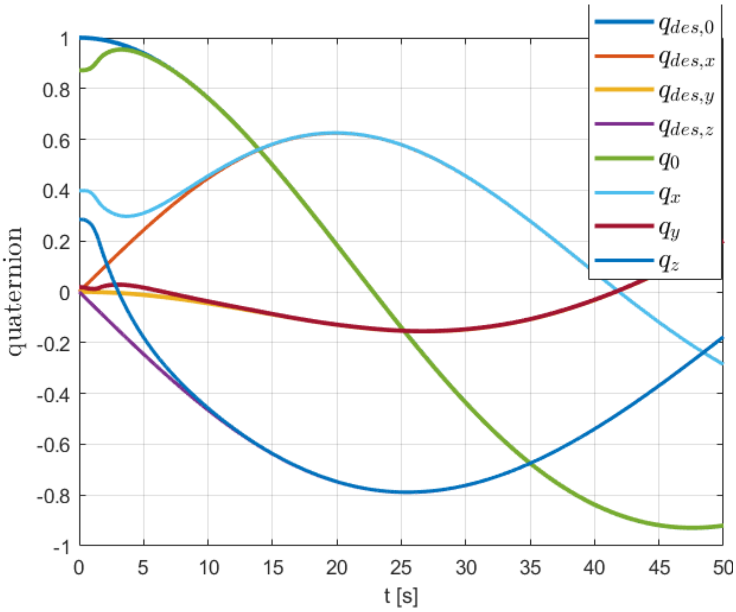

Figure 6.25. Desired vs actual quaternion components for the tracking case.

The control torque computed by the MSMC controller is compared with the torque effectively delivered by the actuator system in Figure 6.27 and 6.2.5, 6.2.5 .

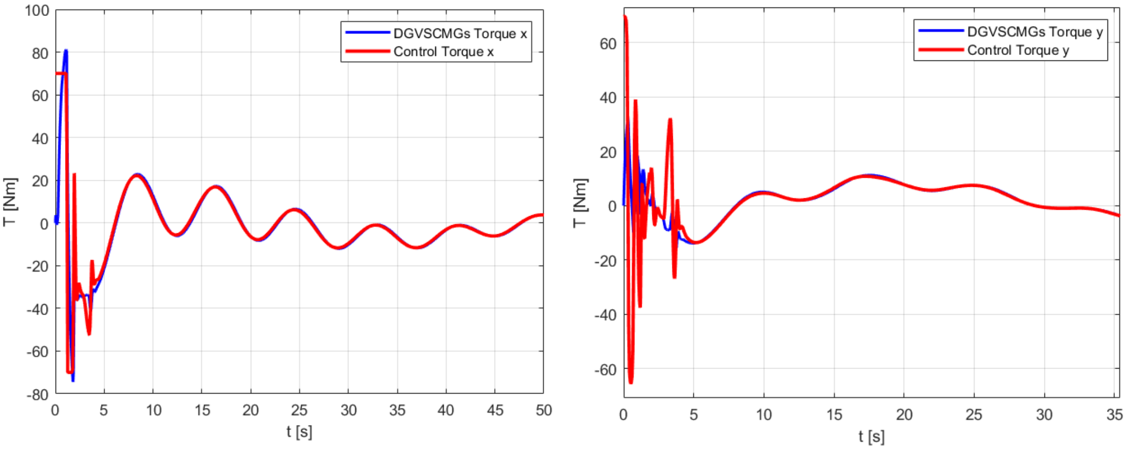

Figure 6.26. Comparison between X-component and Y-component controller torque and actuator torque of first spacecraft of the network.

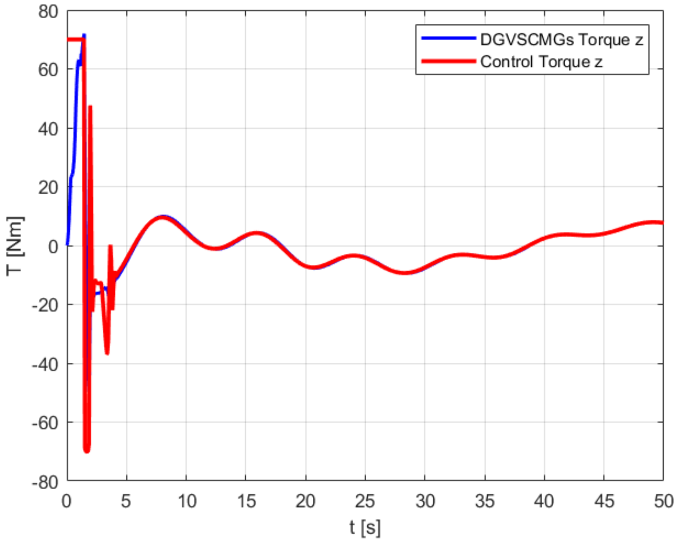

Figure 6.27. Comparison between Z-component controller torque and actuator torque of first spacecraft of the network.

According to the minimum singular value of matrix D, which evolution is shown in Figure(6.28)

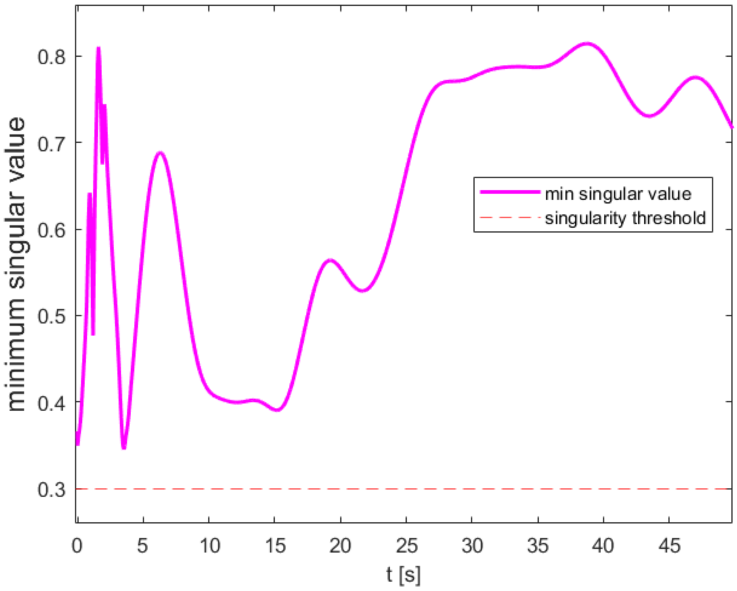

Figure 6.28. Minimum singular value evolution.

that is varying far from the singularity threshold, the torque is completely generated by the CMG-based configuration, apart from the transient phase in which the controller requires a greater amount of torque: the RW-configuration generates the torque that can not be produced by the gimbal velocities due to their saturation constraints.

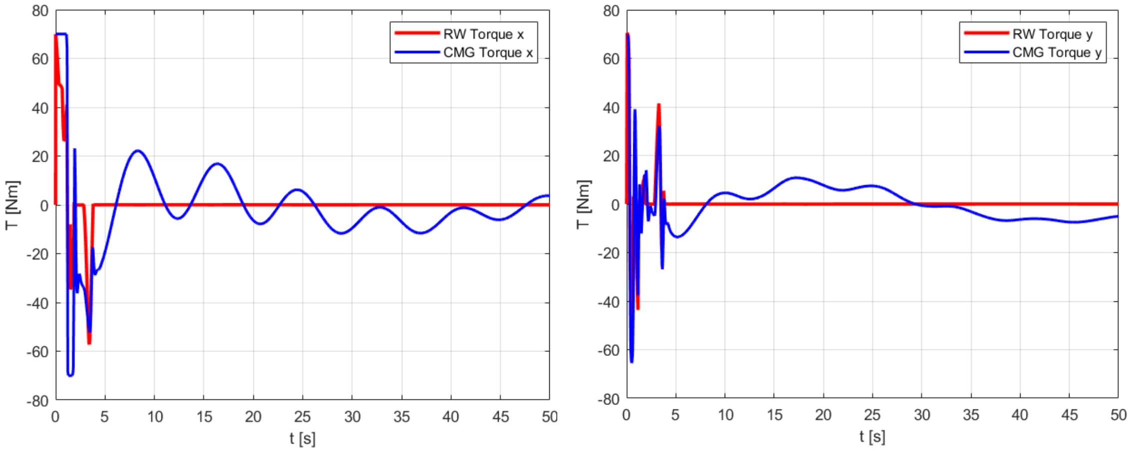

Figure 6.29. Comparison between X-component and Y-component CMG-based torque and RW-based torque of actuators of first spacecraft of the network.

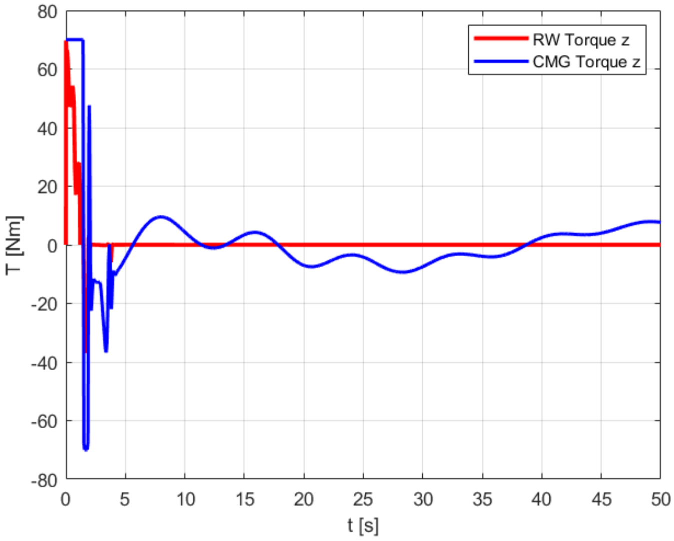

Figure 6.30. Comparison between Z-component CMG-based torque and RW-based torque of actuators of first spacecraft of the network.

Figures(6.31),(6.32),(6.34),(6.35),(6.37),(6.38) show the desired and actual gimbal angular velocities and flywheel speeds and accelerations, respectively. The desired signals are generated by the steering law, while the actual ones result from the controlled actuator dynamics.

Although the overall tracking is accurate, slight attenuation and smoothing effects are observed in correspondence with abrupt variations of the reference signals. This behavior is due to the presence of low-pass filters and actuator inertia, which have been introduced to realistically model electromechanical dynamics.

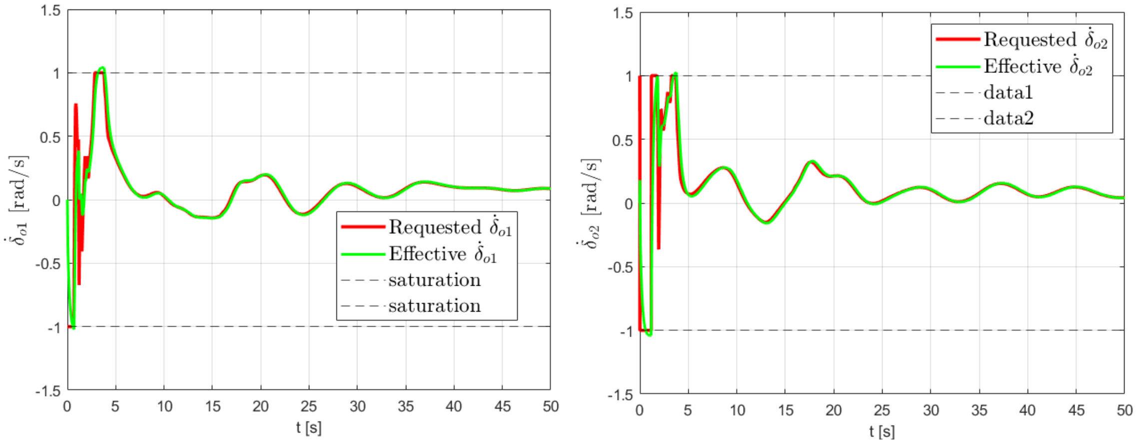

Figure 6.31.

Figure 6.32.

Figure 6.33. Desired and actual outer gimbal angular velocities.

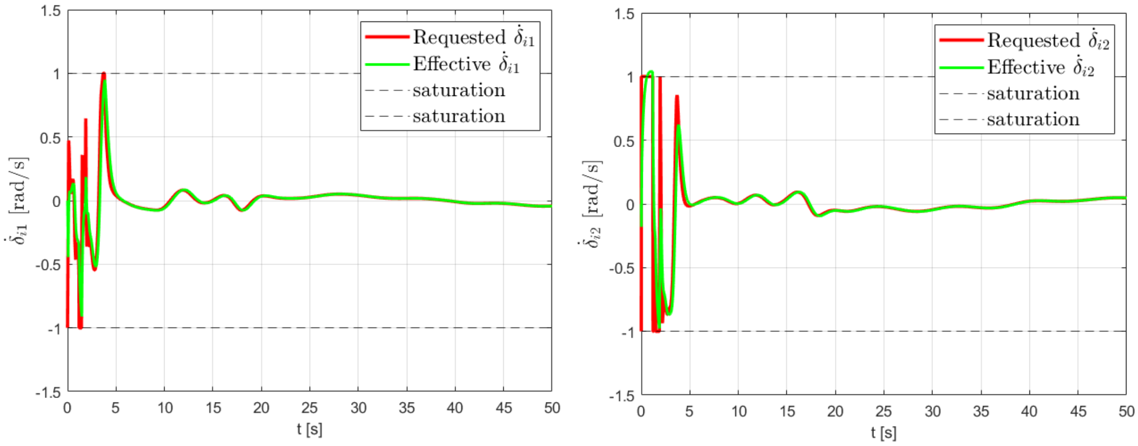

Figure 6.34.

Figure 6.35.

Figure 6.36. Desired and actual inner gimbal angular velocities.

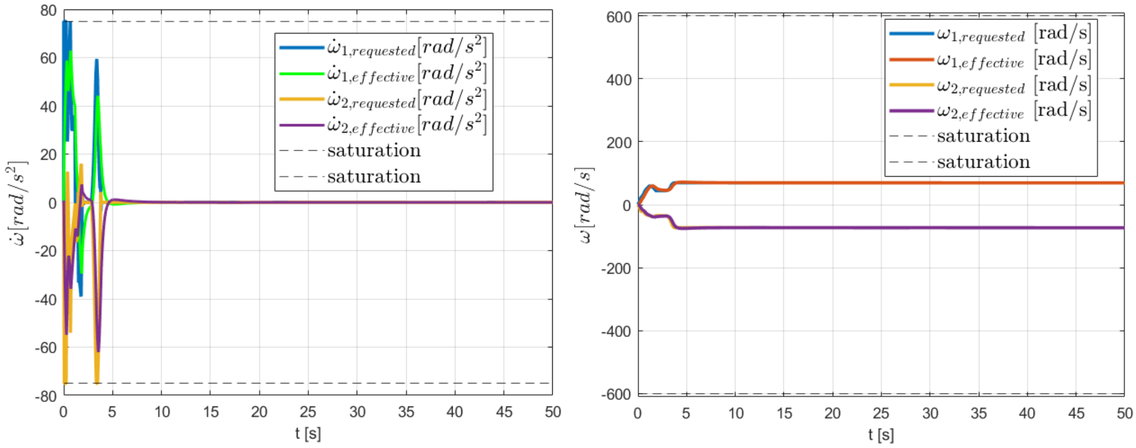

Figure 6.37.

Figure 6.38.

Figure 6.39. Desired vs actual flywheel accelerations and velocities.

The flexible dynamics response is shown in Figure(6.40).

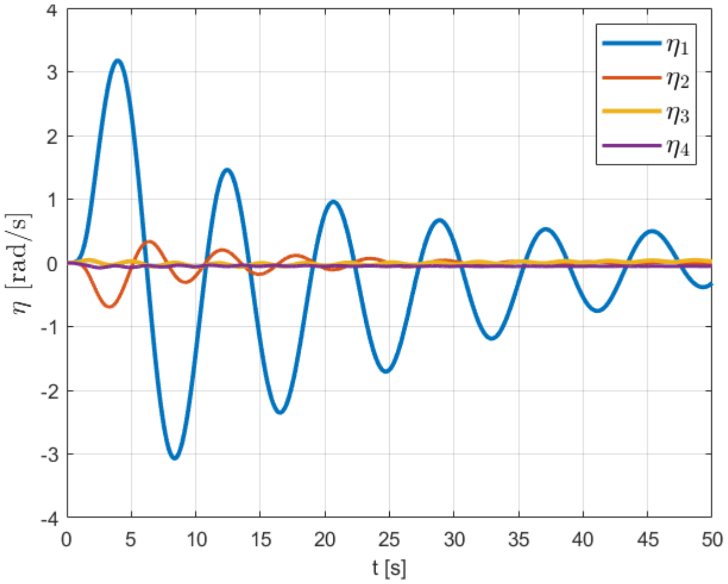

Figure 6.40. Flexible dynamics response of the first spacecraft of the network.

Pure consensus

In this simulation, a pure consensus problem is considered, obtained by setting the weighting matrix associated with the reference tracking term to zero, i.e. $\mathbf{B} = \mathbf{0}$ (all the parameters of the controller in this scenario are reported in Table (6.3)).Under this assumption the closed-loop objective is reduced to achieving state agreement among all satellites through purely distributed interactions.

Parameter	Symbol	Value	Unit of measure
control input gain	k	70	-
lambda	λ	0.8	-
chattering mitigation parameter	η	100	-
sliding weight matrix	B	diag([zeros(12,1)]	-

Table 6.3. Controller parameters for pure consensus.

In the specific case under analysis, the Laplacian matrix describes a directed communication topology with a spanning tree structure, whose root corresponds to Satellite 1. As a consequence, the final consensus value coincides with the initial condition of the root node. Therefore, Satellite 1 preserves its initial state, while all the remaining satellites asymptotically converge towards its attitude and angular velocity profiles.

The simulation results confirm the theoretical predictions. The attitude dynamics, expressed in terms of the vector part of the quaternion, show that all followers converge to the quaternion of the root satellite, despite different initial conditions. A similar behavior is observed for the angular velocity profiles, where the followers progressively synchronize their rotational rates to zero.

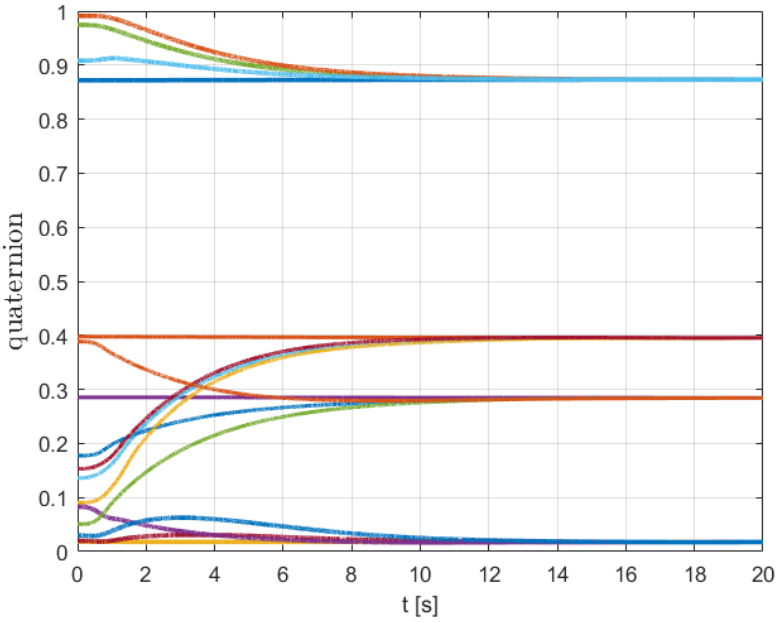

Figure 6.41. Quaternions' components of all the agents converge to consensus values.

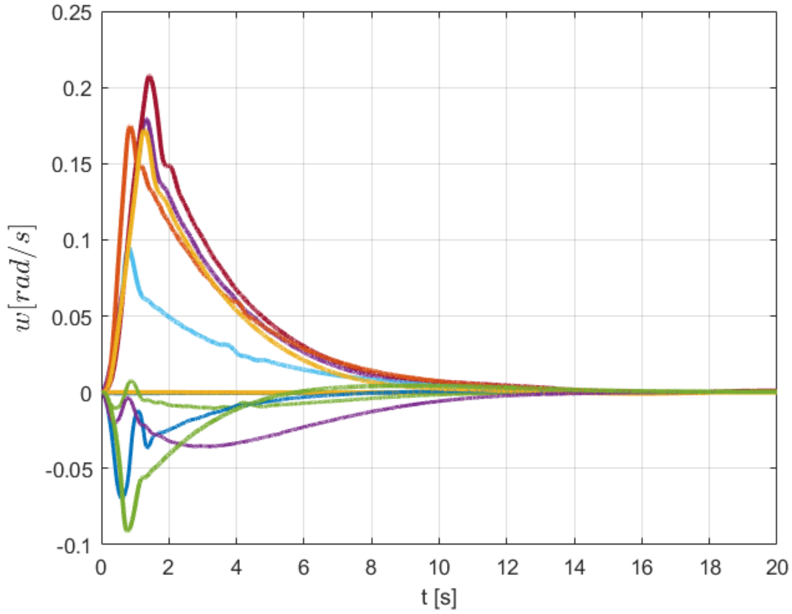

Figure 6.42. Angular velocities' components of all the agents.

It is worth remarking that the convergence of all agents to the initial condition of the root satellite, observed in the present pure consensus scenario, is strictly related to the spanning tree structure of the underlying communication graph. This property is not general.

In particular, when the network topology is not a spanning tree, the resulting consensus value is not trivially associated with the initial state of a specific agent. Moreover, due to the nonlinear nature of the proposed sliding-mode-based consensus controller, the asymptotic consensus value does not coincide with the one predicted by classical linear Laplacian-based consensus theory (function of the initial condition of all the agents).

Indeed, while in linear consensus algorithms the steady-state configuration is analytically determined by the graph eigenstructure and the initial conditions of the agents, in the present framework the equilibrium point is implicitly defined by the dynamics on the sliding manifold.

Although no closed-form expression of the consensus value is derived in this work, it can be stated that such a value is a function of the network topology and of the sliding surface definition, and in general it differs from the one obtained using linear Laplacian-based consensus controllers.

Chapter 7

Conclusions

This work presented the modeling, analysis and simulation of an Attitude and Orbit Control System (AOCS) for a multi-spacecraft network, constituted by the guidance, navigation and control (GNC) architecture, actuators and fault management systems.

The proposed distributed multi-agent control framework, that employs a Multi Sliding Mode Control (MSMC) strategy, controls the multi-hybrid dynamics system, characterized by the coupling of rigid and flexible response. The actuation system relies on Dual Gimbal Variable Speed Control Moment Gyros (DGVSCMGs). The system also incorporates a dedicated Fault Detection and Isolation (FDI) module for actuator fault management and topology reconfiguration. The overall structure proved to be suitable for handling both regulation and tracking tasks in a networked satellite scenario.

Concerning the actuator subsystem, the electromechanical dynamics were modeled using first-order approximations and controlled via three PID controllers (one for each axis of rotation) for each actuator. Simulation results demonstrated that the PID controllers provide satisfactory tracking of the commanded control inputs for the adopted model. As expected, the presence of low-pass filters representing actuator inertia introduces attenuation and phase delay during rapid variations of the control signal (e.g., during transient phases). Nevertheless, the actuators are able to reproduce the required control torques with sufficient fidelity to guarantee closed-loop stability and overall system performance.

A further relevant aspect concerns the steering law adopted for the hybrid actuator system. The steering logic dynamically switches the operating configuration of the actuators depending on the proximity to kinematic singularities. More precisely, when the system operates sufficiently far from singular conditions, the actuators mainly work in a Control Moment Gyro (CMG)-based configuration. Conversely, when the configuration approaches a kinematic singularity or enters a predefined unsafe band, the control strategy progressively shifts the actuation toward a Reaction Wheel (RW)-based configuration. This ensures continuity of torque generation and prevents loss of controllability due to singular behavior.

Simulation results clearly show that, during nominal operations, the system predominantly relies on the CMG-based configuration, while the RW-based mode is mainly used as a compensation mechanism. In particular, the RW contribution becomes significant when the torque demand exceeds what can be effectively generated by the CMG configuration alone, for example during the transient phases of the response or when steering through near-singular configurations. This hybrid steering law therefore guarantees both robustness against singularities and improved torque availability during highly demanding maneuvers, without degrading the overall closed-loop stability and performance.

The satellite model includes both rigid and flexible dynamics, with the latter described through modal coordinates. In the regulation scenario, the modal coordinates exhibit an exponentially decaying oscillatory behavior, consistent with the free response of a damped mass-spring-damper system. This behavior is mainly excited during the transient phase of the rigid body dynamics and gradually vanishes as the angular acceleration tends to zero. Such results confirm the physical consistency of the proposed flexible model.

Regarding the attitude control performance, the proposed MSMC model ensured convergence of both angular velocity and quaternion components toward their desired values, in both regulation and tracking scenarios, for all the agents of the network. The tracking simulations, in particular, confirmed the ability of the controller to follow time-varying references, with zero-steady state errors and good transient behavior. The distributed control structure, characterized by its Laplacian-based sliding surface, guarantees coherent coordination among spacecraft, achieving formation objectives while retaining robustness with respect to bounded disturbances.

An important contribution of this work is the integration of a Fault Detection and Isolation mechanism at the actuator level. Faults were simulated by injecting step-like disturbances into the actuator command signals. The resulting residuals exceeded predefined thresholds, thus correctly triggering the FDI logic. When a primary actuator failed, the reserve actuator was activated in accordance with the designed logic. In the case of a double fault involving both a primary actuator and its redundant unit, the system successfully reconfigured the communication topology by excluding the faulty spacecraft from the network, with the only constraint to respect the strong connectivity condition. Although a slight transient degradation in performance was observed during the topology switching phase, the remaining agents were still able to preserve stability and continue tracking the desired reference. This confirms the effectiveness of the proposed FDI architecture and switching topology mechanism in enhancing system robustness and fault tolerance.

Overall, the results demonstrate that the designed AOCS architecture provides satisfactory attitude control performance for a multi-satellite network, even in the presence of actuator faults and reconfiguration events. Future developments may include the integration of more realistic actuator nonlinearities, higher-order flexible dynamics of the plant, communication delays. These extensions would further strengthen the applicability of the proposed approach to real multi-spacecraft missions.

Bibliography

[1] Patrick Bergner, Andre Posch, and Patrick Bergner. Gafe methodology. *Generic AOCS/GNC Techniques Design Framework for FDIR*, 2018.

[2] JL Burch, TE Moore, RB Torbert, and BL-https Giles. Magnetospheric multiscale overview and science objectives. *Space Science Reviews*, 199(1):5–21, 2016.

[3] Ti Chen, Jinjun Shan, Hao Wen, and Shidong Xu. Review of attitude consensus of multiple spacecraft. *Astrodynamics*, 6(4):329–356, 2022.

[4] Peiling Cui and Jingxian He. Steering law for two parallel variable-speed double-gimbal control moment gyros. *Journal of Guidance, Control, and Dynamics*, 37(1):350–359, 2014.

[5] Xiaoyan He, Qingyun Wang, and Wenwu Yu. Finite-time distributed cooperative attitude tracking control for multiple rigid spacecraft. *Applied Mathematics and Computation*, 256:724–734, 2015.

[6] J-F Kaufeler, A Parkes, and A Pidgeon. The european space agency standard for space packet utilisation. In *JPL, SpaceOps 1992: Proceedings of the Second International Symposium on Ground Data Systems for Space Mission Operations*, 1993.

[7] Hans F Kennel. Steering law for parallel mounted double-gimbaled control moment gyros. revision a. Technical report, 1981.

[8] Vaios J Lappas. *A control moment gyro (CMG) based attitude control system (ACS) for agile small satellites.* University of Surrey (United Kingdom), 2002.

[9] Daren Lee, Krishna Dev Kumar, and Manoranjan Sinha. Fault detection and recovery of spacecraft formation flying using nonlinear observer and reconfigurable controller. *Acta Astronautica*, 97:58–72, 2014.

[10] Peter W Likins. Dynamics and control of flexible space vehicles. Technical report, 1970.

[11] Mauro Mancini. Adaptive variable structure control system for attitude spacecraft applications. 2023.

[12] S Pallaschke, I Muñoz, J Rodriquez-Canabal, D Sieg, and JJ Yde. Cluster: Mission overview and end-of-life analysis. In *Proceedings of the 20th International Symposium on Space Flight Dynamics*, 2007.

[13] Garima Pandey, Giovanni Lapicirella, Marco Lovera, Davide Invernizzi, Salvatore Meraglia, Carole Rosso, Ilario Cantiello, et al. Investigating advanced aocs techniques to improve robustness and reliability of low-cost missions. In *12th International Conference on Guidance, Navigation & Control Systems (GNC) and 9th International Conference on Astrodynamics Tools and Techniques (ICATT)*, pages 1–15, 2023.

[14] Jonas Radtke, Christopher Kebschull, and Enrico Stoll. Interactions of the space debris environment with mega constellationsâusing the example of the oneweb constellation. *Acta Astronautica*, 131:55–68, 2017.

[15] Chang-E Ren and CL Philip Chen. Sliding mode leader-following consensus controllers for second-order non-linear multi-agent systems. *IET Control Theory & Applications*, 9(10):1544–1552, 2015.

[16] Wei Ren and Randal W Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on automatic control*, 50(5):655–661, 2005.

[17] Daniele Russo, Fausto Francesco Lizzio, Elisa Capello, and Yasumasa Fujisaki. Minimal-order distributed observer for a network of heterogeneous satellites with flexible appendages. In *2025 European Control Conference (ECC)*, pages 286–291. IEEE, 2025.

[18] Takahiro Sasaki, Takashi Shimomura, and Hanspeter Schaub. Robust attitude control using a double-gimbal variable-speed control moment gyroscope. *Journal of Spacecraft and Rockets*, 55(5):1235–1247, 2018.

[19] Massimo Tipaldi and Bernhard Bruenjes. Survey on fault detection, isolation, and recovery strategies in the space domain. *Journal of Aerospace Information Systems*, 12(2):235–256, 2015.

[20] Vadim Utkin, Alex Poznyak, Yury Orlov, and Andrey Polyakov. Conventional and high order sliding mode control. *Journal of the Franklin Institute*, 357(15):10244–10261, 2020.

[21] Qishao Wang, Zhisheng Duan, and Yuezu Lv. Distributed attitude synchronization control for multiple flexible spacecraft without modal variable measurement. *International Journal of Robust and Nonlinear Control*, 28(10):3435–3453, 2018.

[22] Baolin Wu, Danwei Wang, and Eng Kee Poh. Decentralized sliding-mode control for attitude synchronization in spacecraft formation. *International Journal of Robust and Nonlinear Control*, 23(11):1183–1197, 2013.

[23] Yujia Wu. *Distributed Sliding Mode Control for Nonlinear Consensus.* University of California, Berkeley, 2019.

[24] Deyin Yao, Hongyi Li, Renquan Lu, and Yang Shi. Distributed sliding-mode tracking control of second-order nonlinear multiagent systems: An event-triggered approach. *IEEE Transactions on Cybernetics*, 50(9):3892–3902, 2020.

[25] Chengfei Yue, Qiang Shen, Xibin Cao, Feng Wang, Cher Hiang Goh, and Tong Heng Lee. Development of a general momentum exchange devices fault model for spacecraft fault-tolerant control system design. *arXiv preprint arXiv:1907.06751*, 2019.

[26] Kuan Zhang, Shuchen Wang, Saijin Wang, and Qizhi Xu. Anomaly detection of control moment gyroscope based on working condition classification and transfer learning. *Applied Sciences*, 13(7):4259, 2023.

[27] Hui Zhao, Feng Liu, and Yu Yao. Optimization design steering law for vscmgs with the function of attitude control and energy storage. *Aerospace Science and Technology*, 65:9–17, 2017.

[28] Qun Zong and Shikai Shao. Decentralized finite-time attitude synchronization for

multiple rigid spacecraft via a novel disturbance observer. *Isa Transactions*, 65:150–163, 2016.