



**TÉCNICO**  
LISBOA

# **A Convex Allocation Framework for Singularity Avoidance in Control Moment Gyro Clusters**

**Hugo André Chamusca Pereira**

Thesis to obtain the Master of Science Degree in

## **Aerospace Engineering**

Supervisors: Prof. Pedro Tiago Martins Batista  
Dr. Pedro António Duarte Marques Lourenço

### **Examination Committee**

Chairperson: Prof. Paulo Jorge Coelho Ramalho Oliveira  
Supervisor: Dr. Pedro António Duarte Marques Lourenço  
Member of the Committee: Dr. Sérgio Daniel Gonçalves Gante Brás

**November 2022**



This thesis is dedicated to my loved ones.



## Acknowledgments

First of all, I would like to thank my supervisors, professor Pedro Batista and Doctor Pedro Lourenço, for giving me the opportunity to work on a topic that I am passionate about. Their support, guidance, and availability were paramount for the progress of this thesis. Moreover, their knowledge and theoretical insights helped me to carry out novel research on such a challenging topic.

Then, I would like to express my gratitude towards AeroTéc, the DanSTAR group, and the FST Lisboa team, where I had the opportunity to work on many challenging engineering projects. Thanks for contributing to my knowledge and for allowing me to work with the smartest and most dedicated people I could have ever imagined. Lastly, thanks for providing me with the opportunity to compete in some of the best student engineering competitions in the world.

I also thank Instituto Superior Técnico for giving me the best group of friends I could have asked for. They convinced me that success cannot be achieved if we are not surrounded by the right people. I will not forget the endless nights that we shared studying and working on our academic assignments. Thank you Henrique, Francisco, Daniel, Luís, João, and José.

Last but not least, I would like to thank my parents, sister, and grandparents for helping me fulfil my dream of becoming an aerospace engineer. They always backed me in the most stressful moments and were there to celebrate my achievements. At the end of the day, nothing of this would have been possible without their unconditional support. To my closest friends João, Rúben, Filipe, and Francisco, a huge thanks for providing me with the best social life I could have had while studying to become an engineer. Finally, a special thanks to my girlfriend Beatriz for her everlasting love during my past five years at Instituto Superior Técnico. You bring out the best in me.



## Resumo

Em robótica, a ocorrência de singularidades está associada à perda de um grau de liberdade. No contexto de satélites que utilizam sistemas de giroscópios de controlo de momento como atuadores de direção, as singularidades podem surgir devido a alinhamentos dos rotores. Isto resulta na perda da autoridade de controlo sobre o sistema, que tem consequências diretas no controlo da orientação do satélite. Nesta tese propõe-se uma nova abordagem para evitar as singularidades, que consiste numa estratégia de alocação complementar a um sistema de controlo convencional. Esta abordagem permite antever as posições futuras dos rotores, com base num modelo matemático do sistema, e com essa informação determinar as ações de controlo que conduzem a configurações não singulares do sistema. A utilização de um atuador redundante permite evitar as singularidades enquanto o sistema obedece às referências de torque impostas pelo controlador. Para além disso, é derivada uma nova métrica, computacionalmente eficiente e numericamente robusta, para avaliar a proximidade das singularidades ao sistema. Esta solução é relativamente simples em comparação com a maioria das soluções existentes, como é o caso do número de condição, podendo ser integrada como uma restrição linear num problema de otimização convexo. Por fim, o potencial da abordagem proposta é demonstrado em ambiente de simulação através do uso de um modelo bidimensional de giroscópios de controlo de momento. Verifica-se que é possível evitar todas as singularidades internas do sistema de uma forma eficiente do ponto de vista computacional.

**Palavras-chave:** giroscópios de controlo de momento; singularidades; governador de ações de controlo; controlo por modelo preditivo; convexidade.



## Abstract

In robotics, the occurrence of singularities typically results in a loss of a degree of freedom. For spacecrafts employing control moment gyro clusters, singularities may occur due to the alignment of the gimbals, which inhibits the creation of torque in at least one direction. This translates into a loss of control authority that has a direct impact on the spacecraft's attitude control system. In this work, an optimal allocation framework for singularity avoidance is proposed. The presented solution aims to provide a singularity-robust allocation scheme that can be used as an add-on to a conventional attitude controller. This algorithm resorts to the model predictive control framework to predict the future states of the gimbals and, subsequently, take control actions that lead to singularity-free configurations while minimizing the control energy spent. The use of a redundant actuator makes it possible to avoid singularities while the system meets the torque references defined by the controller. Moreover, a novel, computationally efficient and numerically robust, singularity metric is derived to assess the proximity of a singularity. This function overcomes the complexity of the standard literature solutions, such as the condition number, and can be integrated as a linear constraint in a convex optimization problem. Finally, the proposed approach is applied to a two-dimensional control moment gyro cluster in a simulation environment. It is verified that the system is capable of avoiding all of the internal singularities of the cluster at a relatively low computational expense.

**Keywords:** control moment gyros; singularities; action governor; model predictive control; convexity.



# Contents

Acknowledgments . . . . .	v
Resumo . . . . .	vii
Abstract . . . . .	ix
List of Figures . . . . .	xiii
List of Tables . . . . .	xv
Acronyms . . . . .	xvii
Notation and Nomenclature . . . . .	xix
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.1.1 Spacecraft Control Overview . . . . .	1
1.1.2 Reaction Wheels vs. Control Moment Gyros . . . . .	2
1.1.3 Types of Control Moment Gyros . . . . .	4
1.1.4 Singularities . . . . .	4
1.1.5 Applications and Future of CMGs . . . . .	5
1.2 Objectives . . . . .	5
1.3 Literature Review . . . . .	6
1.3.1 Singularity Escape Algorithms . . . . .	6
1.3.2 Singularity Avoidance Algorithms . . . . .	8
1.3.3 Singularity Avoidance and Escape Algorithms . . . . .	9
1.4 Contributions . . . . .	11
1.5 Thesis Outline . . . . .	12
<b>2 Background</b>	<b>13</b>
2.1 Spacecraft Actuators . . . . .	13
2.1.1 Magnetic Torque Rods . . . . .	13
2.1.2 Thrusters . . . . .	13
2.1.3 Reaction and Momentum Wheels . . . . .	14
2.1.4 Control Moment Gyros . . . . .	14
2.2 Attitude Representations . . . . .	14
2.3 Singular Value Decomposition . . . . .	15

2.4	Underdetermined Systems . . . . .	17
2.5	Optimization Problems and Convexity . . . . .	18
2.6	Action Governor . . . . .	19
2.7	Model Predictive Control . . . . .	20
<b>3</b>	<b>Model Definition</b>	<b>23</b>
3.1	Actuator Dynamics . . . . .	23
3.2	Satellite Dynamics . . . . .	26
3.3	Satellite Kinematics . . . . .	27
3.4	State-Space Representation . . . . .	29
3.5	Model Discretization . . . . .	29
<b>4</b>	<b>Singularity Analysis</b>	<b>31</b>
4.1	Analysis of the 2D System . . . . .	31
4.2	Analysis of the 3D System . . . . .	36
4.3	Singularity Classification . . . . .	38
4.4	Vicinity of a Singularity . . . . .	44
<b>5</b>	<b>Controller Design and Control Allocation</b>	<b>47</b>
5.1	Singularity Metrics . . . . .	47
5.2	Action Governor . . . . .	50
5.3	Integrated Allocation . . . . .	54
5.4	MPC-based Allocation . . . . .	56
<b>6</b>	<b>Simulation Results</b>	<b>59</b>
6.1	Action Governor Results . . . . .	60
6.2	MPC-based Allocation Results . . . . .	65
6.2.1	Main Results . . . . .	65
6.2.2	Execution Times . . . . .	70
6.2.3	Linearization Analysis . . . . .	71
<b>7</b>	<b>Conclusions and Future Work</b>	<b>73</b>
7.1	Conclusions . . . . .	73
7.2	Future Work . . . . .	74
	<b>Bibliography</b>	<b>75</b>
<b>A</b>	<b>CMGs and RWs Specifications</b>	<b>79</b>

# List of Figures

1.1	Attitude control diagram overview. . . . .	2
1.2	RWs and CMGs working principle. . . . .	3
1.3	RWs and CMGs mass and torque comparison. . . . .	3
1.4	Types of CMG allocation algorithms. . . . .	6
2.1	Geometric interpretation of SVD. . . . .	16
2.2	Example of a convex function. . . . .	19
2.3	Example of a non-convex function. . . . .	19
2.4	Action governor scheme. . . . .	20
2.5	MPC structure overview. . . . .	20
2.6	MPC-based control diagram. . . . .	21
2.7	MPC operating principle. . . . .	22
3.1	Satellite model breakdown. . . . .	23
3.2	Triangular array of CMGs. . . . .	24
3.3	Roof array of CMGs. . . . .	24
3.4	Angular momentum envelopes. . . . .	25
3.5	Actuator modelling. . . . .	25
4.1	Types of singularities in the triangular array of CMGs. . . . .	32
4.2	Examples of singularities in the triangular array of CMGs. . . . .	32
4.3	Jacobian's principal components for the different gimbal configurations in the triangular array of CMGs. . . . .	33
4.4	Torque envelope for the different gimbal configurations in the triangular array of CMGs. . . . .	34
4.5	Singularities envelope for the triangular array of CMGs. . . . .	35
4.6	Examples of rank-1 singularities in the roof array of CMGs. . . . .	37
4.7	Examples of rank-2 singularities in the roof array of CMGs. . . . .	37
4.8	Principal components and torque envelope for a non-singular gimbal configuration in the roof array of CMGs. . . . .	38
4.9	Principal components and torque envelope for a rank-2 singularity in the roof array of CMGs. . . . .	39
4.10	Principal components and torque envelope for a rank-1 singularity in the roof array of CMGs. . . . .	39
4.11	Classification of SGCMG singularities. . . . .	40

4.12 Vicinity of a singularity - a practical example for the triangular system. . . . .	45
5.1 Exclusion requirement for the 2D case. . . . .	49
5.2 AG-based control diagram for singularity avoidance. . . . .	50
5.3 MPC-based control diagram for singularity avoidance. . . . .	57
6.1 Satellite model. . . . .	60
6.2 Action governor - torque error for different threshold values. . . . .	61
6.3 Action governor - gimbal rates for different threshold values. . . . .	62
6.4 Distance to the closest singularity: with AG vs. without AG. . . . .	62
6.5 Manipulability index: with AG vs. without AG. . . . .	63
6.6 Condition number: with AG vs. without AG. . . . .	63
6.7 Nullspace scaling component. . . . .	64
6.8 Reaction wheel speed. . . . .	64
6.9 Gimbal trajectory without AG. . . . .	64
6.10 Gimbal trajectory with AG. . . . .	65
6.11 MPC-based allocation: control torques vs. actual torques. . . . .	66
6.12 Reference tracking results using the MPC-based allocation algorithm. . . . .	66
6.13 Gimbal rates using the MPC-based allocation algorithm. . . . .	67
6.14 Distance to the closest singularity using the MPC-based allocation algorithm. . . . .	68
6.15 Gimbal trajectory using the MPC-based allocation algorithm. . . . .	69
6.16 Torque error vs. horizon. . . . .	69
6.17 Control energy magnitude vs. horizon. . . . .	70
6.18 Control energy variation vs. horizon. . . . .	70
6.19 Execution times for the different formulations. . . . .	71

# List of Tables

1.1	Relative differences between a RW and a CMG. . . . .	4
2.1	Comparison of attitude representation methods. . . . .	15
2.2	Advantages and disadvantages of MPC over conventional controllers. . . . .	22
4.1	Description of the different singularity classes. . . . .	40
6.1	Model parameters used in simulation. . . . .	59
6.2	Linearization: worst-case error analysis. . . . .	72
6.3	Linearization: average error analysis. . . . .	72
A.1	Control moment gyros specifications. . . . .	79
A.2	Reaction wheels specifications. . . . .	80



# Acronyms

**ACS** attitude control system

**AG** action governor

**AOCS** attitude and orbit control system

**CMG** control moment gyro

**DGCMG** double gimbal control moment gyro

**KKT** Karush–Kuhn–Tucker

**ML** machine learning

**MPC** model predictive control

**NLP** nonlinear programming

**RW** reaction wheel

**SGCMG** single gimbal control moment gyro

**SVD** singular value decomposition

**VSCMG** variable speed control moment gyro



# Notation and Nomenclature

Throughout this thesis, math symbols and letters are represented in italic. Scalar variables and constants are represented in either small or capital letters, vectors are represented in small boldface letters, and matrices are represented in capital boldface letters. In particular,  $\mathbf{I}_n$  represents the  $n \times n$  identity matrix, and  $\text{diag}(a_1, \dots, a_N)$  represents the diagonal matrix whose diagonal entries are given by  $a_1, \dots, a_N$ . The  $i$ -th element of the vector  $v$  is denoted by  $v_i$ , and the entry  $(i, j)$  of a matrix  $M$  is denoted by  $M_{ij}$ .

For a square symmetric matrix  $P$ ,  $P \succ 0$  and  $P \succeq 0$  are used to indicate that  $P$  is positive-definite and positive-semidefinite, respectively. The operator  $\|\cdot\|_2$  is used to indicate the 2-norm of a vector, the operator  $\|\cdot\|_\infty$  is used to indicate the  $\infty$ -norm of a vector, and the operator  $\|\cdot\|_S$  is equivalent to  $\|\cdot\|_S = \sqrt{(\cdot)^T S (\cdot)}$ , where  $S$  is a positive-definite matrix.

The time derivative of any quantity  $a$  is denoted by  $\dot{a} := \frac{da}{dt}$ . These two possibilities are used interchangeably depending on which is best for legibility.

Given a quaternion  $\bar{q}$ , its real part is denoted by  $q_0$ , and its vectorial part is denoted by  $\mathbf{q}$ . The Hamilton product operator between two quaternions is given by  $\otimes$ .

Finally, the nomenclature used to denote the physical quantities used in this thesis is provided below.

---

## Actuator variables and parameters

---

$\gamma$	gimbal angles	$\mathbf{h}$	angular momentum
$\dot{\gamma}$	gimbal rates	$h_0$	momentum magnitude
$\ddot{\gamma}$	gimbal accelerations	$\dot{\mathbf{h}}$	torque
$\ \dot{\gamma}\ _\infty$	largest gimbal rate	$\mathbf{J}$	Jacobian
$\dot{\gamma}_{\max}$	maximum gimbal rate	$N$	number of CMGs
$\dot{\gamma}_0$	energy-optimal gimbal rates	$\mathbf{N}$	normalized nullspace matrix

---

**Satellite variables and parameters**

---

$\bar{q}$	attitude quaternion	$\omega_{\max}$	maximum slew rate
$\bar{r}$	reference quaternion	$h_s$	spacecraft angular momentum
$\omega$	angular velocity	$I_s$	spacecraft inertia
$\dot{\omega}$	angular acceleration	$\tau$	external torques

---

**Control variables and parameters**

---

$\dot{h}_c$	command torque	$\bar{q}_e$	error quaternion
$m$	manipulability index	$P_q$	error quaternion gain
$c$	condition number	$P_\omega$	angular velocity gain
$d$	distance to the closest singularity	$H$	torque error optimization weight
$\rho$	singularity term weight	$U$	gimbal rates optimization weight
$e$	torque error	$M$	gimbal accelerations optimization weight

---

**Singularity variables and parameters**

---

$u$	singular direction	$\gamma_s$	singular gimbal configuration
$\sigma$	singular values	$\gamma_p$	singularity projection
$\Sigma$	Jacobian singular values matrix	$A_s$	singularity line projection matrix
$U$	left singular vectors matrix	$b_s$	singularity line offset vector
$V$	right singular vectors matrix	$\kappa$	distance threshold

---

**Other symbols**

---

$\dot{h}_\omega$	reaction wheel torque	$\mathcal{I}$	inertial axes
$J_\omega$	reaction wheel inertia	$\mathcal{B}$	body axes
$\omega_\omega$	reaction wheel velocity	$T$	simulation time
$\dot{\omega}_\omega$	reaction wheel acceleration	$T_s$	sampling time

# Chapter 1

## Introduction

### 1.1 Motivation

This thesis tackles the singularity problem that occurs in spacecrafts using control moment gyros as steering actuators. Singularities are critical for missions requiring high-pointing capabilities, as they compromise the effectiveness of a spacecraft's attitude controller. In this section, the state-of-the-art framework for spacecraft control is revised. Herein, the hardware used for controlling a spacecraft, including sensors and actuators, is presented. In addition, two of the most popular spacecraft attitude actuators, i.e. reaction wheels and control moment gyros, are compared. It becomes evident that the use of control moment gyros is advantageous for larger spacecrafts. The different variants of control moment gyros are introduced in the following section. Later on, the singularity problem, as it is widely known in robotics, is briefly discussed. Finally, some historical applications of control moment gyros are addressed as well as the future prospects of their use.

#### 1.1.1 Spacecraft Control Overview

Since the early days of space exploration that engineers had to find a way of steering a spacecraft in its orbit. Even though some of the first spacecrafts launched into space did not employ any kind of on-board control system, such as the very own Sputnik 1, it did not take long before the space community realized the importance of stabilizing a satellite as well as controlling its orientation. Indeed, a spacecraft that is not capable of steering itself to a desired orientation is not generally suited for space missions. A satellite powered by solar arrays has to point to the Sun during most of its orbit, Earth-observation satellites must precisely track certain regions of the surface of the Earth, deep-space communications are only possible due to rigorous on-board antenna pointing mechanisms, and a space rendezvous demands slow and controlled movements between two spacecrafts until contact. Therefore, spacecraft control is paramount for the success of a space mission and is a widely studied field since the beginning of space exploration.

Within spacecraft control, one can disjoin orbit control from attitude control. Orbit control relies on the use of a propulsion system capable of provoking velocity increments for orbit corrections, orbit transfers

or even orbit injections. On the other hand, attitude control refers to angular motion control. It can be interpreted as the process of steering a spacecraft from a given orientation to a different orientation under performance requirements. To achieve this, satellites equipped with attitude and orbit control systems (AOCS) encompass attitude and orbit sensors, estimation methods, control algorithms, and physical actuators to realize the desired control commands. In short, a conventional attitude control system (ACS) for a satellite can be explained by the diagram depicted in Fig. 1.1.

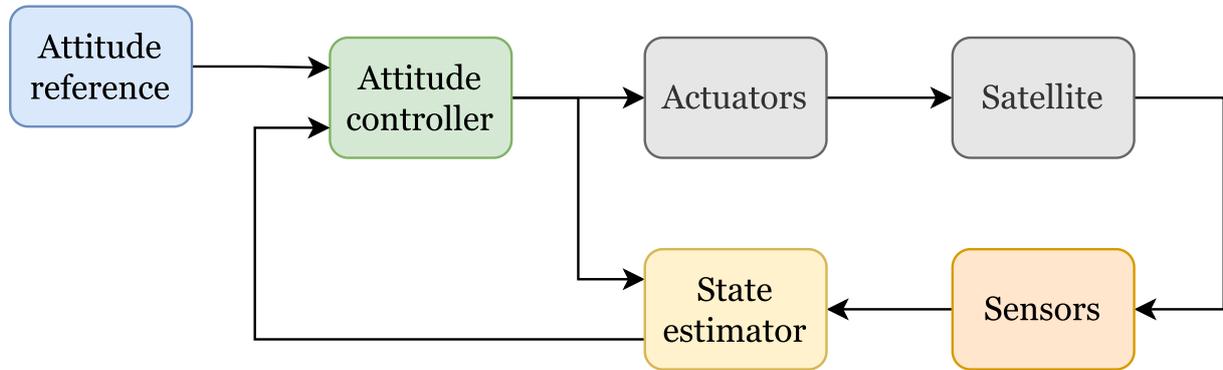


Fig. 1.1: Attitude control diagram overview.

To determine the orientation of a satellite, many different sensors can be employed. Sun sensors can easily determine the attitude of a spacecraft relative to the Sun, however, these devices require an active line-of-sight to the Sun. Magnetometers can be used to determine the heading of a spacecraft, by comparing the sensed magnetic field strength and direction to the Earth's magnetic field. Earth and horizon sensors provide Earth-relative information for Earth-pointing spacecrafts. Finally, star trackers estimate the orientation of a spacecraft by comparing to the position of known stars around it. These attitude measurements can be further combined with a state observer and the refined attitude estimation is given by a Kalman filter, for example. To control the spacecraft to a desired orientation, spacecraft actuators are used. Since we are referring to rotational motion, these devices must be able to produce control torques. The most frequently used actuators are those that rely on the application of forces on the spacecraft, and those that exchange angular momentum with the spacecraft. The first includes magnetic torque rods and thrusters, and the latter includes reaction/momentum wheels and control moment gyros.

### 1.1.2 Reaction Wheels vs. Control Moment Gyros

Both reaction wheels (RWs) and control moment gyros (CMGs) exploit the principle of conservation of angular momentum to steer a spacecraft. While CMGs create reaction torque by gimbaling a rotating mass, reaction wheels produce torque by varying the speed of the rotating mass. A common issue with reaction wheels is that they do not work well with large spacecrafts. In fact, when the spacecraft mass increases, the inertia of the reaction wheel has to increase proportionally, in order to be able to produce the same angular motion. In the case of massive objects, the gyroscopic effects from the control moment gyros are usually preferred. Indeed, CMGs allow to produce higher amounts of torque for much less power, making them far more power efficient. It should be noted, however, that CMGs are traditionally heavier and more mechanically complex than RWs, which confined their use to larger spacecrafts.

Recently, there have been efforts to miniaturize them for agile small-to-nano satellite platforms [1][2]. Fig. 1.2 illustrates the principle of operation of both reaction wheels and control moment gyros. Here,  $h$  is the angular momentum vector of the flywheel,  $\dot{h}$  denotes the torque, and  $\dot{\gamma}$  represents the direction of the gimbal axis. It can be observed that while the momentum and torque vectors are aligned with the spinning axis for RWs, these vectors are orthogonal for CMGs.

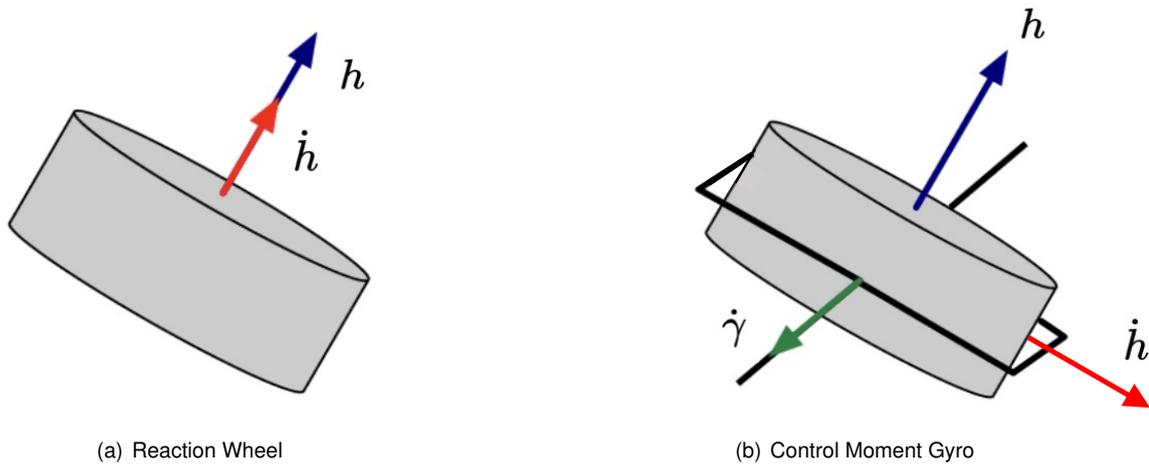


Fig. 1.2: Actuators working principle (Adapted from [3]).

The plot in Fig. 1.3 compares masses and torques of many different RWs and CMGs, whose specifications are available online (see Appendix A). CMGs are predominantly heavier and likely to create higher amounts of torque, whereas RWs are generally lighter and produce limited amounts of torque. Besides this, the torque/mass ratio is higher for CMGs, confirming their superior power efficiency. Ta-

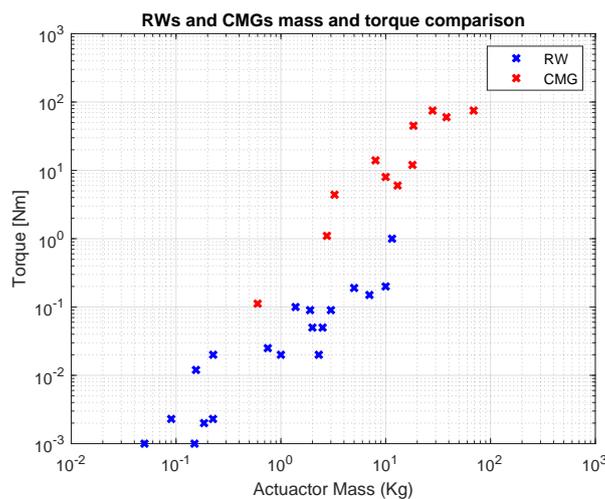


Fig. 1.3: RWs and CMGs mass and torque comparison.

ble 1.1 summarizes the main differences between a reaction wheel and a control moment gyro.

Table 1.1: Relative differences between a RW and a CMG.

	<b>Torque</b>	<b>Mass</b>	<b>Power efficiency</b>	<b>Mechanical complexity</b>	<b>Reliability</b>	<b>Cost</b>
<b>RW</b>	Lower	Lower	Lower	Lower	Higher	Lower
<b>CMG</b>	Higher	Higher	Higher	Higher	Lower	Higher

### 1.1.3 Types of Control Moment Gyros

There are three common types of control moment gyros.

- The first type is the single gimbal CMG (SGCMG). This device uses a single gimbal motor that produces torque on the satellite by altering the direction of the rotor's angular momentum. In a SGCMG, the spinning rotor is constrained to rotate on a circle located in a plane normal to the gimbal axis.
- The second type is the double gimbal CMG (DGCMG). This CMG comprises two gimbal motors, allowing elevation and azimuth control of the rotor's angular momentum vector. Indeed, DGCMGs are more versatile than SGCMGs as the former can point the rotor's angular momentum in any direction. Yet, SGCMGs offer significant cost, power, weight and reliability advantages over DGCMGs. Nonetheless, as each gimbal offers an extra degree of freedom, the gimbal steering problem is much simpler for DGCMGs.
- Most CMGs hold rotor speed constant, for simplicity, by using small and low power motors. Nonetheless, novel research has focused on the possibility of changing the rotor speed while the CMG gimbals, introducing variable speed CMGs (VSCMGs). The advantages brought by changing the rotor speed control are very few, when considering actuation capability, because the torque produced by the rotor is almost negligible when compared to that caused by gimbal motion. However, these devices offer an additional degree of freedom that can be explored for VSCMG cluster reorientation and singularity avoidance [4].

### 1.1.4 Singularities

A spacecraft using SGCMGs as steering actuators requires at least three of these devices for attitude control. Indeed, there are as many degrees of freedom as gimbals. While gimbaling a flywheel, there is the chance of witnessing alignments in the system. When these situations happen, the spacecraft is not able to produce net torque along certain directions. These configurations are known as singularities and are related to the kinematics of a robotic arm that encounters motion limits on the end-effector due to joint alignments [5]. Also denoted by gimbal-lock, these special configurations happen when the spacecraft is holding the maximum amount of angular momentum in one direction - in this case referred to as saturation singularities, or when the spacecraft loses controllability due to an anti-parallel alignment in the system. In the first scenario, the ACS computes a command torque that exceeds the physical limitations of the actuators, causing all the CMGs to stay aligned exerting angular momentum in the same direction. In the second case, a geometrical alignment as a consequence of gimbal motion

causes one CMG to be spinning in the opposite direction of a different one, thus not allowing the creation of torque along the flywheel's spinning direction.

### **1.1.5 Applications and Future of CMGs**

Control moment gyros have been widely used for spacecraft attitude control since the early 70s, when Skylab was launched. This spacecraft encompassed three DGCMGs arranged in a way such that the rotors, when all the gimbals were at their zero positions, pointed in three mutually perpendicular directions. From this date onwards, CMGs have been used in numerous space missions where large spacecrafts are used and high pointing accuracy is required. The International Space Station is another spacecraft that employs CMGs for orientation control. This spacecraft is controlled by a CMG system composed by four parallel mounted DGCMGs. Two of these CMGs are mounted antiparallel with the other two. The utmost goal of this system is to absorb the momentum created by external disturbances acting on the space station. The soviet, and later russian, space station Mir also employed a system of control moment gyros [4].

CMGs have revolutionized spacecraft control since their appearance. With these devices, larger spacecrafts could achieve reasonable amounts of torque at a considerably low energy expense. Future and even contemporary space missions, such as Earth-observation missions, demand high agility requirements from the satellites. The current reaction wheel models are not able to cope with such requirements and the existing CMGs are incompatible from a mass-wise perspective. For this reason, new CMG models are being developed with the scope of minimizing their mass while guaranteeing the torques needed for agile missions. This research is paving the way for agile missions that, until then, were not possible due to the physical limitations of the actuators.

## **1.2 Objectives**

Given the latest achievements in the field of control moment gyros with the scope of optimizing their mass/size while ensuring high agility, it is foreseen that the use of CMGs in both large and small satellites will be a big trend in the coming years. At the same time, research is being conducted with the goal of avoiding the singular configurations provoked by gimbal motion. Indeed, singularities constitute an issue of major concern as they can affect the spacecraft's orientation during a manoeuvre, and therefore be critical for missions requiring high-pointing capabilities. The major goal of this thesis is to provide an attitude control framework for singularity avoidance. This can be accomplished by first understanding the different types of singularities of a specific CMG array, concluding about their influence on the spacecraft's motion, and finding a way of minimizing their effect.

To validate and showcase the potential of the solutions proposed in this work, a mathematical model of a satellite is ought to be implemented in a known environment, where many control strategies can be tested.

## 1.3 Literature Review

Modern spacecrafts equipped with CMGs require singularity avoidance algorithms for precise tracking of targets, proper payload functioning, and various types of attitude manoeuvres while in orbit. Thus, control algorithms for CMG allocation are an important piece of a spacecraft's AOCS system. The following paragraphs summarize the state-of-the-art of singularity avoidance for spacecrafts using control moment gyroscopes.

Several allocation algorithms have been studied to determine the inputs to the gimbals, i.e., the gimbal rates. The pseudoinverse steering law, known as the Moore-Penrose pseudoinverse, is one of the simplest and most frequent steering laws. This minimal two-norm approach receives the torques from the controller and computes the gimbal rates using the right pseudoinverse method. This approach seeks to minimize

$$\mathcal{L}(\dot{\gamma}) = \|\dot{\gamma}\|_2^2, \quad (1.1)$$

subject to

$$\mathbf{J}\dot{\gamma} = \dot{\mathbf{h}}_c. \quad (1.2)$$

In this equation,  $\dot{\mathbf{h}}_c \in \mathbb{R}^3$  denotes the command torque,  $\mathbf{J} \in \mathbb{R}^{3 \times N}$  designates the actuator Jacobian, and  $\dot{\gamma} \in \mathbb{R}^N$  represents the gimbal rates. The number of CMGs,  $N$ , may vary depending on the configuration of the CMG cluster. Generally, satellites employ either four or six CMGs for redundancy. Finally, the solution to the optimization problem can be obtained using the Karush–Kuhn–Tucker (KKT) conditions [6], and is given by

$$\dot{\gamma} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}\dot{\mathbf{h}}_c. \quad (1.3)$$

Obviously, both this method and the weighted pseudo-inverse method, that considers a different gain for each CMG in the cluster, do not contemplate the proximity to a singularity. Among the existing solutions that acknowledge the existence of singularities, one can distinguish singularity escape algorithms from singularity avoidance algorithms. Some approaches have both capabilities, as depicted in Fig. 1.4.

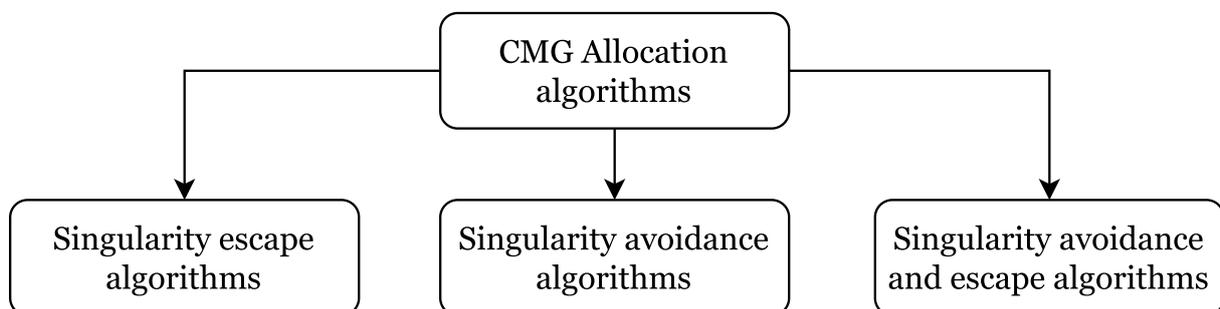


Fig. 1.4: Types of CMG allocation algorithms.

### 1.3.1 Singularity Escape Algorithms

Singularity escape algorithms aim to overcome singularities through small torque deviations from the command torque. The purpose of these algorithms is to keep the conditioning of the actuator's Jacobian

as small as possible. To achieve this, an inexact mapping, from the command torque to the gimbal rates, through a pseudo-inverse method is employed [5].

One of the earliest and most common methods consists in the addition of a diagonal matrix, as given by

$$\dot{\gamma} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + \alpha \mathbf{I}_3)^{-1} \dot{\mathbf{h}}_c, \quad (1.4)$$

where  $\mathbf{I}_3$  is the  $3 \times 3$  identity matrix, and  $\alpha \in \mathbb{R}$  is a scaling parameter. This approach introduces a torque error that depends on the parameter  $\alpha$ . This parameter may be constant or dependent on the manipulability index,  $m = \sqrt{\det(\mathbf{J}\mathbf{J}^T)}$ , as given by

$$\alpha = \alpha_0 e^{-\mu m}, \quad (1.5)$$

where  $\alpha_0$  and  $\mu$  are real, positive constants [7]. By choosing a variable  $\alpha$ , the torque error decreases as the system gets further away from the singularity and increases as it gets closer. The objective of this approach, called the singularity robust inverse method, is to regularize the singular values of the matrix  $\mathbf{J}\mathbf{J}^T$ , so that it is always invertible. The time needed to escape a singularity depends on the choice of the tuning parameters. A system that accepts high torque errors can escape quicker than a system that is not tolerant to significant torque deviations.

An approach called singular direction avoidance, proposed by Ford and Hall [8], relates the torque error to the singular direction only. This method consists of regularizing the smallest singular value of the Jacobian, thus, not promoting torque error in non-singular directions. This is obtained by adding a  $3 \times 3$  matrix,  $\mathbf{X}$ , to the pseudo inverse calculation, as given by

$$\dot{\gamma} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + \mathbf{X})^{-1} \dot{\mathbf{h}}_c, \quad (1.6)$$

where  $\mathbf{X} = \mathbf{u} \text{diag}(0, 0, \alpha) \mathbf{u}^T$ ,  $\mathbf{u} \in \mathbb{R}^3$  is the singular direction, and  $\alpha$  is the same singularity parameter as in the previous method. This method clearly reduces the overall torque error, although it requires an online computation of the Jacobian singular values.

The most common singularity escape algorithm has been proposed by Wie et al. [9] and is a generalized version of the singularity robust inverse method. This method mitigates the gimbal lock effect by adding a time-varying matrix to the pseudo inverse, as given by

$$\dot{\gamma} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + \alpha \mathbf{E})^{-1} \dot{\mathbf{h}}_c, \quad (1.7)$$

with

$$\mathbf{E} = \begin{bmatrix} 1 & \epsilon_0 \sin(\omega_3 t + \phi_3) & \epsilon_0 \sin(\omega_2 t + \phi_2) \\ \epsilon_0 \sin(\omega_3 t + \phi_3) & 1 & \epsilon_0 \sin(\omega_1 t + \phi_1) \\ \epsilon_0 \sin(\omega_2 t + \phi_2) & \epsilon_0 \sin(\omega_1 t + \phi_1) & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 3}, \quad (1.8)$$

where  $t$  denotes the time,  $\epsilon_0 \in \mathbb{R}$  denotes a positive amplitude, and  $\omega_i \in \mathbb{R}$  and  $\phi_i \in \mathbb{R}$  are the frequencies and phase shifts of the dither used to drive the array out the singularity. This approach introduces a wiggling effect to the allocation problem, allowing to escape singularities. However, there is no guaran-

tee on the time required for escape. The advantage of this method over the previous is that it does not require an online computation of the singular values of the Jacobian. However, similarly to the singularity robust inverse method, it introduces a torque error in all directions, which magnitude depends on the tuning parameters.

Other singularity escape steering algorithms are the feedback steering law, published by Pechev [10], which provides a gimbal rate solution without resorting to the inverse computation, and the singularity penetration with unit delay, described in [11], that escapes singularities by using the previous gimbal rate commands. This last method is not meant for attitude tracking manoeuvres due to the amount of accumulated error. Moreover, the torque error introduced in the system, as well as the time required to escape the singularity, cannot be explicitly quantified.

Although these techniques have been proven effective to escape from singularities, these methods are likely impractical when attitude tracking accuracy is of primary importance. In these cases, where torque matching is strictly necessary, other alternatives that do not explore the use of torque error are often preferred.

### 1.3.2 Singularity Avoidance Algorithms

Some types of singularities, i.e., non-degenerate hyperbolic singularities, that will be described in detail later, offer the possibility of avoidance without the introduction of any torque error. Singularity avoidance algorithms make use of null motion to avoid such singularities. This is possible due to the use of a redundant CMG, that allows for infinite different CMG configurations for every command torque. These algorithms are mainly divided into local gradient methods and gimbal angle or angular momentum constraint methods.

Local gradient methods use the gradient of a singularity index to push the gimbals away from the singularities. The underlying formulation of these methods can be described by

$$\dot{\gamma} = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1} \dot{\mathbf{h}}_c + \beta (\mathbf{I}_n - \mathbf{J}^+ \mathbf{J}) \mathbf{d}, \quad (1.9)$$

where  $\beta \in \mathbb{R}$  is a singularity parameter,  $\mathbf{I}_n - \mathbf{J}^+ \mathbf{J}$  is the projection matrix of any vector  $\mathbf{d}$  onto the nullspace of  $\mathbf{J}$ , and  $\mathbf{d} \in \mathbb{R}^n$  is the gradient-search direction, chosen to maximize the distance to the singularity. As can be seen in (1.9), the gimbal rate vector is given by the sum of the minimum two-norm solution, and a singularity avoidance term that lies within the nullspace of the Jacobian. The latter term promotes distancing from the singularity, and does so, producing no net torque. Local gradient methods are extensively studied by Cornick in [12] and by Kuhns and Rodriguez in [13]. In the first work, the CMG cluster is commanded to follow a specific gimbal angle configuration via null space motion while performing an attitude manoeuvre. In the second publication, the distance from the singularity is maximized by computing the gradient of a singularity measure function with respect to the gimbal angles. Kurokawa, in [14], has shown that the use of an appropriate gradient-search direction can be very effective in avoiding non-degenerate hyperbolic singularities for SGCMGs and any kind of singularities for DGCMGs.

Restricting the gimbal angle/angular momentum space is another way of avoiding singularities. With this approach, software or even hardware-based constraints can be defined to limit the motion of the gimbals. These constraints can be either linear or nonlinear. If defined in software, linear constraints are usually preferred due to the low computational cost. However, linear methods may limit the workspace beyond singularities, as these restrictions are usually obtained through convexification methods that approximate the real, usually nonlinear, constraints. The resulting reduced workspace provides a singularity-free environment for gimbal motion. Small manoeuvres, that do not require a huge exploitation of the gimbal space, do not suffer much penalty with this method. However, the performance of agile spacecrafts is typically affected by these methods. In [15], a singularity-free steering law for a roof array of CMGs is accomplished. This solution relies on a feasible angular momentum chart that provides singularity-free bound. In addition, a new deterministic allocation algorithm, called half-leading steering logic, is suggested instead of the common pseudoinverse technique. Other reduced workspace approaches are thoroughly investigated by H. Kurokawa in [14] and [16].

### 1.3.3 Singularity Avoidance and Escape Algorithms

Some singularities, namely hyperbolic singularities, can be avoided with the use of null motion, while others, particularly elliptic singularities, cannot be avoided. In these cases a torque error is acceptable to escape the singularity effect. Many algorithms that comprise both singularity avoidance and escape capabilities are described below.

The first one is a steering algorithm based on the pseudoinverse method, which aims to overcome singularities by adding both singularity avoidance and singularity escape terms to the pseudoinverse. This strategy is called hybrid steering logic and takes the form

$$\dot{\gamma} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + \mathbf{X})^{-1} \dot{\mathbf{h}}_c + \beta \mathbf{V} \text{diag}(0, 0, 0, 1) \mathbf{V}^T \mathbf{d}, \quad (1.10)$$

being an hybrid of the singularity direction avoidance algorithm and the local gradient method. In (1.10),  $\mathbf{V}$  denotes the right-singular vectors of  $\mathbf{J}$  organized in descending order. Singularity parameter metrics to achieve the desired performance are provided in [17].

Optimal control approaches are commonly employed for singularity avoidance and escape. A research done by H. Leeghim et al. [18] computes the optimal gimbal rates via a one-step ahead strategy. This method uses a mathematical model of the system to predict the state of the gimbals one step further in time. By including a singularity index in the cost function, it is possible to maximize the distance to the singularities by penalizing the overall cost. Yet, this approach does not consider the gimbal dynamics as well as the physical limitations of the actuators.

Receding horizon (RH) control was extensively studied by K. Takada and H. Kojima in [19]. This work tackles the low manoeuvring time associated with trajectories that pass through or near singularities, and are therefore critical for agile missions. To overcome this, RH control is used to obtain fast attitude manoeuvre times. Nonetheless, this method prioritizes manoeuvring time over torque error, which is unwanted sometimes. Other optimal control approaches apply RH control, such as [20]. This research

establishes a singular-surface cost function to determine the perturbation torques to be added to the controller output in order to avoid singularities. This method, however, does not consider the null motion for such purpose, hence introducing torque error. Besides this, the algorithm requires large memory space to store all the information regarding the singular surfaces in the angular momentum envelope.

The work [21] describes a model predictive controller for a simple scissored-pair CMG cluster. Within this work, various formulations for both fast manoeuvring and minimum control energy purposes were developed. A hybrid strategy that combines both features was also synthesized. The model predictive controller computes the gimbal rates in an integrated fashion, merging both the controller and control allocation steps into a single optimization problem. The drawback of this approach is that tractability is lost, and a highly complex formulation, that involves multiple constraints, ought to be used. In [22], a nonlinear model predictive controller was developed for attitude tracking of constant references. The determinant of the Jacobian is included in the objective function aiming to penalize the overall cost as the system approaches a singularity. Subsequently, the control torque command given by the nonlinear model predictive controller is translated into gimbal rates using the right pseudoinverse. This approach, however, exhibits very high computational times.

In a recent work published by SENER Aerospace, sequential convex programming was investigated for satellite steering in an agile context [23]. This work divides the attitude control algorithm into two optimization problems. The first is responsible for attitude manoeuvring planning and the second handles CMG allocation. These algorithms are solved in a sequential fashion and present reasonable results with a fairly low computational effort. However, singularity avoidance is not assured but guarantees are given that the slew profile is optimum in time and control required.

Variable speed control moment gyroscopes are also widely studied for singularity avoidance. Even though these actuators are not commonly used in spacecrafts, they offer an additional degree of freedom due to the variable speed rotors, that can be explored for singularity avoidance. Sufficient conditions for non-singular VSCMG steering are achieved under some assumptions in [24]. The model predictive control framework was also applied to these actuators by Y. Yang in [25].

Hardware changes to the actual CMG cluster configurations have also been thought to mitigate singularities. In [26], the addition of an external motor to a pyramidal CMG array was considered. This allowed the steering of the whole CMG cluster along the  $z$  direction, thus introducing an extra degree of freedom used for singularity avoidance.

In [27] and [28] heuristic search algorithms have been derived for gimbal path planning. In the first, rapidly-exploring random trees are considered in tandem with a classical steering law to manoeuvre the spacecraft along a path that minimizes the occurrence of singularities. In the second, a global search algorithm, that determines the null motion added upon the singularity robust inverse steering law, was derived. This algorithm considers global information of the whole manoeuvre, instead of just local information, and relies on a ternary tree where each child denotes negative, zero, or positive null motions.

Machine learning (ML) approaches are becoming popular in spacecraft attitude control. A work published by W. MacKunis et al. [29] uses a neural network-based adaptive attitude controller to steer a

spacecraft in the presence of parametric uncertainties, nonlinear actuator disturbances, unmodeled external disturbances, and uncertainties in the input torque caused by unknown static CMG gimbal friction. ML techniques were also explored by C. Papakonstantinou et al. [30] with some promising results. In his work, learning techniques were used to predict the required null motion for attitude trajectories that were not included in the training dataset.

Different steering algorithms outlined in this section serve different purposes. Some algorithms escape singularities by performing an inexact mapping from the command torque to the gimbal rates. These methods are usually fast from a computational perspective but may be impractical for space missions requiring high-pointing accuracy. Other methods try to avoid singularities by adding null motion to the gimbal rates inputs, as the system gets closer to a singularity. However, these methods disregard elliptical singularities, where avoidance is impossible. Limiting the work environment of the gimbals is also a common solution. However, the discontinuities introduced in the gimbal/angular momentum envelope might promote significant torque error in the system if the gimbal trajectory is not carefully planned. Finally, strategies resorting to the optimal control framework are studied. These strategies explore the redundant space of solutions of the system, as well as the acceptance of torque error in situations where avoiding singularities is infeasible.

## 1.4 Contributions

Leading-edge research has mostly focused on the development of steering laws for both singularity avoidance and escape. Optimal control strategies are capable of both and allow for constraint handling, which is paramount for achieving optimal performance while accounting for the physical limitations of the actuators. Nonetheless, optimal control algorithms are usually computationally heavy and inefficient. The formulations can sometimes be complex and highly nonlinear, leading to non-convex optimization problems. Given this, the solution proposed in this thesis aims to fill the gap that separates high performance and low computational effort. An end-to-end controller that directly computes gimbal rate commands given the attitude error does not generally cope with these requirements. Therefore, there is a strong commitment throughout this work to find a decoupled and complementary strategy to a known standard controller. The proposed solution exhibits both high performance, through the synthesis of an optimal CMG allocation algorithm, and low computational cost, as convexity is later achieved. Finally, the outcome of this dissertation contributes to the state-of-the-art as follows:

- A closed-form description of the singularity envelope is accomplished. For some CMG clusters, the singularities are organized in lines or planes in the gimbal space. This information is leveraged for singularity avoidance by assessing the Euclidean distance between the gimbals and the closest singularity in the envelope. According to the best of the author's knowledge, this is the first time that the distance to the singularities is exploited for singularity avoidance.
- A novel algorithm for CMG allocation is derived. It consists of an add-on allocation technique to a literature controller, that receives the torques computed by the controller, and outputs a sequence

of gimbal rates that lead to singularity-free gimbal configurations. This solution resorts to the model predictive control framework and employs a singularity term that considers the Euclidean distance to the closest singularity.

- A convex approximation to the singularity term is suggested. This overcomes the excessive computational cost usually associated with an optimal control approach, resulting in a convex optimization problem.

## 1.5 Thesis Outline

This thesis is organized as follows:

- **Chapter 2: Background.** In this chapter, some theoretical concepts used during this dissertation are explained. Firstly, the principle of operation of different spacecraft actuators is presented. Further, a comparison of different attitude representation methods is made. This section explains the reason why unit quaternions are commonly preferred. Following this, a short introduction to the singular value decomposition of rectangular matrices is given. Then, concepts such as the nullspace of a matrix and the solution space of an underdetermined system of equations are revised. Later, some principles of optimal control theory, including convex optimization, are disclosed. Finally, the action governor and the model predictive control frameworks are briefly described.
- **Chapter 3: Model Definition.** Here, the mathematical description of the satellite angular motion is presented. To start, the actuator dynamics of two different control moment gyro arrays are derived. Then, the satellite attitude dynamics based on the principle of conservation of angular momentum is presented. At last, the kinematics are described via a quaternion-based formulation.
- **Chapter 4: Singularity Analysis.** Before presenting the controller and control allocation algorithms, an analytical and visual analysis of the actuator singularities is performed. Indeed, in this part of the document, the underlying foundations of a singularity are explained. In parallel, the Jacobian of two different CMG clusters is analysed with particular focus on singularity situations.
- **Chapter 5: Controller Design and Control Allocation.** In this chapter, an optimal control strategy is derived to overcome the singularity problem. To accomplish this, a literature controller is firstly used to compute the required torques for a given manoeuvre, and then, an optimal convex control allocation framework is proposed to calculate the actuator inputs.
- **Chapter 6: Simulation Results.** The results achieved are shown and compared with some literature solutions. Various simulations based on different formulations and optimization parameters are reported and explained in detail.
- **Chapter 7: Conclusions and Future Work.** The conclusions accomplished during this dissertation are presented, as well as some directions for future work on this research topic.

# Chapter 2

## Background

In this part of the document, the fundamentals underlying the methodologies used in this dissertation are covered. Firstly, some of the most common spacecraft actuators are introduced. After this, an explanation follows on why quaternions are the preferred attitude representation method for spacecraft control. Later, some algebra concepts, such as the solution to an underdetermined system of linear equations and singular value decomposition, are briefly explained. These sections reveal the foundation for singularity analysis, that is later performed in Chapter 4. Thereafter, optimization theory and convexity are succinctly addressed. Finally, optimal control strategies, such as the action governor and the model predictive control framework, are described. The principles disclosed in these last sections constitute the foundations of the main achievements of this dissertation.

### 2.1 Spacecraft Actuators

#### 2.1.1 Magnetic Torque Rods

Most low Earth orbit satellites make use of magnetic torque rods due to their proximity to the Earth's surface. Magnetic torque rods consist of planar coils of wire, rigidly placed along the spacecraft axes, that tend to align their magnetic field according to the Earth's magnetic field. The change in the rod's angular momentum,  $\mathbf{m}$ , causes a magnetic torque,  $\mathbf{t}$ , that is given by

$$\mathbf{t} = \mathbf{m} \times \mathbf{r}_m, \quad (2.1)$$

where  $\mathbf{r}_m$  denotes the geomagnetic field at that point of the orbit [31].

#### 2.1.2 Thrusters

Thrusters constitute a different type of actuators. These devices operate propellants, such as cold gas propellant, solid chemical propellant, liquid chemical propellant or even electrical propellant. In the case of the first three, chemicals have to be kept in tanks, making the amount of stored fuel a major

limitation of the use of these actuators. Moreover, thrusters are not generally a good fit for precision manoeuvres as they are meant to produce large amounts of torque. Depending on the location where these devices are mounted relative to the center of mass, the torque magnitude is proportional to the normal component of the applied force,  $f$ . This last one is derived from the well known Reynolds transport theorem, with magnitude

$$f = v_e \frac{dm}{dt} + a_e(p_e - p_a), \quad (2.2)$$

where  $v_e$  is the exhaust velocity,  $\frac{dm}{dt}$  is the propellant consumption rate,  $a_e$  is the nozzle exit area,  $p_e$  is the nozzle exit pressure, and  $p_a$  is the ambient pressure [31].

### 2.1.3 Reaction and Momentum Wheels

Reaction wheels and momentum wheels are very similar in the way that both employ flywheels driven by electric motors. A reaction wheel is spun up or down to create the torque and force the vehicle to rotate. A momentum wheel is constantly spinning at a very high speed which induces the stabilization of the spacecraft, making it resistant to attitude changes caused by external disturbances. The torques generated by these devices are achieved by accelerating or decelerating the flywheel and are given by

$$\mathbf{t} = \frac{d\mathbf{h}_\omega}{dt} = \mathbf{J}_\omega \dot{\boldsymbol{\omega}}_\omega, \quad (2.3)$$

where  $\mathbf{h}_\omega$  is the angular momentum vector of the flywheel,  $\mathbf{J}_\omega$  is the inertia of the flywheel, and  $\dot{\boldsymbol{\omega}}_\omega$  is its angular acceleration [31]. A spacecraft requires at least three reaction/momentum wheels for full controllability, i.e., pitch, roll, and yaw control.

### 2.1.4 Control Moment Gyros

A control moment gyro is another type of ACS. A CMG generates gyroscopic torque by spinning a flywheel, similarly to reaction wheels, but comprises a motorized gimbal capable of tilting the rotor's angular momentum. Accordingly, the output torque of a CMG is given by

$$\mathbf{t} = \frac{d\mathbf{h}}{dt} = \mathbf{h} \times \dot{\boldsymbol{\gamma}}, \quad (2.4)$$

where  $\mathbf{h}$  denotes the CMG angular momentum and  $\dot{\boldsymbol{\gamma}}$  represents the gimbal spinning rate [31].

## 2.2 Attitude Representations

There are several methods to describe the rotational kinematics of a rigid body in three dimensions. Among the existing methods, Euler angles, axis-angle vectors, rotation matrices, and unit quaternions are the most popular.

Euler angles are one of the most frequently used methods for representing an object's orientation. Three different angles designated by roll, pitch, and yaw ( $\phi, \theta, \psi$ ) are used for representing rotations about

the cartesian  $x, y$  and  $z$  axes. The order of these rotations is important as a roll-pitch-yaw rotation can result in a different final orientation than that of applying the same angles in a yaw-pitch-roll sequence. One of the major drawbacks of this representation is the existence of a singularity at  $\theta = \frac{\pi}{2}$ . In this configuration, both roll and yaw perform the same operation. There is also a discontinuity in the angles at  $\pm\pi$ . The axis-angle vector representation, also known by Euler's principal rotation theorem, parameterize any arbitrary orientation shift,  $\theta$ , by a single rotation about a principal axis  $\hat{e}$ , with rotation angle  $\theta$ . Even though this representation is minimal, similarly to the Euler Angles, the rotations are described by nonlinear trigonometric equations, which can be computationally expensive. Besides this, it exhibits ambiguity as a rotation of  $2\pi$  rad yields the exact same orientation. Rotation matrices are the most general form of representing the attitude of a rigid body. This method requires nine parameters to fulfill a  $3 \times 3$  matrix. Rotation matrices satisfy six constraints as each column of the matrix is a unit vector, and the columns and rows are orthogonal to each other. This makes this representation significantly larger than the others, and consequently, computationally expensive. Yet, this representation is often adopted as it presents no singularities or ambiguities. Finally, unit quaternions are the method that holds the most benefits, as they do not present any singularity, just like the rotation matrices, but they also do not require many parameters. Besides this, the kinematics equations along with the rotation operations are not complex, making them more efficient than the Euler angles. The major drawback of this representation is that a physical understanding of a rotation by a unit quaternion is generally hard to grasp. Besides this, the quaternions  $-\bar{q}$  and  $\bar{q}$  encode the exact same rotation. There is also the problem of unwinding and the chattering that may occur when the scalar part is zero, causing a positive and a negative angle to represent the same attitude.

Finally, weighing the pros and cons of all these methods, it is clear that unit quaternions are the most advantageous attitude representation method. Table 2.1 summarizes the differences between each of these methods [4].

Table 2.1: Comparison of attitude representation methods.

	Number of parameters	Kinematics	Rotation form	Singularity	Ambiguity	Discontinuity
<b>Euler angles</b>	3	Nonlinear	Trigonometric	At $\theta = \frac{\pi}{2}$	$\phi$ and $\psi$ equivalent at $\theta = \frac{\pi}{2}$	At $\pm\pi$
<b>Axis-angle vector</b>	3	Linear	Trigonometric	None	At $\theta = \pm 2k\pi$ $k \in \mathbb{Z}$	None
<b>Rotation matrices</b>	9	Bilinear	Linear	None	None	None
<b>Unit quaternions</b>	4	Bilinear	Quadratic	None	Double cover of $SO(3)$	None

## 2.3 Singular Value Decomposition

Singular value decomposition (SVD) can be interpreted as a factorization of a real or complex matrix. SVD generalizes the eigen-decomposition of a square  $m \times m$  matrix to any rectangular matrix  $m \times n$ . From a practical standpoint, SVD can be seen as a method for identifying the principal components of

a dataset. These are the directions along which datapoints display the most variation. SVD can also be used for data reduction by using the directions of most variation, and therefore, approximate the original data with fewer datapoints. This is widely used for image compression, for example. SVD is also an important tool for finding the coefficients of a reduced rank regression or ridge regression. In the robotics domain, SVD can be applied to analyse the conditioning of a dynamic matrix and conclude about the presence of a singular direction. This last application is considered later in this document.

The singular value decomposition of a matrix consists in breaking it down into the product of three matrices - an orthogonal matrix  $U$ , a diagonal matrix  $\Sigma$ , and the transpose of an orthogonal matrix  $V$ . The SVD of a  $m \times n$  matrix  $M$  can be written as

$$M = U\Sigma V^T, \quad (2.5)$$

where  $U$  is a  $m \times m$  complex unitary matrix,  $\Sigma$  is a  $m \times n$  rectangular diagonal matrix, and  $V$  is a  $n \times n$  complex unitary matrix. A geometric interpretation of (2.5) is provided in Fig. 2.1.

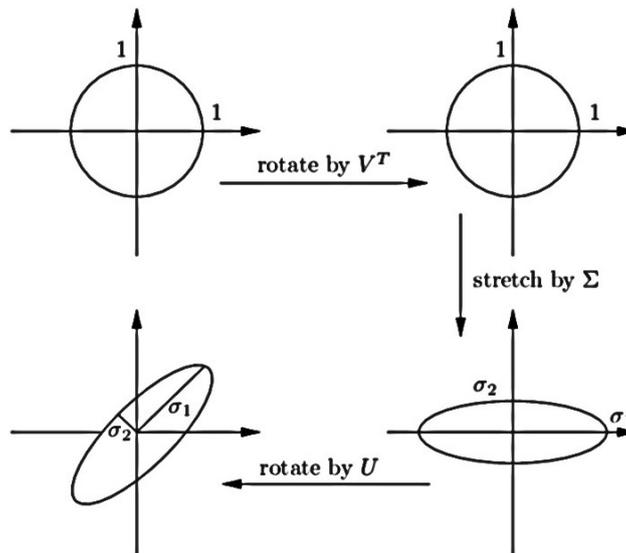


Fig. 2.1: Geometric interpretation of SVD (Image credits: [32]).

The matrices  $U, \Sigma$ , and  $V$  are given as follows.

- $U$ : Consists of the normalized left singular vectors of  $M$ . In other words, this matrix contains the eigenvectors of  $MM^T$  organized by columns and arranged in descending order. This depends on the magnitude of the corresponding eigenvalues.
- $\Sigma$ : Contains the singular values of  $M$  displayed in descending order along the diagonal. These are computed by taking the square root of the eigenvalues of  $MM^T$  (or  $M^T M$ ). The maximum number of singular values is equal to the minimum number of rows or columns of the matrix.
- $V$ : Consists of the normalized right singular vectors of  $M$ . In other words, this matrix contains the eigenvectors of  $M^T M$  organized by columns and arranged in descending order.

Finally, expanding (2.5) results in

$$M = \begin{pmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_k & \mathbf{u}_{k+1} & \dots & \mathbf{u}_m \end{pmatrix} \begin{pmatrix} \sigma_1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_k & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^T \\ \dots \\ \mathbf{v}_k^T \\ \mathbf{v}_{k+1}^T \\ \dots \\ \mathbf{v}_n \end{pmatrix}, \quad (2.6)$$

where  $k$  represents the index of the smallest non-zero singular value. If  $k = \min\{m, n\}$ , the matrix  $M$  is full rank. If  $k < \min\{m, n\}$ , the matrix  $M$  is not full rank [33].

## 2.4 Underdetermined Systems

In linear algebra, a system of linear equations is considered underdetermined if there are less equations than unknown variables. On the other hand, it is considered overdetermined if there are more equations than unknowns. Consider the following linear system:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (2.7)$$

where  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{x} \in \mathbb{R}^n$ , and  $\mathbf{b} \in \mathbb{R}^m$ . Note that  $\mathbf{x}$  is the vector of unknowns. If  $m < n$ , (2.7) is an underdetermined system that typically has an infinite number of solutions.

The solution of the homogeneous system

$$\mathbf{A}\mathbf{x} = 0, \quad (2.8)$$

is a subset of  $\mathbb{R}^n$  called the nullspace of the matrix  $\mathbf{A}$  [34]. Designated by  $\mathcal{N}(\mathbf{A})$ , its dimension depends on both the number of unknowns and the rank of the matrix  $\mathbf{A}$ , being given by

$$\dim(\mathcal{N}(\mathbf{A})) = n - \text{rank}(\mathbf{A}). \quad (2.9)$$

Among the existing solutions of (2.7), some may be more relevant from a physical standpoint. For example, there is interest in finding the solution that minimizes the 2-norm of the vector  $\mathbf{x}$ . By doing this, the system meets a certain requirement using the least energy possible. This translates into finding the solution of the cost function

$$\mathcal{L} = \|\mathbf{x}\|_2^2, \quad (2.10)$$

subject to

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \quad (2.11)$$

The solution to this optimization problem can be obtained using the KKT conditions [6] and is given by

$$\mathbf{x}_0 = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b}. \quad (2.12)$$

Finally, the solution space of (2.7) can be described by the sum of the minimum two-norm solution with a nullspace component, as given by

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{N}\boldsymbol{\lambda}, \quad (2.13)$$

where  $\mathbf{N} \in \mathbb{R}^{n \times (n - \text{rank}(\mathbf{A}))}$  is the normalized nullspace and  $\boldsymbol{\lambda} \in \mathbb{R}^{n - \text{rank}(\mathbf{A})}$  is a scaling vector.

These results will be particularly useful when exploring the null motion of the CMG array, provided by the use of a redundant CMG.

## 2.5 Optimization Problems and Convexity

An optimization problem consists of finding the best solution to a problem from all the possible solutions. The standard form of an optimization problem is

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && f(\mathbf{x}) \\ & \text{subject to} && h_i(\mathbf{x}) = 0, \quad i = 1, \dots, m, \\ & && g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, p \end{aligned} \quad (2.14)$$

where

- $\mathbf{x} \in \mathbb{R}^n$  is the optimization variable,
- $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is the cost function to be minimized,
- $h_i(\mathbf{x}) = 0$  are the equality constraints, and
- $g_j(\mathbf{x}) \leq 0$  are the inequality constraints.

The solution to an optimization problem can also result from a maximization problem. This is equivalent to minimizing the symmetric of its cost function. However, the most standard way consists of a minimization problem. If there are no constraints, i.e.  $m = p = 0$ , the optimization problem is called unconstrained.

Depending on the nature of the cost and constraint functions, one can distinguish between two different types of optimization problems. The first concerns convex optimization. In a convex optimization problem, optimality is guaranteed, i.e., if there exists a local minimum, that local minimum is also a global minimum. A convex optimization problem can be solved efficiently up to a very large size. To guarantee the convexity of an optimization problem, the following conditions have to be met [6]:

- the cost function  $f(\mathbf{x})$  has to be convex (see Fig. 2.2),
- the equality constraints have to be affine maps, and

- the inequality constraints have to be convex.

Gathered all these conditions, there is the presence of a convex cost function and a convex feasible region. If any of these conditions cannot be verified, convexity is not guaranteed, meaning that the optimization problem is likely non-convex. Such problem may have multiple feasible regions and various optimal points within each region. In this domain, reaching a local minimum does not mean global optimality, as non-convex problems generally have multiple local minima. This type of optimization problems are typically hard to solve and can be computationally expensive. An example of a non-convex function is provided in Fig. 2.3.

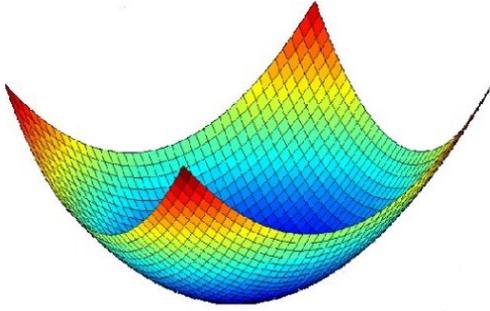


Fig. 2.2: Convex function (Image credits: [35]).

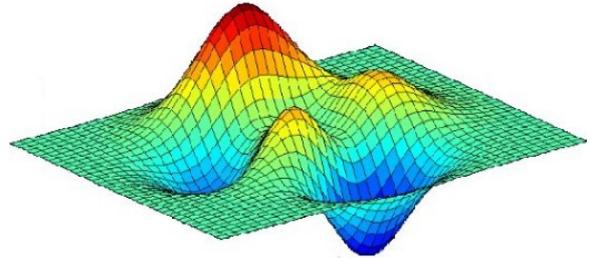


Fig. 2.3: Non-convex function (Image credits: [35]).

## 2.6 Action Governor

Consider a system represented by the discrete-time model

$$\mathbf{x}(k+1) = f_d(\mathbf{x}(k), \mathbf{u}(k)), \quad (2.15)$$

where  $\mathbf{x}(k) \in \mathbb{R}^m$  represents the system state at the discrete time instant  $k$ , with  $k \in \mathbb{R}_0^+$ , and  $\mathbf{u}(k) \in \mathbb{R}^n$  represents the control input. It is assumed that the nominal control policy

$$\mathbf{u}_\phi(k) = \phi(\mathbf{x}(k), \mathbf{r}(k), k) \quad (2.16)$$

has been defined for the system (2.15), where  $\mathbf{r} \in \mathbb{R}^p$  represents the reference signal. The control policy  $\phi$  may be linear, nonlinear, time-invariant, or time-variant. Furthermore, consider that the system state  $\mathbf{x}(k)$  is subject to an exclusion zone requirement of the form

$$\mathbf{x}(k) \notin \mathcal{X}_0, \quad \forall k \in \mathbb{Z}_0^+, \quad (2.17)$$

where the set of all feasible  $\mathbf{x}$  is denoted by  $\mathcal{X}$ . The nominal control policy does not necessarily contemplate the exclusion zone requirement (2.17). Therefore, there may be situations where the system state lies in  $\mathcal{X}_0$ , which may lead to undesired consequences. The goal of the algorithm proposed in [36] aims to enforce (2.17) by monitoring and minimally modifying the nominal input  $\mathbf{u}_\phi$  when necessary. This is realized through an add-on supervisory scheme, referred to as Action Governor (AG), as depicted in

Fig. 2.4.

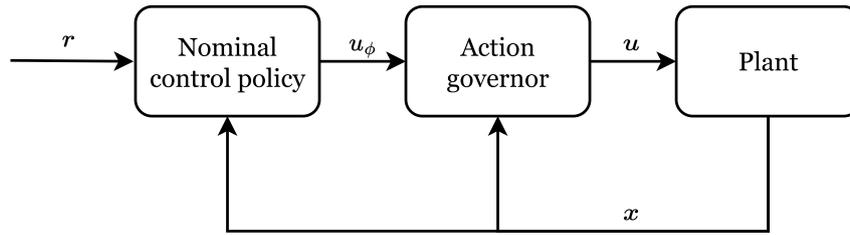


Fig. 2.4: Action governor scheme.

The AG consists of an online constrained optimization problem of the form

$$\begin{aligned} & \text{minimize} && \|u(k) - u_\phi(k)\|_S^2 && (2.18a) \\ & \mathbf{u}(k) \in \mathcal{U} \end{aligned}$$

$$\text{subject to } f_d(\mathbf{x}(k), \mathbf{u}(k)) \in \mathcal{X} \setminus \mathcal{X}_0, \quad (2.18b)$$

where  $\mathcal{U}$  represents the range of control inputs, and the function  $\|\cdot\|_s = \sqrt{(\cdot)^T \mathbf{S}(\cdot)}$ , with  $\mathbf{S} \in \mathbb{R}^{n \times n}$  being a positive-definite matrix, is used to penalize the difference between the nominal control command  $\mathbf{u}_\phi$  and the modified control command  $\mathbf{u}$  that satisfies (2.18b) [36][37].

## 2.7 Model Predictive Control

Model predictive control (MPC) is an advanced control method typically used to control a system while satisfying a set of constraints. MPC uses a mathematical model to predict the future behaviour of the system and take control actions accordingly. The model used is often obtained theoretically, given prior knowledge of the physics of the system. It can also be determined empirically through experiments and system identification methods. A graphical overview of the MPC structure is provided in Fig. 2.5. A

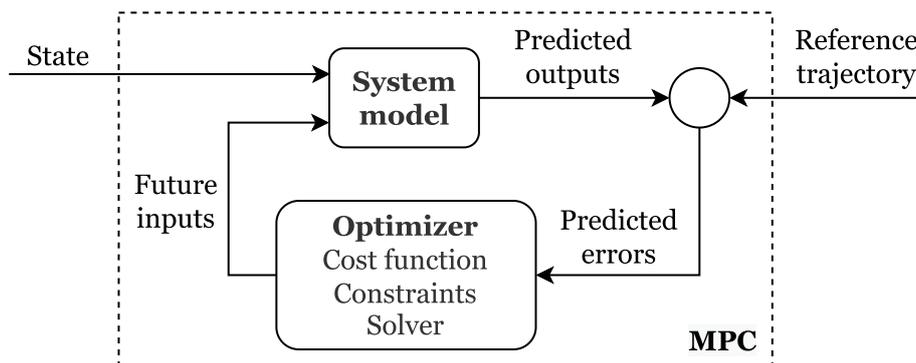


Fig. 2.5: MPC structure overview.

model is derived to predict the system outputs based on the current state of the system and the computed inputs. These control actions, that drive the system to the setpoints, generally result from solving a constrained optimization problem. The inputs to the MPC are a sequence of setpoints over a finite duration time-window. Finally, the integration of an MPC is similar to conventional controllers, as it also requires state feedback for the computation of the control actions (see Fig. 2.6). Notwithstanding, the

combination of prediction and optimization is the main difference from conventional control approaches, which use precomputed control laws [38].

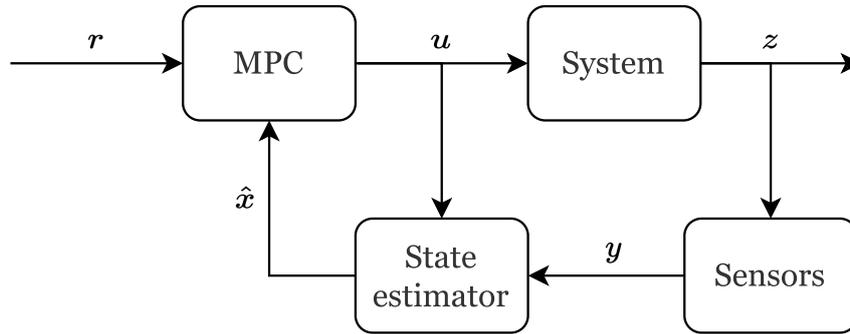


Fig. 2.6: MPC-based control diagram.

The operating principle of an MPC is visually explained in Fig. 2.7 and can be described as follows:

- The future outputs for a given prediction horizon  $N_p$ , are predicted sequentially using a dynamic model of the system. The predicted outputs,  $y(t+k|t), k = 0, \dots, N_p - 1$ , depend on the state of the system and the computed control signals up to the time instant  $t - 1$ .
- The sequence of future control actions is determined by optimizing a cost function under specific constraints. The cost function usually comprises a term that penalizes deviations from the setpoint over the prediction horizon and a term to regularize the control effort. The constraints of the optimization problem include the model equations that dictate the next state of the system given its current state and the input. These may also include saturation functions, input rate limiters, or even state restrictions. Usually, state constraints can be relaxed by being penalized in the cost function instead.
- The first control action in the control sequence  $u$  is applied to the system while the remaining control signals are discarded. On the next sampling instant, the control sequence is re-computed for a different state  $x$ . The control horizon,  $N_c$ , dictates the number of optimized control actions. If the control horizon is smaller than the prediction horizon, the last computed input is kept constant for the remaining length of the horizon [38] [39].

Lastly, Table 2.2 summarizes the advantages and disadvantages of MPC over conventional controllers.

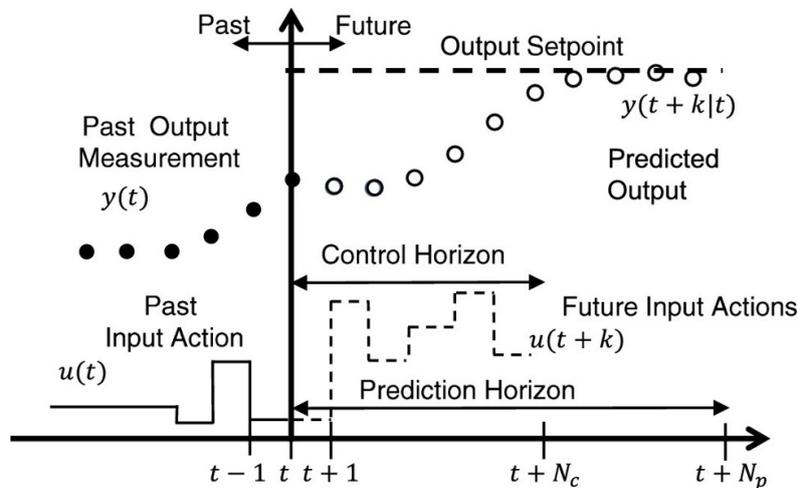


Fig. 2.7: MPC operating principle (Image credits: [38]).

Table 2.2: Advantages and disadvantages of MPC over conventional controllers.

Advantages	Disadvantages
<p><b>Optimality.</b> Uses an optimal solution that is tailored to the choice of the operational constraints and the cost function.</p>	<p><b>Complex derivation.</b> An MPC accounts for complex model dynamics, interactions among the various system variables, and delay between the execution of a control action and its effect on the state variables.</p>
<p><b>Multivariable controller.</b> Extension to the multivariable case is straightforward. Easily handles multiple inputs and outputs.</p>	<p><b>High number of control parameters.</b> An MPC can be difficult to tune depending on the number of optimization weights in the cost function.</p>
<p><b>Constraint handling.</b> An MPC can handle numerous constraints. This is an important feature as violating constraints may lead to undesirable consequences.</p>	<p><b>High computational cost.</b> Requires large amounts of calculations, especially when the number of constraints is high, or when using long horizons.</p>
<p><b>Customizable.</b> Easily shaped to the specifics of the current problem and is fairly simple to implement.</p>	

# Chapter 3

## Model Definition

In this chapter, the satellite's angular motion dynamics are derived using a state-space description. The satellite model comprises different steps as suggested in Fig. 3.1. Firstly, the CMG receives the gimbal commands,  $\dot{\gamma}$ , from the ACS. Consequently, this triggers gimbal motion in the CMG array that translates into a reaction torque,  $\dot{h}$ , applied to the spacecraft. The satellite dynamics laws convert the incoming torques into angular accelerations and angular velocities,  $\omega$ . Finally, the satellite kinematics are propagated using attitude quaternions,  $\bar{q}$ .

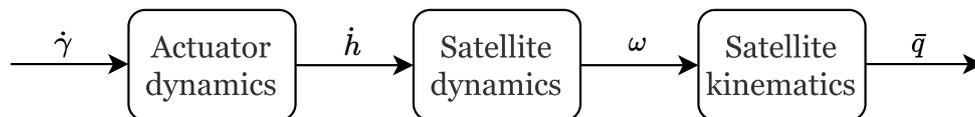


Fig. 3.1: Satellite model breakdown.

### 3.1 Actuator Dynamics

Two different CMG arrays will be considered throughout this document. The first consists of three different control moment gyros arranged in a way such that every CMG sits on a vertex of an equilateral triangle (Fig. 3.2). This is a simple model, not commonly used in real spacecrafts as it only provides roll and pitch control. However, this model will be frequently mentioned in this report for singularity visualization purposes. Moreover, it contains a convenient singular space that will be further explored to demonstrate the potential of the proposed control strategies. To compensate for yaw control and achieve full 3-dimensional controllability, a reaction wheel is later added to the  $z$  direction. The second model is a roof array frequently utilized for spacecraft control. This version, employed by [15], considers two pairs of CMGs that are orthogonally equipped. Each pair of CMGs can produce torque along two different axis. As can be observed in Fig. 3.3, the pair composed by CMG #1 and CMG #2 can generate torque in the  $xz$  plane, while the pair composed by CMG #3 and CMG #4 only creates torque in the  $yz$  plane. As each pair of CMGs is capable of acting in the  $z$  direction, the capacity of angular momentum and torque along the  $z$  axis is twice as large as in the other directions.

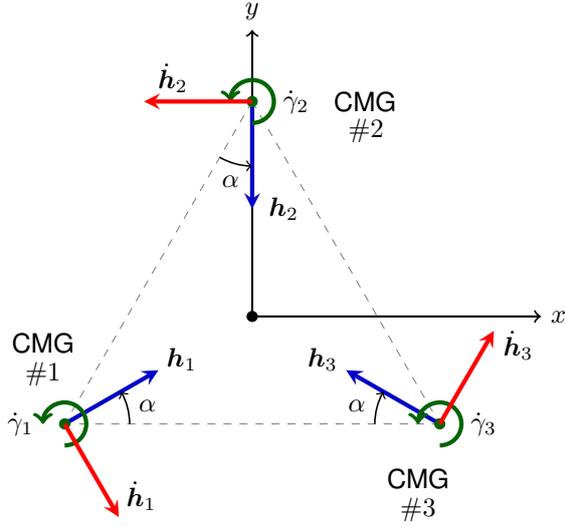


Fig. 3.2: Triangular array of CMGs.

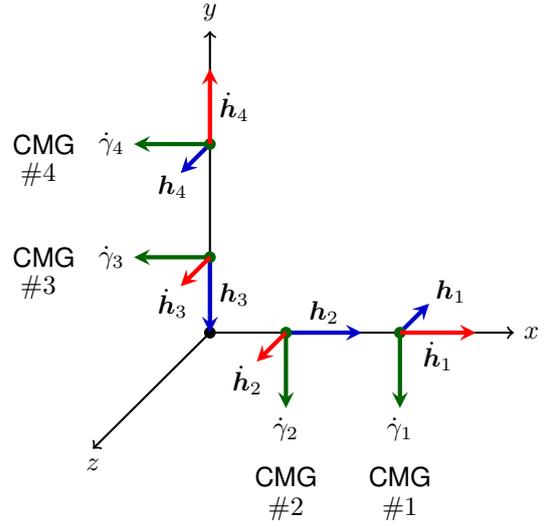


Fig. 3.3: Roof array of CMGs.

Designating the momentum of the  $i$ -th CMG as  $\mathbf{h}_i$ , the angular momentum of the entire system is given by the sum of the angular momentum produced by each individual CMG, i.e.,

$$\mathbf{h} = \sum_{i=1}^N \mathbf{h}_i, \quad (3.1)$$

where  $N$  denotes the number of CMGs in the system. Taking this into consideration, the total angular momentum of the triangular array is

$$\mathbf{h} = h_0 \begin{pmatrix} \cos(\alpha + \gamma_1) \\ \sin(\alpha + \gamma_1) \end{pmatrix} + h_0 \begin{pmatrix} \sin \gamma_2 \\ -\cos \gamma_2 \end{pmatrix} + h_0 \begin{pmatrix} -\cos(\alpha - \gamma_3) \\ \sin(\alpha - \gamma_3) \end{pmatrix} \in \mathbb{R}^2, \quad (3.2)$$

where  $h_0 \in \mathbb{R}$  is the initial momentum given by the rotation speed of the flywheels,  $\alpha$  equals  $30^\circ$  for equilateral triangle configurations, and  $\boldsymbol{\gamma} \in \mathbb{R}^N$  is the gimbal angle vector, with  $N = 3$ .

The derivative of the angular momentum can be computed by applying the chain rule to (3.2), yielding

$$\dot{\mathbf{h}}(\boldsymbol{\gamma}) = \frac{\partial \mathbf{h}}{\partial \boldsymbol{\gamma}} \frac{\partial \boldsymbol{\gamma}}{\partial t} = \mathbf{J}(\boldsymbol{\gamma}) \dot{\boldsymbol{\gamma}}, \quad (3.3)$$

where  $\mathbf{J}$  represents the Jacobian matrix, given by

$$\mathbf{J}(\boldsymbol{\gamma}) = h_0 \begin{pmatrix} -\sin(\alpha + \gamma_1) & \cos \gamma_2 & -\sin(\alpha - \gamma_3) \\ \cos(\alpha + \gamma_1) & \sin \gamma_2 & -\cos(\alpha - \gamma_3) \end{pmatrix} \in \mathbb{R}^{2 \times 3}. \quad (3.4)$$

The same reasoning applies for the roof array. The angular momentum and the Jacobian for this configuration are given by, respectively,

$$\mathbf{h}(\boldsymbol{\gamma}) = h_0 \begin{pmatrix} \sin \gamma_1 \\ 0 \\ -\cos \gamma_1 \end{pmatrix} + h_0 \begin{pmatrix} \cos \gamma_2 \\ 0 \\ \sin \gamma_2 \end{pmatrix} + h_0 \begin{pmatrix} 0 \\ -\cos \gamma_3 \\ \sin \gamma_3 \end{pmatrix} + h_0 \begin{pmatrix} 0 \\ \sin \gamma_4 \\ \cos \gamma_4 \end{pmatrix} \in \mathbb{R}^3, \quad (3.5)$$

and

$$\mathbf{J}(\boldsymbol{\gamma}) = h_0 \begin{pmatrix} \cos \gamma_1 & -\sin \gamma_2 & 0 & 0 \\ 0 & 0 & \sin \gamma_3 & \cos \gamma_4 \\ \sin \gamma_1 & \cos \gamma_2 & \cos \gamma_3 & -\sin \gamma_4 \end{pmatrix} \in \mathbb{R}^{3 \times 4}. \quad (3.6)$$

The angular momentum envelopes for both models are presented in Fig. 3.4. It can be seen that the triangle layout is able to produce uniform amounts of momentum in every direction in the  $xy$  plane. On the other hand, the roof array has the capability of producing twice as much as momentum in the  $z$  direction, comparing to the  $x$  and  $y$  directions. Note that the results shown are dimensionless. In fact, the real angular momentum envelope is only achieved after scaling these results by  $h_0$ .

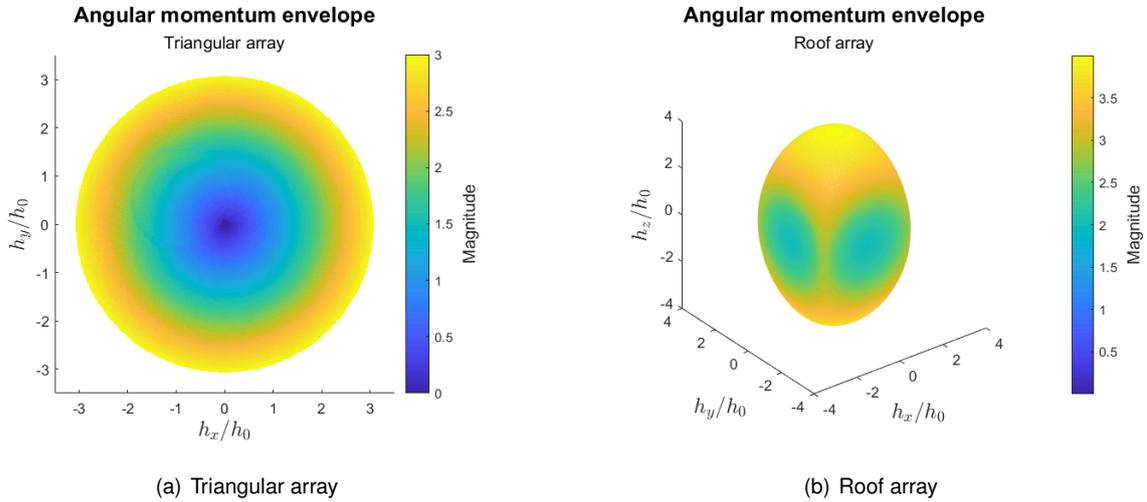


Fig. 3.4: Angular momentum envelopes.

Finally, to simulate the behaviour of a real actuator, dynamics and saturation must be considered. A rather simple actuator model that considers the dynamics and the physical limitations of a real CMG is provided in Fig. 3.5. This model introduces a first-order system used to replicate the actuator dynamics



Fig. 3.5: Actuator modelling.

as given by

$$\ddot{\gamma} = K(-\dot{\gamma} + \dot{\gamma}_d), \quad (3.7)$$

with a real pole at  $-K$  rad/s and unitary gain. Additionally, the output of the integrator is constrained to a known interval to account for saturation effects.

## 3.2 Satellite Dynamics

Control moment gyros exploit the principle of conservation of angular momentum to steer the spacecraft. The rate of change of angular momentum equals the sum of the external torques applied on the spacecraft. In vector form, this translates into

$$\left. \frac{d\mathbf{h}_s}{dt} \right|_{\mathcal{I}} = R_{\mathcal{I} \leftarrow \mathcal{B}}(\boldsymbol{\tau}), \quad (3.8)$$

where  $\mathbf{h}_s \in \mathbb{R}^3$  denotes the total angular momentum of the spacecraft and  $\boldsymbol{\tau} \in \mathbb{R}^3$  represents the external torques. Furthermore, the derivative is evaluated in an inertial frame, represented by the index  $\mathcal{I}$ , and the external torques have to be rotated from the body axes,  $\mathcal{B}$ , to the inertial axes,  $\mathcal{I}$ , through the rotation operation  $R_{\mathcal{I} \leftarrow \mathcal{B}}$ . To express these quantities in the body frame, which is non-inertial, one has to include the gyroscopic term in the equation, as given by

$$R_{\mathcal{B} \leftarrow \mathcal{I}} \left( \left. \frac{d\mathbf{h}_s}{dt} \right|_{\mathcal{I}} \right) = \left. \frac{d\mathbf{h}_s}{dt} \right|_{\mathcal{B}} + \boldsymbol{\omega} \times \mathbf{h}_s, \quad (3.9)$$

where  $\boldsymbol{\omega} \in \mathbb{R}^3$  is the angular velocity of the body axes with respect to the inertial axes, expressed in the body frame [40]. For the sake of simplicity, it is assumed that the actuator frame coincides with the body frame.

The total angular momentum of the spacecraft can be written as

$$\mathbf{h}_s = \mathbf{I}_s \boldsymbol{\omega} + \mathbf{h}, \quad (3.10)$$

where the first term denotes the angular momentum respective to the satellite motion and the second term represents the angular momentum produced by the CMG system. In this equation,  $\mathbf{I}_s \in \mathbb{R}^{3 \times 3}$  designates the matrix of inertia of the spacecraft. Since spacecrafts can be regarded as rigid bodies, the time derivative of  $\mathbf{h}_s$  with respect to the body axis simplifies to

$$\left. \frac{d\mathbf{h}_s}{dt} \right|_{\mathcal{B}} = \underbrace{\dot{\mathbf{I}}_s \boldsymbol{\omega}}_{=0} + \mathbf{I}_s \dot{\boldsymbol{\omega}} + \dot{\mathbf{h}} = \mathbf{I}_s \dot{\boldsymbol{\omega}} + \dot{\mathbf{h}}. \quad (3.11)$$

Substituting (3.11) and (3.10) in (3.9), and using (3.8), yields

$$\mathbf{I}_s \dot{\boldsymbol{\omega}} + \dot{\mathbf{h}} + \boldsymbol{\omega} \times (\mathbf{I}_s \boldsymbol{\omega} + \mathbf{h}) = \boldsymbol{\tau}. \quad (3.12)$$

Isolating the internal torque produced by the CMGs, it follows that

$$\begin{cases} \mathbf{I}_s \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}_s \boldsymbol{\omega} = \boldsymbol{\nu} + \boldsymbol{\tau}, \\ \boldsymbol{\nu} = -(\dot{\mathbf{h}} + \boldsymbol{\omega} \times \mathbf{h}), \end{cases} \quad (3.13)$$

where  $\boldsymbol{\nu} \in \mathbb{R}^3$  denotes the reaction torque expressed in the inertial frame. If no external torques are applied to the spacecraft, i.e.,  $\boldsymbol{\tau} = 0$ , the total angular momentum of the system is conserved, and so,

any momentum change provoked by the actuator triggers a reaction torque on the spacecraft with the same magnitude but opposite direction, i.e.

$$\mathbf{I}_s \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}_s \boldsymbol{\omega} = -(\dot{\mathbf{h}} + \boldsymbol{\omega} \times \mathbf{h}). \quad (3.14)$$

Finally, solving (3.12) with respect to the satellite's angular acceleration, one obtains

$$\dot{\boldsymbol{\omega}} = \mathbf{I}_s^{-1}(-\dot{\boldsymbol{\omega}} \times \mathbf{I}_s \dot{\boldsymbol{\omega}} - \dot{\mathbf{h}} - \boldsymbol{\omega} \times \mathbf{h} + \boldsymbol{\tau}). \quad (3.15)$$

### 3.3 Satellite Kinematics

In this section, the kinematics of the satellite are described using unit quaternions. A unit quaternion is defined as the sum of a scalar and a vector, as given by

$$\bar{\mathbf{q}} = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}, \quad (3.16)$$

where  $q_0$  is also denoted the real part of the quaternion, and

$$\mathbf{q} = q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k} \quad (3.17)$$

contains the imaginary components of the quaternion. In aerospace applications, it is common to use a special quaternion for attitude representation, where the real part is

$$q_0 = \cos\left(\frac{\theta}{2}\right), \quad (3.18)$$

and the vectorial part is

$$\mathbf{q} = \hat{e} \sin\left(\frac{\theta}{2}\right), \quad (3.19)$$

where  $\hat{e}$  designates the normalized axis of rotation and  $\theta$  represents the angle of rotation. To derive the equation that relates the attitude quaternion with the satellite's angular velocity, one has to determine the derivative of  $\bar{\mathbf{q}}$ .

Let  $\bar{\mathbf{q}}(t)$  be the quaternion relative to a reference frame at time  $t$ , and  $\bar{\mathbf{q}}(t + \delta t)$  be the quaternion relative to the same reference frame at time  $t + \delta t$ . The quaternion that brings  $\bar{\mathbf{q}}(t)$  to  $\bar{\mathbf{q}}(t + \delta t)$  is given by

$$\bar{\mathbf{p}}(t) = \cos\left(\frac{\delta\theta}{2}\right) + \hat{e} \sin\left(\frac{\delta\theta}{2}\right). \quad (3.20)$$

For a small time shift ( $\delta t \rightarrow 0$ ), the simplifications

- $\cos\left(\frac{\delta\theta}{2}\right) \rightarrow 1$  and
- $\sin\left(\frac{\delta\theta}{2}\right) \rightarrow \frac{\delta\theta}{2}$

are valid, which yields

$$\bar{\mathbf{p}}(t) \approx 1 + \hat{\mathbf{e}} \frac{\delta\theta}{2}. \quad (3.21)$$

The quaternion  $\bar{\mathbf{q}}(t + \delta t)$  is finally obtained by multiplying  $\bar{\mathbf{q}}(t)$  with  $\bar{\mathbf{p}}(t)$ , as given by

$$\begin{aligned} \bar{\mathbf{q}}(t + \delta t) &= \bar{\mathbf{q}}(t) \otimes \bar{\mathbf{p}}(t) \\ &= \bar{\mathbf{q}}(t) \otimes \left( 1 + \hat{\mathbf{e}} \frac{\delta\theta}{2} \right). \end{aligned} \quad (3.22)$$

Dividing both sides by  $\delta t$  and letting  $\delta t \rightarrow 0$ , one gets

$$\begin{aligned} \frac{\bar{\mathbf{q}}(t + \delta t) - \bar{\mathbf{q}}(t)}{\delta t} &= \bar{\mathbf{q}}(t) \otimes \left( \frac{1}{2} \hat{\mathbf{e}} \frac{\delta\theta}{\delta t} \right) \Leftrightarrow \\ \Leftrightarrow \frac{d\bar{\mathbf{q}}}{dt} &= \bar{\mathbf{q}}(t) \otimes \left( \frac{1}{2} \boldsymbol{\omega}(t) \right), \end{aligned} \quad (3.23)$$

where  $\boldsymbol{\omega}(t) = \hat{\mathbf{e}} \zeta(t)$  and  $\zeta(t) = \lim_{\delta t \rightarrow 0} \frac{\delta\theta}{\delta t}$ . Knowing that the product of two generic quaternions  $\bar{\mathbf{u}}$  and  $\bar{\mathbf{v}}$  is given by

$$\bar{\mathbf{u}} \otimes \bar{\mathbf{v}} = u_0 v_0 - \mathbf{u} \mathbf{v} + u_0 \mathbf{v} + v_0 \mathbf{u} + \mathbf{u} \times \mathbf{v}, \quad (3.24)$$

one can expand (3.23) [25] and obtain

$$\frac{d}{dt} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}, \quad (3.25)$$

which may be written in matrix notation as

$$\dot{\bar{\mathbf{q}}} = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}) \bar{\mathbf{q}}. \quad (3.26)$$

Finally, the kinematics equation can be split into the dynamics of its real and vectorial part, as given by

$$\frac{d}{dt} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 \\ q_0 & 0 & 0 \\ 0 & q_0 & 0 \\ 0 & 0 & q_0 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} \omega_1 & \omega_2 & \omega_3 \\ 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix}, \quad (3.27)$$

or in the compact format

$$\frac{d}{dt} \begin{pmatrix} q_0 \\ \mathbf{q} \end{pmatrix} = \frac{1}{2} q_0 \begin{pmatrix} 0 \\ \mathbf{I}_3 \end{pmatrix} \boldsymbol{\omega} - \frac{1}{2} \begin{pmatrix} \boldsymbol{\omega}^T \\ \mathbf{S}(\boldsymbol{\omega}) \end{pmatrix} \mathbf{q}, \quad (3.28)$$

where

$$\mathbf{S}(\boldsymbol{\omega}) = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \quad (3.29)$$

### 3.4 State-Space Representation

The combined dynamics and kinematic equations derived above result in the nonlinear state-space model

$$\frac{d}{dt} \begin{pmatrix} q_0 \\ \mathbf{q} \\ \boldsymbol{\omega} \\ \mathbf{h} \\ \gamma \\ \dot{\gamma} \end{pmatrix} = \begin{pmatrix} -\frac{1}{2}\boldsymbol{\omega}^T \mathbf{q} \\ \frac{1}{2}q_0 \mathbf{I}_3 \boldsymbol{\omega} - \frac{1}{2}\mathbf{S}(\boldsymbol{\omega})\mathbf{q} \\ \mathbf{I}_s^{-1}(-\dot{\boldsymbol{\omega}} \times \mathbf{I}_s \dot{\boldsymbol{\omega}} - \dot{\mathbf{h}} - \boldsymbol{\omega} \times \mathbf{h} + \boldsymbol{\tau}) \\ \mathbf{J}(\gamma)\dot{\gamma} \\ \dot{\gamma} \\ K(-\dot{\gamma} + \dot{\gamma}_d) \end{pmatrix}, \quad (3.30)$$

where

- the state is  $\mathbf{x} = (q_0, \mathbf{q}, \boldsymbol{\omega}, \mathbf{h}, \gamma, \dot{\gamma})^T$ ,
- the input is  $\mathbf{u} = \dot{\gamma}_d$ , and
- the disturbance is  $\mathbf{d} = \boldsymbol{\tau}$ .

### 3.5 Model Discretization

Given the continuous-time system

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)), \quad (3.31)$$

one can find a discretized state-space representation. The forward Euler method offers a simple and efficient solution. This method relies on a first-order numerical approximation for solving ordinary differential equations. By employing this approach, the discrete-time system takes the form

$$\mathbf{x}(k+1) = f_d(\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}(k), T_s), \quad (3.32)$$

where

$$f_d(\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}(k), T_s) = \mathbf{x}(k) + T_s f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}(k)), \quad (3.33)$$

$k \in \mathbb{N}_0^+$ , and  $T_s$  denotes the sampling time.



## Chapter 4

# Singularity Analysis

In robotics, a singularity is defined as a configuration where at least one degree of freedom is lost. Since the output of the CMG system represents torques, it becomes impossible to produce torque in a given direction under the presence of a singularity. Mathematically, this is evaluated by the rank of the Jacobian. Therefore, there is a singularity whenever

$$\text{rank}(\mathbf{J}(\gamma)) < \min(m, n), \quad (4.1)$$

giving that  $\mathbf{J}(\gamma) \in \mathbb{R}^{m \times n}$ . For non-square matrices where  $m < n$ , this is equivalent to verifying when the determinant of  $\mathbf{J}\mathbf{J}^T$  equals zero, i.e.,

$$\det(\mathbf{J}\mathbf{J}^T) = 0. \quad (4.2)$$

The rank of the Jacobian, for the triangular CMG system, is equal to two in non-singular circumstances. This is the maximum rank it can take, as this matrix has two rows and three columns. When the system meets a singularity, the Jacobian becomes rank deficient. In this scenario, the rank of the Jacobian drops to one, making it impossible to create torque in 2-dimensions. For the roof CMG system, a non-singular Jacobian has rank 3. When the system meets a singularity, the rank of the Jacobian becomes either two, a so-called rank-2 singularity, or one, a so-called rank-1 singularity. In a rank-2 singularity, torques can only be created in 2-dimensions, whereas in a rank-1 singularity, torques can only be produced in 1-dimension. As SGCMGs comprise flywheels that are permanently spinning at a fixed rotation speed, it is impossible to witness rank-0 singularities in SGCMG systems.

### 4.1 Analysis of the 2D System

There are two different types of singularities in the triangular system, as shown in Fig. 4.1. When all CMGs are aligned and spinning in the same direction, there is a so-called 3h singularity. This type of singularity, also called saturation singularity, restrains the creation of torque in the wheel's direction. Indeed, when all the flywheels are spinning in the same direction, the system can no longer provoke momentum increments since it reaches the limit of the momentum envelope. In fact, the whole bound-

ary of the momentum envelope in Fig. 3.4(a) corresponds to a 3h singularity. In these situations, the available torque is maximum in the direction orthogonal to the axes of the flywheels. An example of a 3h singularity, given by the gimbal configuration  $\gamma_s = (0, 2\pi/3, -2\pi/3)^T$ , is provided in Fig. 4.2(a). When all CMGs are aligned, but one is spinning in the opposite direction of the other two, the system undergoes a so-called 1h singularity. This singularity, caused by an anti-parallel alignment, inhibits the creation of torque in the wheel's direction. In this configuration, the system can only produce torque along the orthogonal direction to the axes of the flywheels. An example of this type of singularities is provided in Fig. 4.2(b). The example depicts the singularity  $\gamma_s = (0, 2\pi/3, \pi/3)^T$ .

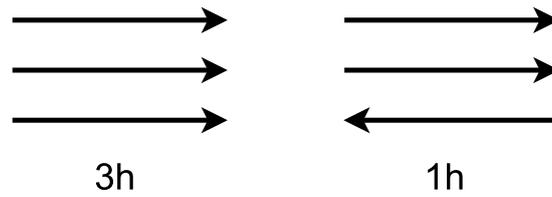


Fig. 4.1: Types of singularities in the triangular array of CMGs.

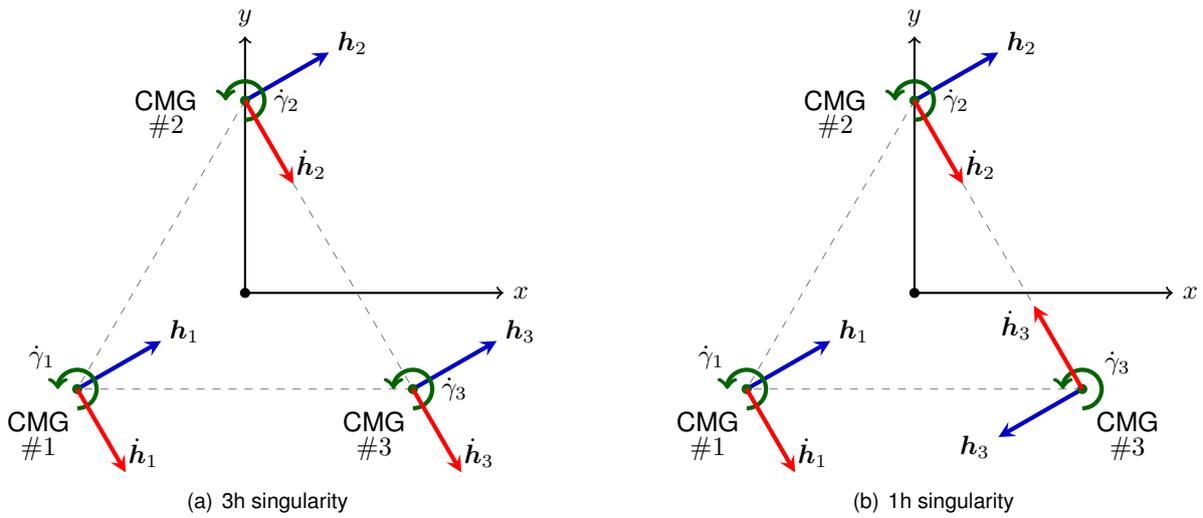


Fig. 4.2: Examples of singularities in the triangular array of CMGs.

As discussed in the beginning of this section, the rank of the Jacobian decreases whenever the system crosses a singularity. The Jacobian is analysed next for both the non-singular and singular states.

For the non-singular gimbal configuration  $\gamma = (0, 0, 0)^T$  the Jacobian takes the form

$$\mathbf{J}(\gamma) = h_0 \begin{pmatrix} -0.5000 & 1.000 & -0.5000 \\ 0.8660 & 0 & -0.8660 \end{pmatrix} \quad (4.3)$$

and is full rank, i.e.,  $\text{rank}(\mathbf{J}(\gamma)) = 2$ . The singular value decomposition for this configuration is given by

the product of the matrices

$$U = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \Sigma = h_0 \begin{pmatrix} 1.225 & 0 & 0 \\ 0 & 1.225 & 0 \end{pmatrix}, \text{ and } V^T = \begin{pmatrix} -0.4082 & 0.8165 & -0.4082 \\ -0.7071 & 0 & 0.7071 \\ 0.5774 & 0.5774 & 0.5774 \end{pmatrix}, \quad (4.4)$$

where both singular values are non-zero and share the same value. The principal directions of the Jacobian are given by the left-singular vectors of  $J$ , i.e., the columns of  $U$ . With this information, an ellipse can be drawn to visualize the conditioning of the Jacobian matrix, as shown in Fig. 4.3(a). The direction of the semiaxes of the ellipse is given by the columns of  $U$ , and the corresponding magnitude coincides with the respective singular values. It is clear that the Jacobian is well conditioned for this gimbal configuration.

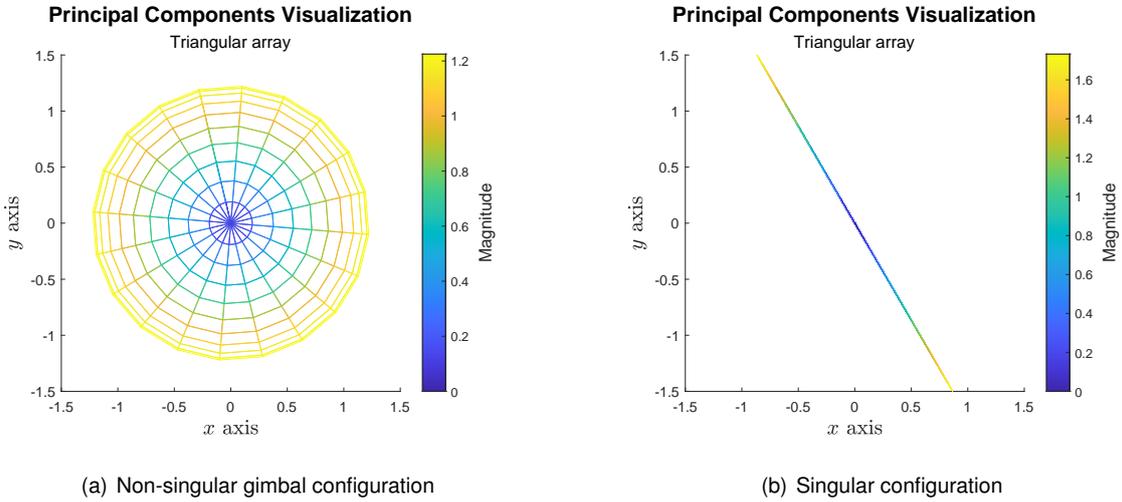


Fig. 4.3: Jacobian's principal components for the different gimbal configurations in the triangular array of CMGs.

For the singular gimbal configuration shown in Fig. 4.2(a), where  $\gamma_s = (0, 2\pi/3, -2\pi/3)^T$ , the Jacobian is given by

$$J(\gamma_s) = h_0 \begin{pmatrix} -0.5000 & -0.5000 & -0.5000 \\ 0.8660 & 0.8660 & 0.8660 \end{pmatrix}, \quad (4.5)$$

which has two linearly dependent rows. The Jacobian has rank 1 and its SVD contains a zero singular value. The SVD matrices, in this case, are given by

$$U = \begin{pmatrix} 0.5000 & -0.8660 \\ -0.8660 & -0.5000 \end{pmatrix}, \Sigma = h_0 \begin{pmatrix} 1.732 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \text{ and } V^T = \begin{pmatrix} -0.5774 & -0.5774 & -0.5774 \\ 0.8165 & -0.4082 & -0.4082 \\ 0 & -0.7071 & 0.7071 \end{pmatrix}. \quad (4.6)$$

The ellipse for this gimbal configuration is reduced to a line segment as one of its semiaxes has size zero (Fig. 4.3(b)). Regardless of the magnitude of the gimbal inputs, the torque produced by the system will be zero along the singular direction  $u = (-0.8660, -0.5000)^T$ . Indeed, if one defines a gimbal input

envelope in  $\mathbb{R}^3$ , where

$$\dot{\gamma}_i \in [-1.5, 1.5] \text{ rad/s}, \quad \forall i \in \{1, 2, 3\}, \quad (4.7)$$

and map it to torques via (3.3), it becomes possible to visualize the torque envelopes for both the non-singular and singular gimbal configurations. These plots are presented in Fig. 4.4 and suggest that torques are possible in every 2D direction for the non-singular gimbal configuration, whereas torques are only feasible along one dimension, i.e. along the non-singular direction, for the singular situation.

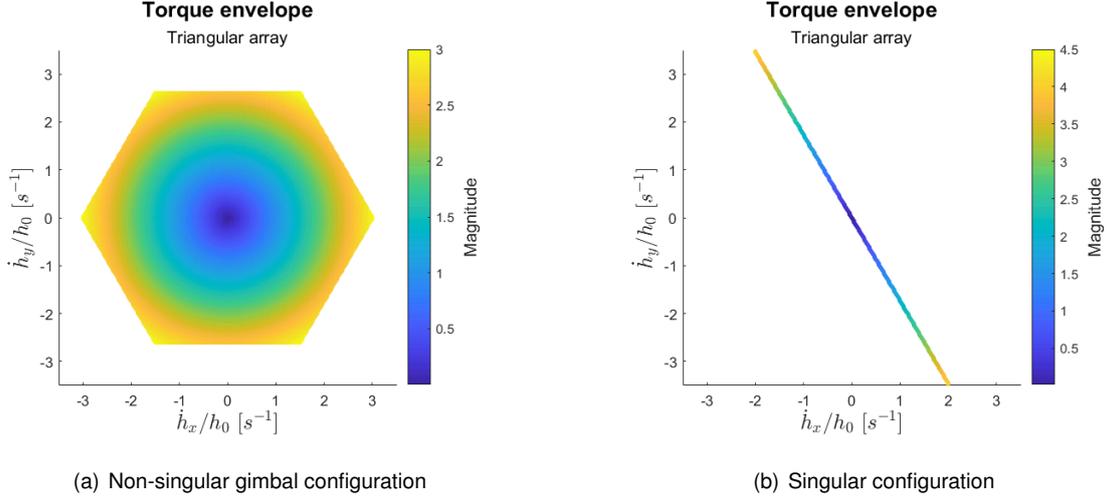


Fig. 4.4: Torque envelope for the different gimbal configurations in the triangular array of CMGs.

To determine the whole singularity envelope for the triangular system, one has to solve (4.2). Since the expression for the determinant may become quite complex to solve analytically, it is easier to break down the determinant of  $\mathbf{J}\mathbf{J}^T$  into the sum of the determinants of its submatrices. Recalling the Cauchy-Binet theorem [41], the determinant of the product between two generic matrices  $\mathbf{A}$  and  $\mathbf{B}$  is given by

$$\det(\mathbf{A}\mathbf{B}) = \sum_{S \in \binom{[n]}{m}} \det(\mathbf{A}_{[m],S}) \det(\mathbf{B}_{S,[m]}). \quad (4.8)$$

Since  $\mathbf{A} = \mathbf{J}$  and  $\mathbf{B} = \mathbf{J}^T$ , the determinant of  $\mathbf{J}\mathbf{J}^T$  simplifies to

$$\det(\mathbf{J}\mathbf{J}^T) = \sum_{i=1}^n d_i^2, \quad (4.9)$$

where  $d_i = \det(\mathbf{J}_i)$  are the Jacobian minors of order  $m$  and  $\mathbf{J}_i = \mathbf{J}$  with the  $i$ -th column removed. Expanding (4.9) for the triangular system, one gets

$$\det(\mathbf{J}\mathbf{J}^T) = d_1^2 + d_2^2 + d_3^2, \quad (4.10)$$

where the computation of each determinant is straightforward, yielding

$$d_1^2 = \frac{h_0^4}{2} \cos(2\gamma_2 - 2\gamma_3 + 2\alpha) + \frac{h_0^4}{2}, \quad (4.11a)$$

$$d_2^2 = \frac{h_0^4}{2} \cos(2\gamma_3 - 2\gamma_1 + 2\alpha) + \frac{h_0^4}{2}, \quad (4.11b)$$

and

$$d_3^2 = \frac{h_0^4}{2} \cos(2\gamma_1 - 2\gamma_2 + 2\alpha) + \frac{h_0^4}{2}. \quad (4.11c)$$

To verify (4.2), every determinant  $d_i$  has to be zero. This requires finding the solution of the system of linear equations

$$\det(\mathbf{J}\mathbf{J}^T) = 0 \Leftrightarrow \begin{cases} 2\gamma_1 - 2\gamma_2 + \frac{\pi}{3} = \pi + 2k_a\pi, & k_a \in \mathbb{Z}, \\ 2\gamma_3 - 2\gamma_1 + \frac{\pi}{3} = \pi + 2k_b\pi, & k_b \in \mathbb{Z}, \\ 2\gamma_2 - 2\gamma_3 + \frac{\pi}{3} = \pi + 2k_c\pi, & k_c \in \mathbb{Z}. \end{cases} \quad (4.12a)$$

$$(4.12b)$$

$$(4.12c)$$

Since there is an extra equation in (4.12) due to the use of a redundant CMG, one only has to solve two of the three equations. Solving (4.12b) and (4.12c) with respect to  $\gamma_1$  and  $\gamma_2$ , respectively, it is possible to achieve the singularity envelope for the triangular system, that is given by

$$\gamma_1 = \gamma_3 - \frac{\pi}{3} + k_\beta\pi, \quad k_\beta \in \mathbb{Z}, \quad (4.13a)$$

and

$$\gamma_2 = \gamma_3 + \frac{\pi}{3} + k_\gamma\pi, \quad k_\gamma \in \mathbb{Z}. \quad (4.13b)$$

Some of these singularity lines represent 1h singularities, while others represent 3h singularities. The plot shown in Fig. 4.5(a) depicts the singularities for the gimbal space where

$$\gamma_i \in [-\pi, \pi] \text{ rad}, \quad \forall i \in \{1, 2, 3\}. \quad (4.14)$$

The singularities in Fig. 4.5(a) were mapped into angular momentum points via (3.2). By doing so, it

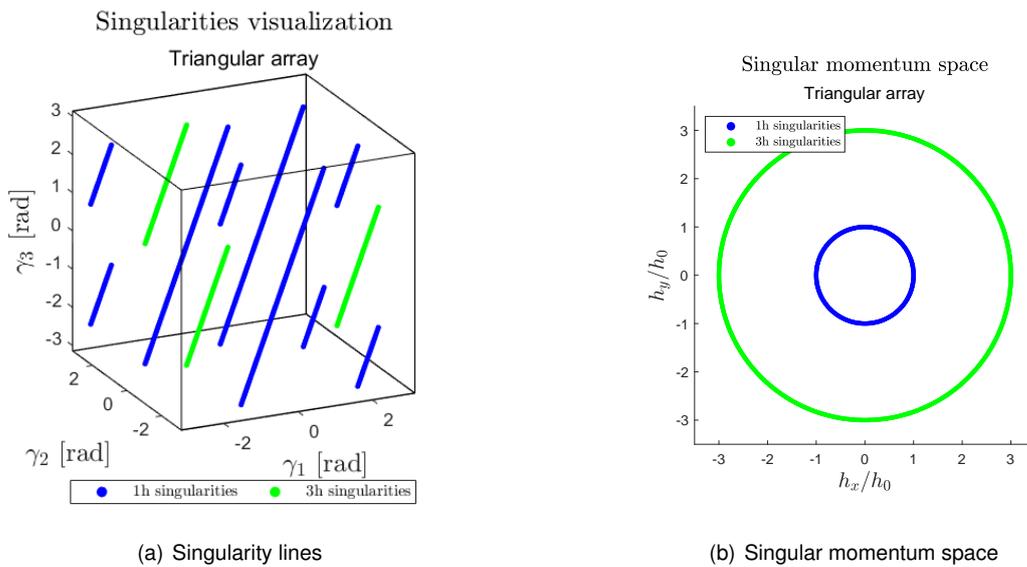


Fig. 4.5: Singularities envelope for the triangular array of CMGs.

was possible to find the singular space in the angular momentum envelope. As shown in Fig. 4.5(b), 3h singularities compose the boundary of the envelope, while 1h singularities are located in a circumference, with magnitude  $h_0$ , inside the envelope. It should be noted, however, that non-singular gimbal configurations can also be mapped into momentum points of magnitude  $h_0$  due to the redundancy of the system.

## 4.2 Analysis of the 3D System

Since there is one extra equation for the 3D system, as well as an additional CMG, the singularities are different. Solving (4.2) for the roof array yields

$$\det(\mathbf{J}\mathbf{J}^T) = d_1^2 + d_2^2 + d_3^2 + d_4^2, \quad (4.15)$$

where the determinants of the submatrices are given by

$$d_1^2 = h_0^6 \sin(\gamma_2)^2 \cos(\gamma_3 - \gamma_4)^2, \quad (4.16a)$$

$$d_2^2 = h_0^6 \cos(\gamma_1)^2 \cos(\gamma_3 - \gamma_4)^2, \quad (4.16b)$$

$$d_3^2 = h_0^6 \cos(\gamma_4)^2 \cos(\gamma_1 - \gamma_2)^2, \quad (4.16c)$$

and

$$d_4^2 = h_0^6 \sin(\gamma_3)^2 \cos(\gamma_1 - \gamma_2)^2. \quad (4.16d)$$

The set of singularities, that results from solving (4.16), is given by the planes

$$\gamma_1 = \frac{\pi}{2} + k_\alpha \pi \quad \wedge \quad \gamma_2 = k_\beta \pi, \quad k_\alpha, k_\beta \in \mathbb{Z}, \quad (4.17a)$$

and

$$\gamma_3 = k_\gamma \pi \quad \wedge \quad \gamma_4 = \frac{\pi}{2} + k_\delta \pi, \quad k_\gamma, k_\delta \in \mathbb{Z}, \quad (4.17b)$$

and the points

$$\gamma_1 = k_\alpha \pi \quad \wedge \quad \gamma_2 = \frac{\pi}{2} + k_\beta \pi \quad \wedge \quad \gamma_3 = \frac{\pi}{2} + k_\gamma \pi \quad \wedge \quad \gamma_4 = k_\delta \pi, \quad k_\alpha, k_\beta, k_\gamma, k_\delta \in \mathbb{Z}. \quad (4.17c)$$

The roof array contains both rank-1 and rank-2 singularities. Rank-2 singularities happen when the system loses one degree of freedom. In these situations, the CMG array is only capable of creating torque in two dimensions. Rank-1 singularities happen when the system loses two degrees of freedom. These cases are more rare than rank-2 singularities, although when they happen, the actuation loss is higher, as the system is only able to produce torque along one direction. Examples of rank-1 singularities are provided in Fig. 4.6. In these examples, no torque is available in the  $xy$  plane due to the alignment of the CMGs. Rank-2 singularities are illustrated in Fig. 4.7. In Fig. 4.7(a), no torque can be generated

along the  $x$  direction, while in Fig. 4.7(b), no torque can be generated along the  $y$  direction. Finally, Fig. 4.7(c) illustrates a singular gimbal configuration where no torque can be created in the  $z$  direction.

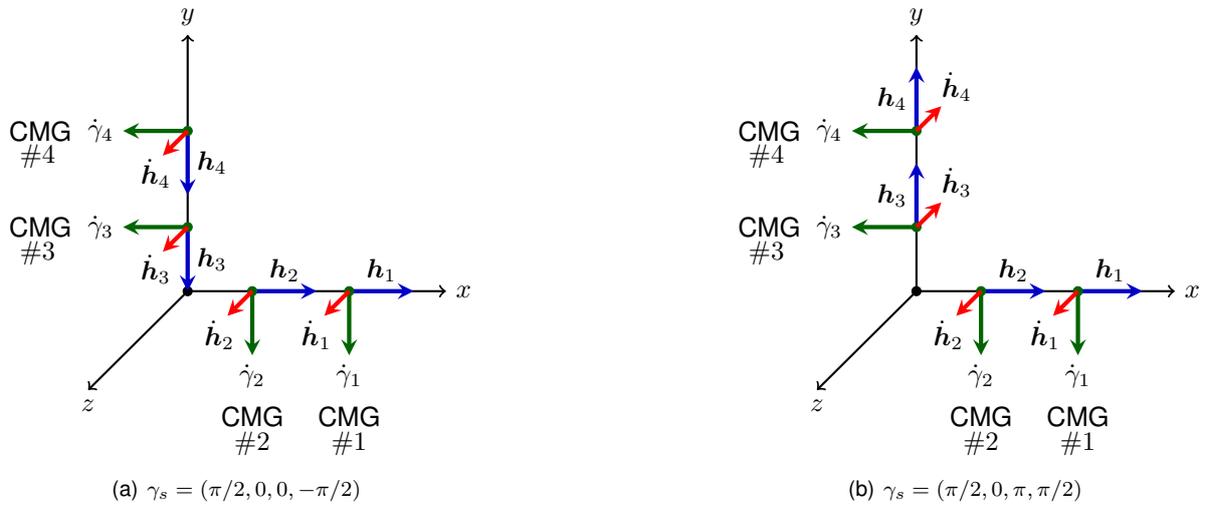


Fig. 4.6: Examples of rank-1 singularities in the roof array of CMGs.

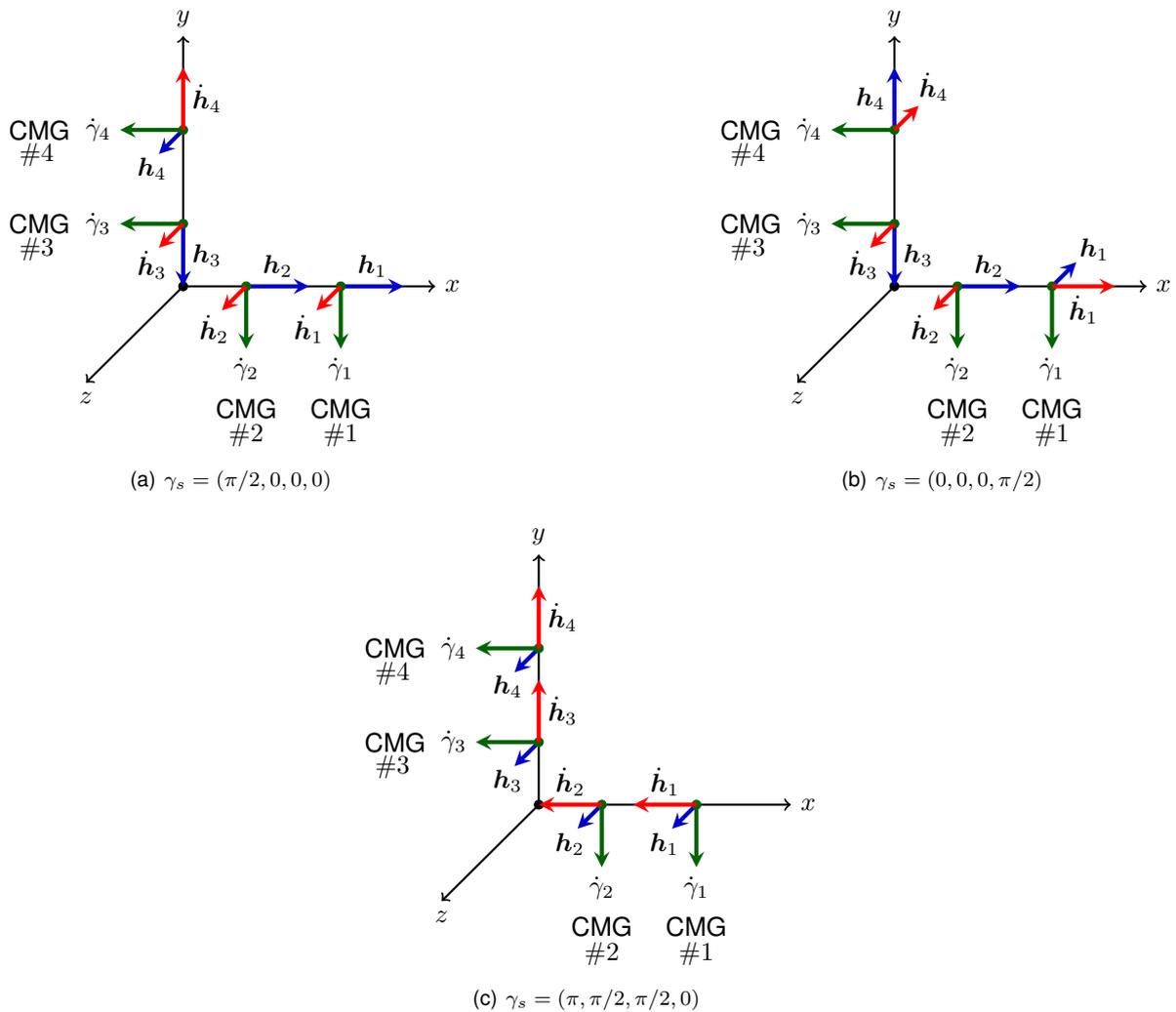


Fig. 4.7: Examples of rank-2 singularities in the roof array of CMGs.

Figs. 4.8-4.10 display both the principal components and the torque envelopes for a non-singular gimbal configuration, a rank-2, and a rank-1 singularity, respectively. To accomplish the torque envelopes for each gimbal configuration, an hypercube was defined in  $\mathbb{R}^4$ , where

$$\dot{\gamma}_i \in [-1.5, 1.5] \text{ rad/s}, \quad \forall i \in \{1, 2, 3, 4\}. \quad (4.18)$$

Finally, since the Jacobian's singular values are all non-zero for the non-singular state, the principal components of the Jacobian exhibit an ellipsoid whose volume is given by

$$V_{\text{ellipsoid}} = \frac{4}{3}\pi\sigma_1\sigma_2\sigma_3, \quad (4.19)$$

where  $\sigma_i$  represents its singular values. For the rank-2 singularity in Fig. 4.9, no torque can be produced along the  $x$  direction. In this situation the ellipsoid is reduced to an ellipse in 2-dimensions, whose volume equals zero due to the existence of a null singular value. For rank-1 singularities, only one direction displays non-zero magnitude. Consequently, the torque plot displays a line segment in  $\mathbb{R}^3$ , as torque can only be created along the non-singular direction.

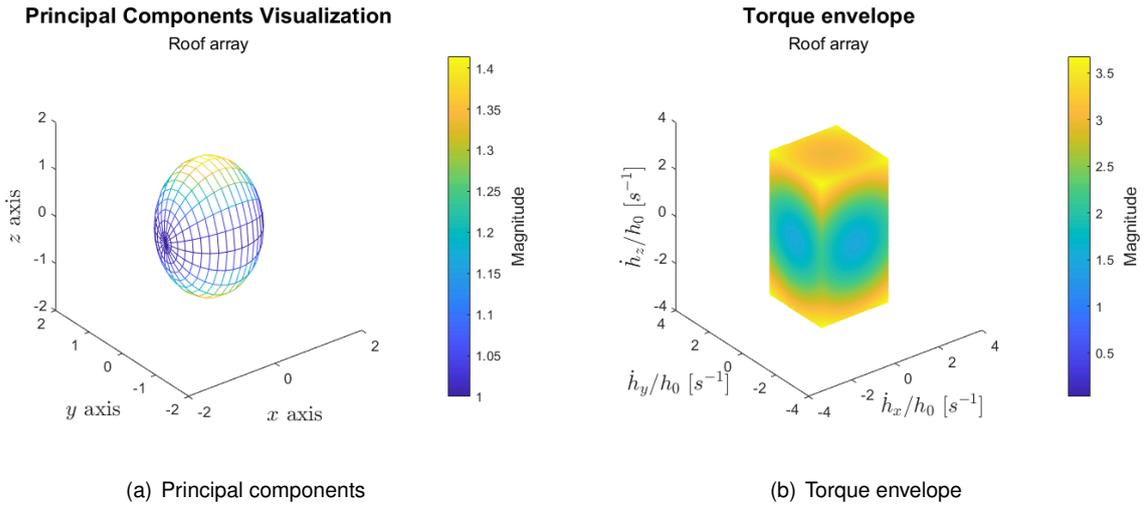


Fig. 4.8: Non-singular gimbal configuration:  $\gamma_s = (0, 0, 0, 0)$ .

### 4.3 Singularity Classification

The singularities of a SGCMG array are organized into classes and sub-classes, as shown in Fig. 4.11. Singularities that can be avoided through the use of null motion are classified as hyperbolic, while singularities that cannot be avoided with the use of null motion are classified as elliptical. Saturation singularities at the limit of the array capability exist in every CMG system. These singularities compose the outer surface of the angular momentum envelope and cannot be avoided through null motion, thus being classified as elliptical. Often referred as external singularities, due to their peripheral positioning in the momentum envelope, escaping these singularities leads to unwanted changes in the spacecraft at-

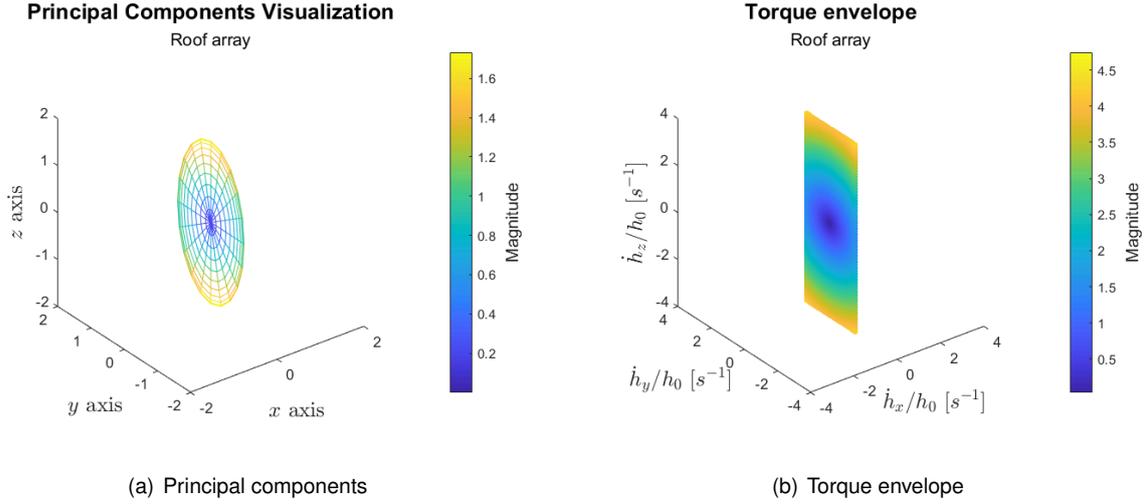


Fig. 4.9: Rank-2 singularity:  $\gamma_s = (\pi/2, 0, 0, 0)$ .

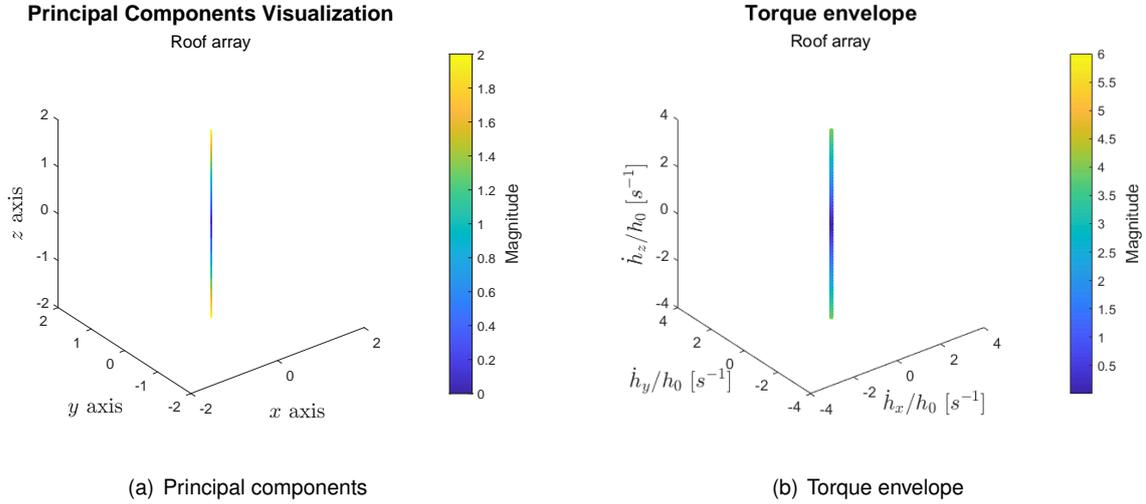


Fig. 4.10: Rank-1 singularity:  $\gamma_s = (\pi/2, 0, 0, -\pi/2)$ .

titude. Singularities that lie inside the momentum envelope are often designated as internal and usually allow for avoidance through the use of null motion. Nonetheless, there are CMG systems that comprise internal singularities inescapable with null motion. Depending on the space for null motion, hyperbolic singularities are subcategorized into degenerate or non-degenerate. Degenerate motion that lies in a space where the rank of the Jacobian is not maximum does not drive the system to a non-singular state. These solutions must be discarded and should be treated similarly to elliptic singularities. Hyperbolic singularities where null motion effectively brings the system to a non-singular state are called non-degenerate [5]. These singularities are passable, as the system can overcome their effect without introducing a torque error in the system. Singularities that are classified differently are called impassable, as for these cases, the introduction of a torque error is necessary to escape the singularity effect. A description of each class of singularities is given in Table 4.1.

The first step to classify a singularity of a SGCMG system is to seek the existence of null motion.

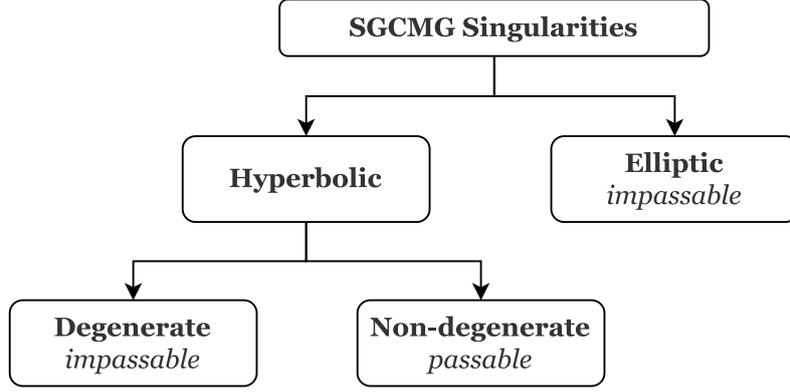


Fig. 4.11: Classification of SGCMG singularities.

Table 4.1: Description of the different singularity classes [5].

<b>Degenerate hyperbolic</b>	<b>Non-degenerate hyperbolic</b>	<b>Elliptic</b>
<ul style="list-style-type: none"> <li>○ Existence of null motion.</li> <li>○ Do not affect the rank of the Jacobian.</li> <li>○ Nullspace displacements cause a transition to a different singular state.</li> <li>○ Null motion solutions lie on a local curve along which the rank of the Jacobian is unchanged.</li> <li>○ Also called impassable.</li> </ul>	<ul style="list-style-type: none"> <li>○ Ability to exploit null motion to escape from the singularity.</li> <li>○ Null motion effectively brings the system to a non-singular state.</li> <li>○ Does not require a change in the array angular momentum, thus not influencing the spacecraft orientation.</li> <li>○ Also called passable.</li> </ul>	<ul style="list-style-type: none"> <li>○ Null motion does not exist for the specific array configuration.</li> <li>○ To escape, it requires a perturbation in the angular momentum, thus changing the spacecraft attitude;</li> <li>○ Also called impassable.</li> </ul>

Mathematically, one starts by performing a Taylor expansion of the angular momentum about the singularity  $\gamma_s$ . This translates into

$$\mathbf{h}(\boldsymbol{\gamma}) = \mathbf{h}(\boldsymbol{\gamma}_s) + \sum_{i=1}^N \left( \left. \frac{\partial \mathbf{h}_i}{\partial \gamma_i} \right|_{\gamma_{i_s}} \delta \gamma_i + \frac{1}{2!} \left. \frac{\partial^2 \mathbf{h}_i}{\partial \gamma_i^2} \right|_{\gamma_{i_s}} \delta \gamma_i^2 + \mathcal{O}(\delta \gamma_i^3) \right), \quad (4.20)$$

where

- $\mathbf{h}(\boldsymbol{\gamma})$  is the angular momentum of the system at a non-singular state in close proximity with the singularity,
- $\mathbf{h}(\boldsymbol{\gamma}_s)$  is the angular momentum of the SGCMG array at the singularity,
- $\mathbf{h}_i$  is the angular momentum of the  $i$ -th CMG at the singular configuration,
- $\delta \gamma_i = \gamma_i - \gamma_{i_s}$  are the null motion gimbal angle displacements,
- $N$  is the number of CMGs, and
- $\mathcal{O}(\delta \gamma_i^3)$  denotes the higher-order terms of the Taylor series.

To facilitate the derivation, define a set of orthonormal axis  $\{\hat{s}_i, \hat{o}_i, \hat{g}_i\}$ , where

- $\hat{s}_i$  represents the spin axis of the flywheel,
- $\hat{o}_i$  represents the torque direction,
- $\hat{g}_i$  represents the gimbal axis direction,

and

$$\hat{o}_i = \hat{s}_i \times \hat{g}_i. \quad (4.21)$$

The first order derivative of  $h_i$  in (4.20) denotes the torque direction of the  $i$ -th CMG as given by

$$\hat{o}_i = \left. \frac{\partial \mathbf{h}_i}{\partial \gamma_i} \right|_{\gamma_{i_s}}. \quad (4.22)$$

The remaining terms are further simplified to

$$\left. \frac{\partial^2 \mathbf{h}_i}{\partial \gamma_i^2} \right|_{\gamma_{i_s}} = \left. \frac{\partial \hat{o}_i}{\partial \gamma_i} \right|_{\gamma_{i_s}} = -\mathbf{h}_i, \quad (4.23)$$

$$\left. \frac{\partial^3 \mathbf{h}_i}{\partial \gamma_i^3} \right|_{\gamma_{i_s}} = - \left. \frac{\partial \mathbf{h}_i}{\partial \gamma_i} \right|_{\gamma_{i_s}} = -\hat{o}_i, \quad (4.24)$$

$$\left. \frac{\partial^4 \mathbf{h}_i}{\partial \gamma_i^4} \right|_{\gamma_{i_s}} = - \left. \frac{\partial \hat{o}_i}{\partial \gamma_i} \right|_{\gamma_{i_s}} = \mathbf{h}_i, \quad (4.25)$$

and can be generalized by

$$\left. \frac{\partial^\alpha \mathbf{h}_i}{\partial \gamma_i^\alpha} \right|_{\gamma_{i_s}} = \hat{o}_i, \quad \alpha = 4k + 1, \quad k \in \mathbb{Z}_0^+, \quad (4.26a)$$

$$\left. \frac{\partial^\alpha \mathbf{h}_i}{\partial \gamma_i^\alpha} \right|_{\gamma_{i_s}} = -\mathbf{h}_i, \quad \alpha = 4k + 2, \quad k \in \mathbb{Z}_0^+, \quad (4.26b)$$

$$\left. \frac{\partial^\alpha \mathbf{h}_i}{\partial \gamma_i^\alpha} \right|_{\gamma_{i_s}} = -\hat{o}_i, \quad \alpha = 4k + 3, \quad k \in \mathbb{Z}_0^+, \quad (4.26c)$$

and

$$\left. \frac{\partial^\alpha \mathbf{h}_i}{\partial \gamma_i^\alpha} \right|_{\gamma_{i_s}} = \mathbf{h}_i, \quad \alpha = 4k + 4, \quad k \in \mathbb{Z}_0^+. \quad (4.26d)$$

Substituting these results in (4.20), follows that

$$\mathbf{h}(\boldsymbol{\gamma}) - \mathbf{h}(\boldsymbol{\gamma}_s) = \sum_{i=1}^n \left( \hat{o}_i \delta \gamma_i - \frac{1}{2!} \mathbf{h}_i \delta \gamma_i^2 - \frac{1}{3!} \hat{o}_i \delta \gamma_i^3 + \frac{1}{4!} \mathbf{h}_i \delta \gamma_i^4 + \dots \right). \quad (4.27)$$

Left multiplying both sides of (4.27) by the singular direction  $\mathbf{u}^T$ , yields

$$\mathbf{u}^T [\mathbf{h}(\boldsymbol{\gamma}) - \mathbf{h}(\boldsymbol{\gamma}_s)] = \sum_{i=1}^n \left( \frac{1}{2!} \mathbf{u}^T \mathbf{h}_i \delta \gamma_i^2 - \frac{1}{4!} \mathbf{u}^T \mathbf{h}_i \delta \gamma_i^4 + \dots \right), \quad (4.28)$$

where the odd order terms vanish because there is no torque along the singular direction. Using the

Taylor expansion of a cosine

$$\cos(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots, \quad (4.29)$$

it is possible to simplify (4.27) to

$$\mathbf{u}^T [\mathbf{h}(\boldsymbol{\gamma}) - \mathbf{h}(\boldsymbol{\gamma}_s)] = \sum_{i=1}^n \mathbf{u}^T \mathbf{h}_i (1 - \cos(\delta\gamma_i)). \quad (4.30)$$

Since the null motion does not affect the total angular momentum of the system, i.e.,

$$\mathbf{h}(\boldsymbol{\gamma}) = \mathbf{h}(\boldsymbol{\gamma}_s), \quad (4.31)$$

the left-hand side of (4.30) equals zero. Consequently, the right-hand side of the equation is also null, yielding

$$\sum_{i=1}^n \mathbf{u}^T \mathbf{h}_i (1 - \cos(\delta\gamma_i)) = 0. \quad (4.32)$$

Therefore, null motion solutions result from finding the gimbal angles  $\gamma_i$  that satisfy (4.32). For simplicity, this equation can be approximated to its  $2^{nd}$  order equivalent, as given by

$$\sum_{i=1}^n \mathbf{u}^T \mathbf{h}_i \delta\gamma_i^2 = 0. \quad (4.33)$$

Designating  $\mathbf{P} = \text{diag}(\mathbf{u}^T \mathbf{h}_i)$  as the singularity projection matrix, (4.33) can be written as

$$(\delta\boldsymbol{\gamma}^T) \mathbf{P} (\delta\boldsymbol{\gamma}) = 0, \quad (4.34)$$

where the difference configuration  $\delta\boldsymbol{\gamma} = \boldsymbol{\gamma} - \boldsymbol{\gamma}_s$  can be described in terms of the nullspace of the Jacobian, i.e.,

$$\delta\boldsymbol{\gamma} = \sum_{i=1}^n c_i \mathbf{v}_i = \mathbf{N} \mathbf{c}, \quad (4.35)$$

where  $\mathbf{c}$  denotes the scaling components of the nullspace vectors. Substituting (4.35) in (4.34) one finally achieves the simplified  $2^{nd}$  order condition for the existence of null motion, that is given by

$$\mathbf{c}^T \underbrace{\mathbf{N}^T \mathbf{P} \mathbf{N}}_S \mathbf{c} = 0, \quad (4.36)$$

where  $S$  is the singularity-definition matrix defined as

$$S = \mathbf{N}^T \mathbf{P} \mathbf{N}. \quad (4.37)$$

The eigenvalues of  $S$  determine whether the singularity is hyperbolic or elliptical. If all the eigenvalues have the same sign (all positive or all negative), only the trivial solution  $\mathbf{c} = \mathbf{0}$  satisfies (4.36). In this situation, where  $S$  is either positive definite or negative definite, null motion does not exist to avoid the

singularity, and therefore the singularity is elliptical. For the scenarios where  $S$  is indefinite or semi-definite, there are non-trivial solutions to (4.36) in which  $c \neq \mathbf{0}$ . These are the so-called hyperbolic singularities.

With all the non-trivial null solutions  $c$  determined in (4.36), one has to perform the degeneracy test to identify the non-degenerate solutions. If any solution meets the constraint

$$d = \sqrt{\det(\mathbf{J}\mathbf{J}^T)} > 0, \quad (4.38)$$

then it passes the degeneracy test. The analysis begins with a Taylor expansion of  $d$  about the singular configuration  $\gamma_s$ , as given by

$$d(\gamma) = d(\gamma_s) + \sum_{i=1}^n \left( \left. \frac{\partial d}{\partial \gamma_i} \right|_{\gamma_s} \delta\gamma_i + \frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n \left. \frac{\partial^2 d}{\partial \gamma_j \partial \gamma_i} \right|_{\gamma_s} \delta\gamma_j \delta\gamma_i + \mathcal{O}(\delta\gamma_i^3) \right). \quad (4.39)$$

By the definition of a singularity, it is known that

- $d(\gamma_s) = 0$ , and
- $\left. \frac{\partial d}{\partial \gamma_i} \right|_{\gamma_s} = 0$  because  $d \geq 0 \forall \gamma_i$  ( $\mathbf{J}\mathbf{J}^T \geq 0$ ).

Similarly to (4.20), the high-order terms can be neglected for simplicity, reducing (4.39) to its 2<sup>nd</sup> order equivalent

$$d(\gamma) = \frac{1}{2} \delta\gamma^T \mathbf{H} \delta\gamma, \quad (4.40)$$

where  $\mathbf{H} = \left. \frac{\partial^2 d}{\partial \gamma \partial \gamma^T} \right|_{\gamma_s}$  is the Hessian matrix evaluated at the singularity. Expressing  $\gamma$  in terms of the nullspace components, one gets

$$d(\gamma) = \frac{1}{2} \mathbf{c}^T \underbrace{\mathbf{N}^T \mathbf{H} \mathbf{N}}_{\mathbf{W}} \mathbf{c}. \quad (4.41)$$

Finally, non-degeneracy is guaranteed if any of the following conditions is met:

- if  $\mathbf{H} \succ 0 \Rightarrow \mathbf{W} \succ 0 \Rightarrow d(\gamma) > 0$ ;
- if  $\mathbf{W} \succ 0 \Rightarrow d(\gamma) > 0$ , regardless of the definiteness of  $\mathbf{H}$ ;
- if  $\mathbf{c}^T \mathbf{W} \mathbf{c} > 0 \Rightarrow d(\gamma) > 0$ , regardless of the definiteness of  $\mathbf{W}$ ;

If  $\mathbf{W}$  is singular, solutions exist to keep the quadratic approximation of the determinant equal to zero. If these solutions are also solutions of (4.36), the test for degeneracy is inconclusive. In this situation, high-order terms should be considered in both tests through a numerical analysis [5].

The singularity  $\gamma_s = (0, 2\pi/3, -2\pi/3)$  in Fig. 4.2(a) is elliptic as its singularity-definition matrix

$$\mathbf{S} = h_0 \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \quad (4.42)$$

is positive definite. On the other hand, the internal singularity  $\gamma_s = (0, 2\pi/3, \pi/3)$  in Fig. 4.2(b), with

singularity-definition matrix given by

$$\mathbf{S} = h_0 \begin{pmatrix} -0.6667 & 0.5773 \\ 0.5773 & 0 \end{pmatrix}, \quad (4.43)$$

whose eigenvalues are  $\lambda_1 = -h_0$  and  $\lambda_2 = 0.3333h_0$ , is hyperbolic. In this case,  $\mathbf{S}$  is indefinite as its eigenvalues have different signs. The degeneracy test yields to

$$\mathbf{W} = h_0 \begin{pmatrix} 4.6667 & -2.3094 \\ -2.3094 & 2 \end{pmatrix}, \quad (4.44)$$

where  $\lambda_1 = 0.6667h_0$  and  $\lambda_2 = 6h_0$ . Therefore,  $d(\gamma) > 0$  since  $\mathbf{W}$  is positive definite, and so, this is the case of a non-degenerate hyperbolic singularity.

## 4.4 Vicinity of a Singularity

The effect of a singularity on the actuation capability of a satellite was both analytically and graphically analyzed in the previous sections. It became clear that when the system crosses a singularity, torque is lost in at least one direction regardless the input given to the gimbals. For real actuators, this is even more problematic as gimbal rates are physically limited. In fact, as the system approaches a singularity, the inputs to the gimbals become larger in order to cope with the torque required by the controller. The system can only match the commanded torque until the point where it is not further possible to create torque by increasing the speed of the gimbals. This situation typically happens in the vicinity of a singularity and should be avoided as torque matching can no longer be verified. Therefore, the problem of avoiding singularities becomes more complex, since now there is the need to avoid not only the singularity, but also to prevent the system from entering its neighborhood. An application of what was described to the triangular system follows next.

Define a line segment  $r$  as

$$\mathbf{r} = \begin{pmatrix} -\frac{\pi}{3} \\ \frac{\pi}{3} \\ 0 \end{pmatrix}^T + k_r \begin{pmatrix} \frac{\pi}{3} \\ -2\frac{\pi}{3} \\ \frac{\pi}{3} \end{pmatrix}, \quad (4.45)$$

where  $k_r \in [-0.5, 1.5]$ , that crosses the singularities

$$\gamma_{s_A} = \begin{pmatrix} -\frac{\pi}{3} \\ \frac{\pi}{3} \\ 0 \end{pmatrix}^T, \quad \text{and} \quad \gamma_{s_B} = \begin{pmatrix} 0 \\ -\frac{\pi}{3} \\ \frac{\pi}{3} \end{pmatrix}^T, \quad (4.46)$$

as observed in Fig. 4.12(a). The distance to the singularities A and B as a function of  $k_r$  is plotted in Fig. 4.12(b). The system crosses the singularity A for  $k_r = 0$  and the singularity B for  $k_r = 1$ . Fig. 4.12(c) illustrates the largest gimbal input, computed with the Moore-Penrose pseudoinverse, as a function of  $k_r$ . These results were achieved for a control torque of  $\dot{\mathbf{h}}_c = (3, 3)^T$  and an angular momentum  $h_0$  of 4 *Nms*. It is possible to conclude that the largest gimbal rate tends to infinity as the system gets closer to the singularities. There is a point in the gimbal space where the largest gimbal rate,  $\|\dot{\gamma}\|_\infty$ , surpasses the

physical limits of the gimbals, denoted by  $\dot{\gamma}_{\max}$ , and the system no longer meets the torques demanded by the controller.

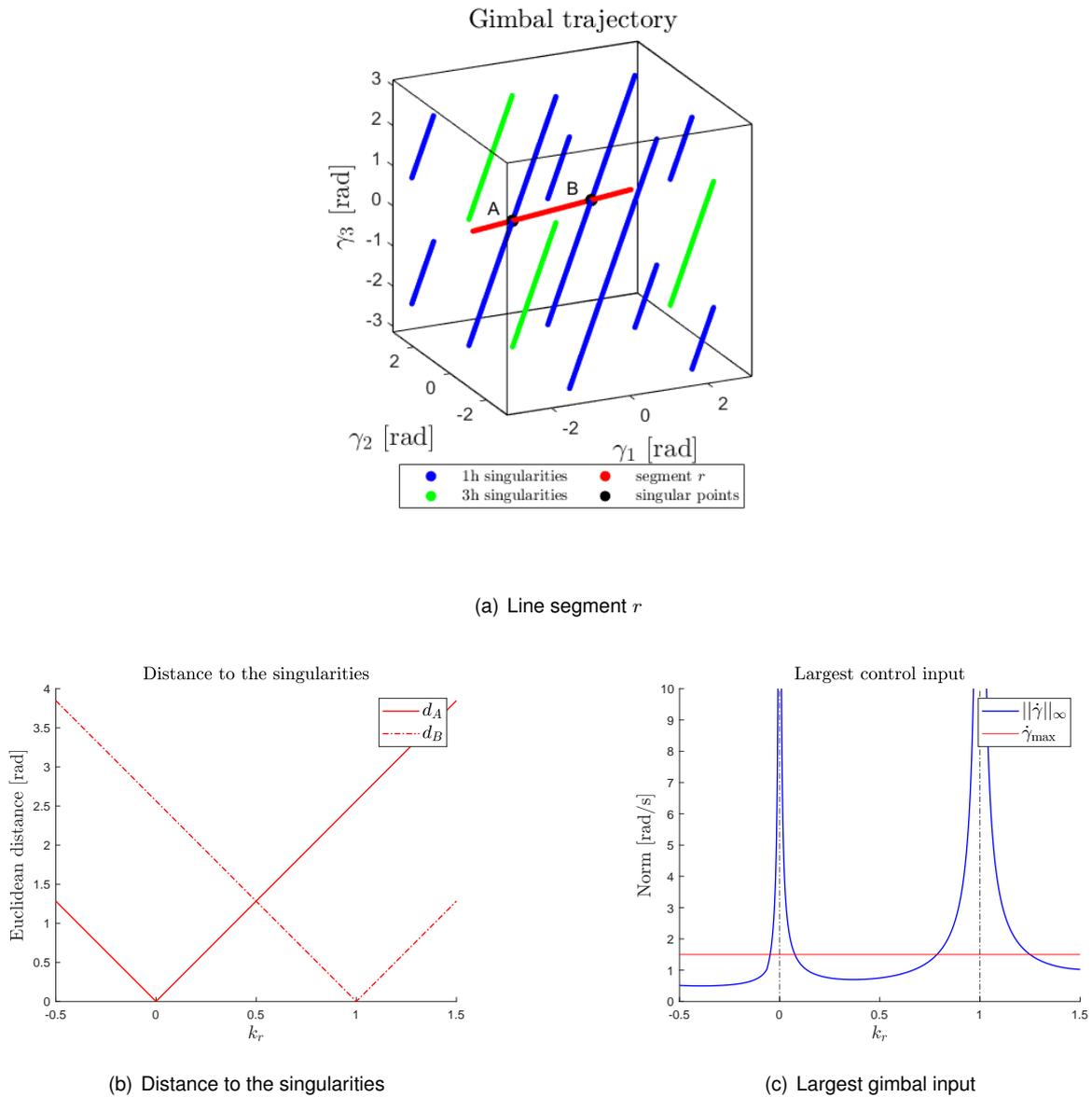


Fig. 4.12: Vicinity of a singularity - a practical example for the triangular system.



## Chapter 5

# Controller Design and Control Allocation

Algorithms for singularity avoidance with application to spacecraft attitude control are proposed throughout this chapter. All of the described approaches resort to the optimal control framework and are included in the ACS as add-on techniques to a well-known literature controller. This way, it is possible to test different CMG allocation algorithms on top of a reliable and conventional controller, and extend their applicability to several control systems. In addition, tractability is preserved as the problem of allocating the actuator inputs is independent from the attitude controller.

Before describing the proposed algorithms, different singularity metrics to evaluate the proximity to a singularity are thoroughly studied in Section 5.1. Herein, a novel singularity function, that uses the Euclidean distance between the system's gimbal configuration and the closest singularity, is derived to overcome the high computational cost associated with standard literature solutions. Then, in Section 5.2, an action governor is synthesized to monitor and adjust the gimbal rates computed with the Moore-Penrose pseudoinverse. If the system enters the vicinity of a singularity, the action governor corrects the inputs fed to the actuator in order to drive the system away from the singularity. When the system is distant from the singularities, the governor directly applies the energy-optimal gimbal rates to the system. An integrated way of computing the optimal gimbal rates is derived in Section 5.3. This method reduces the allocation step into a single optimization problem and displays quite promising results. Finally, in Section 5.4, a MPC-based algorithm is introduced to extend the previous approach over a fixed prediction horizon.

### 5.1 Singularity Metrics

To design a controller for singularity avoidance, the system has to be able to sense the proximity to a singularity. This information is paramount as it allows the system to take control actions that lead to singularity-free gimbal states. As mentioned in Section 4.4, when the system enters the vicinity of a singularity, the gimbal inputs become larger in an effort to meet the desired torque. Indeed, the 2-norm

and the  $\infty$ -norm of the gimbal rate vector can be useful metrics to analyse the closeness of a singular state, as their value increases whenever the system approaches a singularity. However, singularities can be hard to discern in situations where the gimbal inputs are large due to the nature of the control torques. Indeed, if the torques are not properly regulated, they might convert into large gimbal rates that can be mistaken for singularities. Yet, limiting the control torques beyond the physical limitations of the actuators is not usually desired, as by doing so, the actuation capability of the satellite decreases. For these reasons, different metrics to evaluate the proximity to a singularity, in an explicit form, are introduced next.

The most standard singularity metric consists on the evaluation of the determinant of the Jacobian. For rectangular matrices whose number of rows is less than the number of columns, this is accomplished with the manipulability index, that is given by

$$m = \sqrt{\det(\mathbf{J}\mathbf{J}^T)}. \quad (5.1)$$

When the system is approaching a singularity,  $m$  tends to zero. On the other hand, when the system gets further away from singularities,  $m$  increases. This is one of the most intuitive ways of evaluating the proximity of a singularity. However, this approach is computationally expensive since it requires an online computation of the determinant of  $\mathbf{J}\mathbf{J}^T$ .

The SVD of the Jacobian, introduced in Chapter 4, is also a useful tool for singularity avoidance. The singularity index,  $c$ , given by

$$c = \frac{\sigma_{\max}}{\sigma_{\min}}, \quad (5.2)$$

can be used to minimize the ratio between the maximum and minimum singular values of  $\mathbf{J}$ . Also known by condition number, the use of this function promotes the good conditioning of the Jacobian. In singular situations,  $c$  tends to infinity as  $\sigma_{\min}$  tends to zero. Ideally, for a cluster with the same control authority over all axes, the condition number should tend to one.

A different singularity index was proposed by H. Leeghim et al. [18]. In his work, the singularity index

$$v = \frac{1}{2} \sum_{i=j=1, i \neq j}^N \langle \mathbf{f}_i, \mathbf{f}_j \rangle^2 \quad (5.3)$$

was employed to minimize the inner product between the different columns of the Jacobian. In (5.3), the  $i$ -th column of the Jacobian is denoted by  $\mathbf{f}_i$ . In fact, when the system meets a singularity, at least two of these columns become collinear, thus maximizing their inner product. Consequently, the system promotes the orthogonality of the columns of  $\mathbf{J}$  by minimizing  $v$ , thus repelling the system from singularities. When  $v$  is zero, or close to, one may conclude that the rank of the Jacobian is maximum, having no linearly dependent rows. The author also proposed simplifications to approximate this singularity index to the  $2^{nd}$  order. In fact,  $v$ , in its complete form, is nonlinear as it is the result of the product of trigonometric functions.

In this thesis, a novel approach that considers the Euclidean distance to a singularity is proposed. Start by examining the singularities in a 2-dimensional space. Designating the minimum required dis-

tance to a given singularity by  $\kappa$ , one can define a circular exclusion zone centered at the singularity  $\gamma_s$  with radius  $\kappa$ , as depicted in Fig. 5.1. The exclusion requirement is verified whenever the gimbal vector  $\gamma$

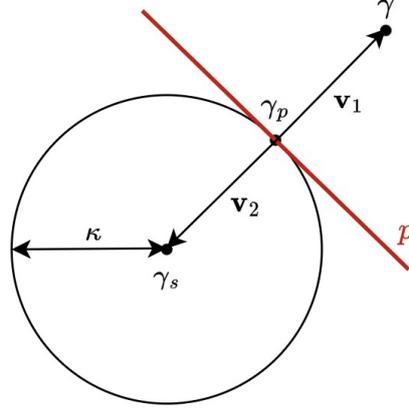


Fig. 5.1: Exclusion requirement for the 2D case.

satisfy

$$\underbrace{(\gamma - \gamma_p)^T}_{v_1^T} \underbrace{(\gamma_s - \gamma_p)}_{v_2} \leq 0, \quad (5.4)$$

where  $\gamma_p$  denotes the projection of the gimbal vector on the exclusion zone boundary, and is calculated by

$$\gamma_p = \gamma_s + \kappa \frac{\gamma - \gamma_s}{\|\gamma - \gamma_s\|_2}. \quad (5.5)$$

These results are applicable to  $n$ -dimensions and can be used for singularity avoidance in many different systems. Before applying (5.4) and (5.5), one has to know, at any given time, the closest singularity in the system. As the singularities are arranged in 3D lines for the triangular system, the closest point on a singularity line consists of the orthogonal projection of  $\gamma$  on the line. This is realized by

$$\gamma_s = \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}(\gamma - \gamma_l) + \gamma_l = \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\gamma + \left(\mathbf{I}_3 - \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right)\gamma_l = \mathbf{A}_v\gamma + \mathbf{b}_v \quad (5.6)$$

where

- $\mathbf{v}$  is a vector director of the singularity line,
- $\gamma_l$  is an arbitrary point on the singularity line,
- $\mathbf{A}_v$  is the projection matrix, and
- $\mathbf{b}_v$  is the offset vector.

Knowing the projection in every singularity line, the closest singularity is the one whose distance to the gimbal is the minimum. Substituting  $\gamma_p$  and  $\gamma_s$  in (5.4) by (5.5) and (5.6), respectively, results in

$$\left(-(\mathbf{I}_3 - \mathbf{A}_v)\gamma + \mathbf{b}_v + \kappa \frac{(\mathbf{I}_3 - \mathbf{A}_v)\gamma - \mathbf{b}_v}{\|(\mathbf{I}_3 - \mathbf{A}_v)\gamma - \mathbf{b}_v\|_2}\right)^T \left(\kappa \frac{(\mathbf{I}_3 - \mathbf{A}_v)\gamma - \mathbf{b}_v}{\|(\mathbf{I}_3 - \mathbf{A}_v)\gamma - \mathbf{b}_v\|_2}\right) \leq 0. \quad (5.7)$$

As for the 3D system, the closest singularity in the singular planes given by (4.17a) and (4.17b) can be

easily obtained by

$$\gamma_s = \left( \frac{\pi}{2} + k_\alpha \pi \quad k_\beta \pi \quad \gamma_3 \quad \gamma_4 \right)^T, \quad k_\alpha, k_\beta \in \mathbb{Z}, \quad (5.8)$$

and

$$\gamma_s = \left( \gamma_1 \quad \gamma_2 \quad k_\gamma \pi \quad \frac{\pi}{2} + k_\delta \pi \right)^T, \quad k_\gamma, k_\delta \in \mathbb{Z}, \quad (5.9)$$

respectively. This is a rather simple metric to assess the proximity of the system to the singularities. In fact, the closest singularity to the system, at any given point, can be obtained by finding the minimum distance between the gimbal angles and all of the lines/planes contained in the singularity envelope. Therefore, instead of storing the information about all of the singularities in the system, one only has to store the equations that describe the singularity lines/planes. Besides this, the number of equations can be minimized by wrapping the gimbal angles to the interval  $[-\pi, \pi]$ . In this way, only the information regarding the singularity lines/planes contained in the interval  $[-\pi, \pi]$  has to be stored. This makes this metric a lightweight and efficient solution that will be later exploited for singularity avoidance.

## 5.2 Action Governor

An AG-based approach is firstly considered to avoid singularities, as shown in Fig. 5.2.

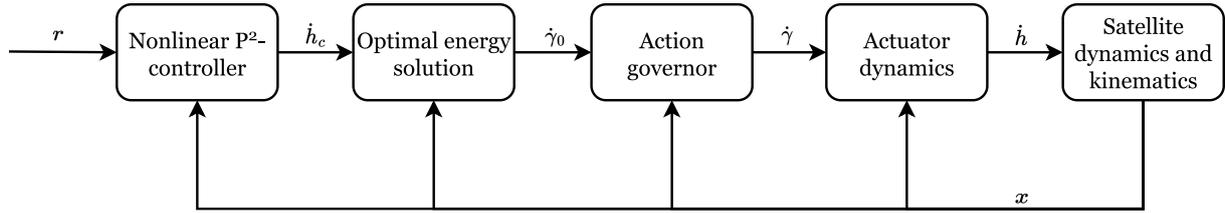


Fig. 5.2: AG-based control diagram for singularity avoidance.

In short:

- A nonlinear proportional square  $P^2$  control algorithm is implemented in the quaternion space. This controller uses the quaternion error and the satellite's angular velocity to compute the required torque,  $\dot{h}_c$ , for a given attitude manoeuvre.
- The nominal set of gimbal rates,  $\dot{\gamma}_0$ , is computed via an energy-driven optimization problem.
- The nominal gimbal rates are subject to an action governor algorithm, that checks and corrects their values for singularity avoidance,  $\dot{\gamma}$ .

The utmost goal of this approach is to promote the matching between the command torque,  $\dot{h}_c$ , and the actual torque produced by the CMG system,  $\dot{h}$ . It is expected that without the use of an algorithm for singularity avoidance, torque matching is not verified in singular situations.

The nonlinear  $P^2$ -controller in its standard form is given by [42]

$$\nu = -P_q q_e - P_\omega \omega, \quad (5.10)$$

where  $\mathbf{q}_e$  denotes the vectorial component of the error quaternion,  $\boldsymbol{\omega}$  represents the angular velocity of the satellite, and  $\mathbf{P}_q$  and  $\mathbf{P}_\omega$  are gain matrices selected, respectively, as [4]

$$\mathbf{P}_q = k_q \mathbf{I}_s, \quad (5.11a)$$

and

$$\mathbf{P}_\omega = k_\omega \mathbf{I}_s, \quad (5.11b)$$

where  $\mathbf{I}_s$  is the inertia matrix of the spacecraft, and  $k_q \in \mathbb{R}$  and  $k_\omega \in \mathbb{R}$  are tuning parameters. Note that this controller uses a reference angular velocity of  $(0, 0, 0)^T$  rad/s, thus only being suitable for constant reference tracking. The error quaternion is calculated with the Hamilton product between the conjugate of the reference quaternion,  $\bar{\mathbf{r}}$ , and the estimated quaternion,  $\bar{\mathbf{q}}$ , as given by [31]

$$\bar{\mathbf{q}}_e = \bar{\mathbf{r}}^* \otimes \bar{\mathbf{q}} = (r_0 - \mathbf{r}) \otimes (q_0 + \mathbf{q}), \quad (5.12)$$

which unfolds into

$$\bar{\mathbf{q}}_e = \begin{pmatrix} r_0 & r_1 & r_2 & r_3 \\ -r_1 & r_0 & r_3 & -r_2 \\ -r_2 & -r_3 & r_0 & r_1 \\ -r_3 & r_2 & -r_1 & r_0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}. \quad (5.13)$$

Only the vectorial part of the error is used in (5.10). If the commanded rotation is more than  $\pi$  radians, the closest rotation is in the opposite direction. This is determined by the sign of  $q_0$ . If  $q_0 < 0$ , the closest rotation is given by the conjugate of  $\bar{\mathbf{q}}_e$  [42]. To account for the physical limitations of a real CMG system, saturation functions are introduced to bound the quaternion error as well as the output torque. Thus, (5.10) becomes

$$\boldsymbol{\nu} = \underset{U}{\text{sat}}(-\mathbf{P}_q \underset{L_i}{\text{sat}}(\mathbf{q}_e) - \mathbf{P}_\omega \boldsymbol{\omega}), \quad (5.14)$$

where  $L_i$  is the error quaternion saturation function given by

$$L_i(\mathbf{q}_e) = \frac{k_\omega}{k_q} \min\{\sqrt{4a_i |q_{e_i}|}, w_{i_{\max}}\}, \quad \forall i \in \{1, 2, 3\}, \quad (5.15)$$

- with  $a_i$  being the maximum acceleration along the  $i$ -axis computed as  $\dot{h}_{\max}/I_{s_{ii}}$ ,
- $|q_{e_i}|$  the modulus of the error quaternion along the  $i$ -axis, and
- $w_{i_{\max}}$  the maximum angular velocity along the  $i$ -axis.

Moreover,  $U$  is the output torque saturation function, given by

$$U(\boldsymbol{\nu}) = \begin{cases} \boldsymbol{\nu} & , \text{if } \|\boldsymbol{\nu}\|_\infty < \dot{h}_{\max}, \\ \dot{h}_{\max} \frac{\boldsymbol{\nu}}{\|\boldsymbol{\nu}\|_\infty} & , \text{if } \|\boldsymbol{\nu}\|_\infty \geq \dot{h}_{\max}. \end{cases} \quad (5.16)$$

Recalling (3.13), the torque expressed in the inertial frame

$$\boldsymbol{\nu} = -\dot{\mathbf{h}} - \boldsymbol{\omega} \times \mathbf{h}, \quad (5.17)$$

is equivalent to

$$\dot{\mathbf{h}}_c = -\boldsymbol{\nu} - \boldsymbol{\omega} \times \mathbf{h}, \quad (5.18)$$

where  $\dot{\mathbf{h}}_c$  is the command torque expressed in the body frame. Finally, replacing  $\boldsymbol{\nu}$  in (5.18) by (5.14), it follows that

$$\dot{\mathbf{h}}_c = -\underset{U}{\text{sat}}\left(-P_q \underset{L_i}{\text{sat}}(\mathbf{q}_e) - P_\omega \boldsymbol{\omega}\right) - \boldsymbol{\omega} \times \mathbf{h}. \quad (5.19)$$

The optimal energy solution for a given control torque is obtained using the Moore-Penrose pseudoinverse, as demonstrated in (1.3). Furthermore, since gimbal rates are physically limited, the energy-optimal gimbal rates,  $\dot{\boldsymbol{\gamma}}_0$ , undergo a saturation function  $S$ , given by

$$S(\dot{\boldsymbol{\gamma}}_0) = \begin{cases} \dot{\boldsymbol{\gamma}}_0 & , \text{ if } \|\dot{\boldsymbol{\gamma}}_0\|_\infty < \dot{\boldsymbol{\gamma}}_{\max}, \\ \dot{\boldsymbol{\gamma}}_{\max} \frac{\dot{\boldsymbol{\gamma}}_0}{\|\dot{\boldsymbol{\gamma}}_0\|_\infty} & , \text{ if } \|\dot{\boldsymbol{\gamma}}_0\|_\infty \geq \dot{\boldsymbol{\gamma}}_{\max}. \end{cases} \quad (5.20)$$

Since the system comprises at least one redundant actuator, it is possible, in general, to meet a control torque with infinite different gimbal inputs. The nominal gimbal input, that corresponds to the minimum control energy, is slightly modified to guarantee that the system does not fall in an unsafe state, or in this case, it is convenient to assure that the system stays far from the singularities. Denoting the set of all  $d \in \mathbb{R}_0^+$  by  $\mathcal{X}$ , a singularity influence region,  $\mathcal{X}_0$ , that the system ought to avoid, can be defined as

$$\mathcal{X}_0 = \{d \in \mathbb{R}_0^+ : d < \kappa\}, \quad (5.21)$$

where  $d$  represents the distance to the closest singularity, and  $\kappa \in \mathbb{R}_0^+$  represents the distance threshold. Knowing this, the action governor in Fig. 5.2 can be formulated as an online constrained optimization problem of the form

$$\underset{\dot{\boldsymbol{\gamma}}(k) \in \mathcal{U}}{\text{minimize}} \quad \|\dot{\boldsymbol{\gamma}}(k) - \dot{\boldsymbol{\gamma}}_0(k)\|_S^2 \quad (5.22a)$$

$$\text{subject to} \quad \boldsymbol{\gamma}(k+1) = \boldsymbol{\gamma}(k) + T_s \dot{\boldsymbol{\gamma}}(k), \quad (5.22b)$$

$$\boldsymbol{\gamma}(k+1) \in \mathcal{X} \setminus \mathcal{X}_0, \quad (5.22c)$$

where  $\mathcal{U}$  is the set of redundant solutions,  $T_s$  is the sampling time, and the function  $\|\cdot\|_S = \sqrt{(\cdot)^T \mathbf{S}(\cdot)}$ , where  $\mathbf{S} \in \mathbb{R}^{n \times n}$  is a positive-definite matrix, is employed to penalize the difference between the nominal gimbal rate command  $\dot{\boldsymbol{\gamma}}_0(k)$  and the modified gimbal rates  $\dot{\boldsymbol{\gamma}}(k)$  to satisfy (5.22c). The inputs to (5.22) are the pre-calculated energy-optimal gimbal rates  $\dot{\boldsymbol{\gamma}}_0(k)$  and the gimbal state  $\boldsymbol{\gamma}(k)$ . For feasibility reasons, (5.22c) can be relaxed by being handled as a soft constraint in the cost function rather than a hard

constraint. By doing so, the action governor takes the form

$$\underset{\dot{\gamma}(k) \in \mathcal{U}}{\text{minimize}} \quad \|\dot{\gamma}(k) - \dot{\gamma}_0(k)\|_S^2 + \rho P_{\mathcal{X}_0}(\gamma(k+1)) \quad (5.23a)$$

$$\text{subject to} \quad \gamma(k+1) = \gamma(k) + T_s \dot{\gamma}(k), \quad (5.23b)$$

where  $P_{\mathcal{X}_0}$  is a penalty function for the set  $\mathcal{X}_0$ , and  $\rho \in \mathbb{R}^+$  is a penalty parameter.  $P_{\mathcal{X}_0}$  is continuous and

$$P_{\mathcal{X}_0}(\gamma) = \begin{cases} 0 & , \text{if } \gamma \in \mathcal{X} \setminus \mathcal{X}_0, \\ > 0 & , \text{if } \gamma \in \mathcal{X}_0. \end{cases} \quad (5.24)$$

Due to the redundancy of the system, the solution to

$$\dot{\mathbf{h}}_c = \mathbf{J} \dot{\gamma} \quad (5.25)$$

is the set

$$\mathcal{S} = \{\dot{\gamma} \in \mathbb{R}^n : \dot{\gamma} = \dot{\gamma}_0 + \mathbf{N} \lambda\}, \quad (5.26)$$

where  $\mathbf{N} \in \mathbb{R}^{n \times (m - \text{rank}(\mathbf{J}))}$  is the normalized nullspace, and  $\lambda \in \mathbb{R}^{m - \text{rank}(\mathbf{J})}$  is the scaling vector. Assuming that the gimbal rates are physically constrained to the interval

$$\mathcal{D} = \{\dot{\gamma}_i \in \mathbb{R} : |\dot{\gamma}_i| \leq \dot{\gamma}_{\max}, \forall i = 1, \dots, N\}, \quad (5.27)$$

the set of feasible solutions to (5.23) is given by

$$\mathcal{U} = \mathcal{S} \cap \mathcal{D}. \quad (5.28)$$

For the triangular CMG system, the equality constraint

$$\|\dot{\gamma}(k) - \mathbf{A}_n \dot{\gamma}(k) - \mathbf{b}_n\|_2 = 0 \quad (5.29)$$

guarantees that  $\dot{\gamma}$  lies inside  $\mathcal{S}$ . In (5.29), the nullspace projection matrix is given by

$$\mathbf{A}_n = \frac{\mathbf{n} \mathbf{n}^T}{\mathbf{n}^T \mathbf{n}}, \quad (5.30)$$

where  $\mathbf{n} \in \mathbb{R}^n$  is the nullspace of the Jacobian, and the nullspace offset vector is given by

$$\mathbf{b}_n = (\mathbf{I}_3 - \mathbf{A}_n) \dot{\gamma}_0(k). \quad (5.31)$$

This ensures that the gimbal vector, after modification, belongs to the set of redundant solutions. Therefore, a change in the gimbal rates, given by the action governor, produces no net torque in the system.

Additionally, the inequality constraint

$$|\dot{\gamma}_i(k)| \leq \dot{\gamma}_{\max}, \forall i = 1, \dots, N \quad (5.32)$$

is added to the optimization problem to limit the gimbals rates to the interval specified in (5.27). Singularity avoidance is sought by keeping the gimbals angles prediction, given by (5.23b), outside of the plane  $p$  (see Fig. 5.1). Consequently, the exclusion requirement is verified if

$$(\gamma(k+1) - \gamma_p(k))^T (\gamma_s(k) - \gamma_p(k)) \leq 0, \quad (5.33)$$

meaning that the boundary plane  $p$ , computed at the time instant  $k$ , is used for singularity avoidance at the time instant  $k+1$ . This allows for a reduction in the number of operations used for singularity avoidance. In fact, the gimbals projection  $\gamma_p$  is obtained through a nonlinear expression. Therefore, to achieve linearity, this one cannot be updated inside the optimization problem. Using the penalty function  $P_{X_0}(\eta) = \max(\eta(k+1), 0)^2$ , where

$$\eta(k+1) = (\gamma(k+1) - \gamma_p(k))^T (\gamma_s(k) - \gamma_p(k)), \quad (5.34)$$

it become possible to penalize any gimbals motion towards the inner side of the plane  $p$ . If the gimbals lie outside of the plane, no penalty is given. Finally, the action governor can be written as

$$\underset{\hat{\gamma}(k)}{\text{minimize}} \quad \|\hat{\gamma}(k) - \dot{\gamma}_0(k)\|_S^2 + \rho \max(\eta(k+1), 0)^2 \quad (5.35a)$$

$$\text{subject to} \quad \gamma(k+1) = \gamma(k) + T_s \hat{\gamma}(k), \quad (5.35b)$$

$$|\dot{\gamma}_i(k)| \leq \dot{\gamma}_{\max}, \forall i = 1, \dots, N, \quad (5.35c)$$

$$\|\hat{\gamma}(k) - \mathbf{A}_n(k) \hat{\gamma}(k) - \mathbf{b}_n(k)\|_2 = 0, \quad (5.35d)$$

$$\eta(k+1) = (\gamma(k+1) - \gamma_p(k))^T (\gamma_s(k) - \gamma_p(k)). \quad (5.35e)$$

### 5.3 Integrated Allocation

The singularity avoidance strategy described in Section 5.2 relies on an add-on algorithm to modify the gimbals inputs computed with the Moore-Penrose pseudoinverse. If the prediction indicates that the gimbals will fall close to a singular state, the gimbals rates are modified in an effort to drive the system away from the singularity region. This is a convenient method, as the on-board computer only needs to solve an optimization problem whenever it senses the proximity of a singularity. However, since linearity is achieved with the use of the Euclidean distance singularity metric, the problem of allocating the gimbals inputs can be reduced into a single optimization problem with fairly low computational burden.

Therefore, the allocation problem may be written as an optimization problem of the form

$$\underset{\dot{\boldsymbol{\gamma}}(k)}{\text{minimize}} \quad \frac{1}{2} \|\dot{\mathbf{h}}(k) - \dot{\mathbf{h}}_c(k)\|_H^2 + \frac{1}{2} \|\dot{\boldsymbol{\gamma}}(k)\|_U^2 + \frac{1}{2} \|\dot{\boldsymbol{\gamma}}(k) - \dot{\boldsymbol{\gamma}}(k-1)\|_M^2 + \frac{1}{2} \rho \max(\eta(k+1), 0)^2 \quad (5.36a)$$

$$\text{subject to} \quad \dot{\mathbf{h}}(k) = \mathbf{J}(k)\dot{\boldsymbol{\gamma}}(k), \quad (5.36b)$$

$$\boldsymbol{\gamma}(k+1) = \boldsymbol{\gamma}(k) + T_s \dot{\boldsymbol{\gamma}}(k), \quad (5.36c)$$

$$|\dot{\gamma}_i(k)| \leq \dot{\gamma}_{\max}, \forall i = 1, \dots, N, \quad (5.36d)$$

$$|\dot{\gamma}_i(k) - \dot{\gamma}_i(k-1)| \leq \Delta \dot{\gamma}_{\max}, \forall i = 1, \dots, N, \quad (5.36e)$$

$$\eta(k+1) = (\boldsymbol{\gamma}(k+1) - \boldsymbol{\gamma}_p(k))^T (\boldsymbol{\gamma}_s(k) - \boldsymbol{\gamma}_p(k)), \quad (5.36f)$$

where the optimization weights are the positive-definite matrices  $\mathbf{H} \in \mathbb{R}^{m \times m}$ ,  $\mathbf{U} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{M} \in \mathbb{R}^{n \times n}$ , and the parameter  $\rho \in \mathbb{R}^+$ . Note that the maximum gimbal rate variation,  $\Delta \dot{\gamma}_{\max}$ , is obtained by multiplying the maximum gimbal acceleration,  $\ddot{\gamma}_{\max}$ , with the sampling time,  $T_s$ . The inputs to (5.36) are the control torques  $\dot{\mathbf{h}}_c$ , the gimbal angles estimation  $\boldsymbol{\gamma}$ , the Jacobian  $\mathbf{J}$ , the closest singularity  $\boldsymbol{\gamma}_s$ , and the singularity projection  $\boldsymbol{\gamma}_p$ , computed at the time instant  $k$ .

Note that:

- The term  $\frac{1}{2} \|\dot{\mathbf{h}}(k) - \dot{\mathbf{h}}_c(k)\|_H^2$  is used to penalize the mismatch between the control torque commands and the torques produced by the CMG system.
- The term  $\frac{1}{2} \|\dot{\boldsymbol{\gamma}}(k)\|_U^2$  is used to minimize the control energy spent.
- The term  $\frac{1}{2} \|\dot{\boldsymbol{\gamma}}(k) - \dot{\boldsymbol{\gamma}}(k-1)\|_M^2$  is another control energy metric. Yet, this term is used to penalize significant variations between consecutive gimbal inputs.
- Finally, the term  $\frac{1}{2} \rho \max(\eta(k+1), 0)^2$  is used for singularity avoidance.

When the system is outside of the singularity region, (5.36) fosters torque matching through minimal actuation effort. When the system enters the singularity region, the term  $\frac{1}{2} \rho \max(\eta(k+1), 0)^2$  becomes non-null, and the system compensates by using a non-minimal energy solution contained in the set of redundant solutions that satisfy (5.36b). In this way, the system repels the singularities without promoting torque violations. To achieve optimal performance, the optimization weights have to be carefully chosen. As a criterion for parameter tuning, which was verified empirically, the torque mismatch matrix  $\mathbf{H}$  should be at least 10 times larger than the control energy weight matrices  $\mathbf{U}$  and  $\mathbf{M}$ , and at least 10 times smaller than the singularity parameter  $\rho$ .

Lastly, (5.36) is a convex optimization problem. This means that any solution is globally optimal and can be achieved at a very low computational expense. Indeed, (5.36a) is the sum of four convex functions, where the first three are quadratic whose Hessian matrices are positive-definite, and the last results from the composition of the affine map (5.36f) with the squared ReLU function ( $f(x) = \max(x, 0)^2$ ), that is known to be convex. Besides this, the equality constraints (5.36b) and (5.36c) are affine functions of  $\dot{\boldsymbol{\gamma}}(k)$ , and the inequality constraints (5.36d) and (5.36e) result from the composition of an affine map with the modulus function, that is also known to be convex.

## 5.4 MPC-based Allocation

The previous approaches rely on the numerical integration of the gimbal rates to obtain a one-step ahead prediction of the gimbal angles. With this information, the system takes control actions that aim to match the command torques while keeping the gimbals away from any singularity. These methods, however, are not effective in situations where it is impossible to avoid a singularity by computing the gimbal rates one-step ahead in time. Besides this, the solutions provided by these approaches are commonly jerky, as the system is required to act immediately. Indeed, in some situations the system has to be able to sense the proximity of the singularity far-ahead in time in order to avoid it. This also allows for a reduction in the control energy used for dodging the singularity. A new approach based on the model predictive control framework is derived to overcome these limitations. This approach tackles the allocation problem by extending the prediction of the gimbal angles over a fixed prediction window. This provides the ability of anticipating the proximity of a singularity region a few steps ahead in time, and consequently take control actions that lead to singularity-free configurations. To achieve this, a predictor is firstly derived to estimate the sequence of control torques,  $\dot{\mathbf{H}}_c$ , over the prediction window. The predictor uses the reduced system model

$$\frac{d}{dt} \begin{pmatrix} q_0 \\ \mathbf{q} \\ \boldsymbol{\omega} \\ \mathbf{h} \end{pmatrix} = \begin{pmatrix} -\frac{1}{2}\boldsymbol{\omega}^T \mathbf{q} \\ \frac{1}{2}q_0 \mathbf{I}_3 \boldsymbol{\omega} - \frac{1}{2}\mathbf{S}(\boldsymbol{\omega})\mathbf{q} \\ \mathbf{I}_s^{-1}(-\dot{\boldsymbol{\omega}} \times \mathbf{I}_s \dot{\boldsymbol{\omega}} - \dot{\mathbf{h}} - \boldsymbol{\omega} \times \mathbf{h} + \boldsymbol{\tau}) \\ \dot{\mathbf{h}}_c \end{pmatrix}, \quad (5.37)$$

where

- the state is  $\mathbf{x} = (q_0, \mathbf{q}, \boldsymbol{\omega}, \mathbf{h})^T$ ,
- the input is  $\mathbf{u} = \dot{\mathbf{h}}_c$ , and
- the disturbance is  $\mathbf{d} = \boldsymbol{\tau}$ .

The control diagram in Fig. 5.3 illustrates the integration of the predictor on the system. The predictor receives a sequence of attitude references  $\mathbf{R}(k) = (\mathbf{r}(k), \dots, \mathbf{r}(k+N_p))$ , where  $N_p$  denotes the prediction horizon, and combines the system model (5.37) with the controller to calculate the sequence of control torques, represented by  $\dot{\mathbf{H}}_c(k) = (\dot{\mathbf{h}}_c(k), \dots, \dot{\mathbf{h}}_c(k+N_p))$ , that are fed to the allocation MPC. Note that the predictor disregards the gimbals, thus assuming that the control torques are effectively applied to the system. This is a reasonable assumption since torque matching is verified whilst the system is navigating through non-singular states.

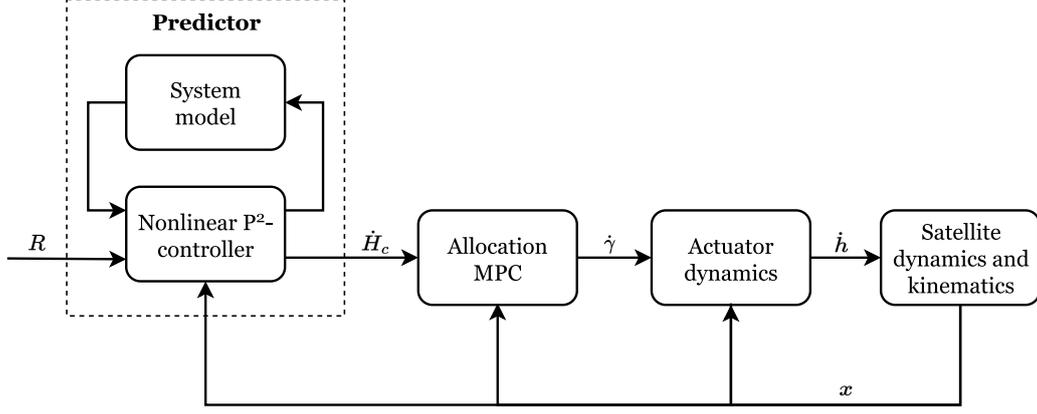


Fig. 5.3: MPC-based control diagram for singularity avoidance.

The MPC-based allocation approach aims to find the optimal sequence of gimbal rates  $\dot{\Gamma}(k) = (\dot{\gamma}(k), \dots, \dot{\gamma}(k + N_p))$  by solving

$$\underset{\dot{\Gamma}(k)}{\text{minimize}} \quad \sum_{\tau=k}^{k+N_p} \left( \frac{1}{2} \|\dot{\mathbf{h}}(\tau) - \dot{\mathbf{h}}_c(\tau)\|_H^2 + \frac{1}{2} \|\dot{\gamma}(\tau)\|_U^2 + \frac{1}{2} \|\dot{\gamma}(\tau) - \dot{\gamma}(\tau-1)\|_M^2 + \frac{1}{2} \rho \max(\eta(\tau+1), 0)^2 \right) \quad (5.38a)$$

$$\text{subject to} \quad \dot{\mathbf{h}}(\tau) = \mathbf{J}(\tau)\dot{\gamma}(\tau), \forall \tau = k, \dots, k + N_p, \quad (5.38b)$$

$$\gamma(\tau+1) = \gamma(\tau) + T_s \dot{\gamma}(\tau), \forall \tau = k, \dots, k + N_p, \quad (5.38c)$$

$$|\dot{\gamma}_i(\tau)| \leq \dot{\gamma}_{\max}, \forall i = 1, \dots, N, \forall \tau = k, \dots, k + N_p, \quad (5.38d)$$

$$|\dot{\gamma}_i(\tau) - \dot{\gamma}_i(\tau-1)| \leq \Delta \dot{\gamma}_{\max}, \forall i = 1, \dots, N, \forall \tau = k, \dots, k + N_p, \quad (5.38e)$$

$$\gamma_s = \gamma_s(\gamma(\tau)), \quad (5.38f)$$

$$\gamma_p(\tau) = \gamma_s + \kappa \frac{\gamma(\tau) - \gamma_s}{\|\gamma(\tau) - \gamma_s\|_2}, \quad (5.38g)$$

$$\eta(\tau+1) = (\gamma(\tau+1) - \gamma_p(\tau))^T (\gamma_s - \gamma_p(\tau)), \forall \tau = k, \dots, k + N_p. \quad (5.38h)$$

This optimization problem is a nonlinear programming problem (NLP). In fact, the Jacobian in (5.38b) has to be dynamically updated with the gimbal angles prediction. Besides this, the problem of finding the closest singularity, given by (5.38f), is a complex nonlinear function of  $\gamma$ . Consequently, (5.38g) and (5.38h) are also nonlinear. However, these constraints can be relaxed through convexification techniques.

To begin with, the linearization of (5.38b) about an equilibrium  $(\gamma_\phi, \dot{\gamma}_\phi)$  can be written as

$$\dot{\mathbf{h}}(\gamma, \dot{\gamma}) = \dot{\mathbf{h}}(\gamma_\phi, \dot{\gamma}_\phi) + \left. \frac{\partial \dot{\mathbf{h}}}{\partial \gamma} \right|_{(\gamma_\phi, \dot{\gamma}_\phi)} (\gamma - \gamma_\phi) + \left. \frac{\partial \dot{\mathbf{h}}}{\partial \dot{\gamma}} \right|_{(\gamma_\phi, \dot{\gamma}_\phi)} (\dot{\gamma} - \dot{\gamma}_\phi), \quad (5.39)$$

which expands to

$$\begin{aligned} \dot{\mathbf{h}}(\gamma, \dot{\gamma}) &= \mathbf{J}(\gamma_\phi)\dot{\gamma}_\phi + \mathbf{A}(\gamma_\phi, \dot{\gamma}_\phi)(\gamma - \gamma_\phi) + \mathbf{J}(\gamma_\phi)(\dot{\gamma} - \dot{\gamma}_\phi) \\ \Leftrightarrow \dot{\mathbf{h}}(\gamma, \dot{\gamma}) &= \mathbf{J}(\gamma_\phi)\dot{\gamma} + \mathbf{A}(\gamma_\phi, \dot{\gamma}_\phi)(\gamma - \gamma_\phi), \end{aligned} \quad (5.40)$$

where  $A$  is given by

$$\mathbf{A}(\boldsymbol{\gamma}_\phi, \dot{\boldsymbol{\gamma}}_\phi) = h_0 \begin{pmatrix} -\dot{\gamma}_1 \cos(\alpha + \gamma_1) & -\dot{\gamma}_2 \sin(\gamma_2) & \dot{\gamma}_3 \cos(\alpha - \gamma_3) \\ -\dot{\gamma}_1 \sin(\alpha + \gamma_1) & \dot{\gamma}_2 \cos(\gamma_2) & -\dot{\gamma}_3 \sin(\alpha - \gamma_3) \end{pmatrix}. \quad (5.41)$$

The linearization matrices  $J$  and  $A$  may be constant or variable along the horizon. In the first case, a single equilibrium is employed to find the linearization matrices, whereas multiple equilibria are used for the second case. Since the MPC computes the optimal states and inputs along the horizon at any time instant, this information can be exploited to find the linearized trajectory in the next time instant. Therefore, the equilibria used for the torque linearization at the time instant  $k$ , are defined by the sequence of gimbal states,  $\boldsymbol{\Gamma}$ , and the sequence of gimbal rates,  $\dot{\boldsymbol{\Gamma}}$ , computed at the time instant  $k-1$ . In addition, the equilibrium corresponding to the last prediction in the horizon, i.e.  $(\boldsymbol{\gamma}_\phi(k+N_p), \dot{\boldsymbol{\gamma}}_\phi(k+N_p))$ , is obtained by duplicating the last set of gimbal angles and gimbal rates in  $\boldsymbol{\Gamma}(k-1)$  and  $\dot{\boldsymbol{\Gamma}}(k-1)$ , respectively. The robustness and the impact of the linearization are thoroughly analyzed in Section 6.2.3.

Furthermore, the computation of the closest singularity line is only performed once, given the state of the gimbals at the time instant  $k$ . Consequently, the exclusion plane  $p$  is defined at the time instant  $k$  and kept constant along the horizon. Finally, the convex representation of the MPC-based allocation algorithm is given by

$$\underset{\dot{\boldsymbol{\Gamma}}(k)}{\text{minimize}} \quad \sum_{\tau=k}^{k+N_p} \left( \frac{1}{2} \|\dot{\mathbf{h}}(\tau) - \dot{\mathbf{h}}_c(\tau)\|_H^2 + \frac{1}{2} \|\dot{\boldsymbol{\gamma}}(\tau)\|_U^2 + \frac{1}{2} \|\dot{\boldsymbol{\gamma}}(\tau) - \dot{\boldsymbol{\gamma}}(\tau-1)\|_M^2 + \frac{1}{2} \rho \max(\eta(\tau+1), 0)^2 \right) \quad (5.42a)$$

$$\text{subject to} \quad \dot{\mathbf{h}}(\tau) = \mathbf{J}(\boldsymbol{\gamma}_\phi(\tau))\dot{\boldsymbol{\gamma}}(\tau) + \mathbf{A}(\boldsymbol{\gamma}_\phi(\tau), \dot{\boldsymbol{\gamma}}_\phi(\tau))(\boldsymbol{\gamma}(\tau) - \boldsymbol{\gamma}_\phi(\tau)), \forall \tau = k, \dots, k+N_p, \quad (5.42b)$$

$$\boldsymbol{\gamma}(\tau+1) = \boldsymbol{\gamma}(\tau) + T_s \dot{\boldsymbol{\gamma}}(\tau), \forall \tau = k, \dots, k+N_p, \quad (5.42c)$$

$$|\dot{\gamma}_i(\tau)| \leq \dot{\gamma}_{\max}, \forall i = 1, \dots, N, \forall \tau = k, \dots, k+N_p, \quad (5.42d)$$

$$|\dot{\gamma}_i(\tau) - \dot{\gamma}_i(\tau-1)| \leq \Delta \dot{\gamma}_{\max}, \forall i = 1, \dots, N, \forall \tau = k, \dots, k+N_p, \quad (5.42e)$$

$$\boldsymbol{\gamma}_s = \boldsymbol{\gamma}_s(\boldsymbol{\gamma}(k)), \quad (5.42f)$$

$$\boldsymbol{\gamma}_p = \boldsymbol{\gamma}_p(\boldsymbol{\gamma}(k)), \quad (5.42g)$$

$$\eta(\tau+1) = (\boldsymbol{\gamma}(\tau+1) - \boldsymbol{\gamma}_p)^T (\boldsymbol{\gamma}_s - \boldsymbol{\gamma}_p), \forall \tau = k, \dots, k+N_p. \quad (5.42h)$$

The operations that led to the linearization of the torque and the use of a constant exclusion zone along the horizon, allowed for the synthesis of a convex optimization problem. Therefore, the global optimal gimbal rates can be achieved at a low computational expense. The linearization of the torque was verified empirically, and displays reasonable results for small prediction horizons. Furthermore, the use of a constant exclusion plane constrains the gimbal motion towards the closest singularity at the time instant  $k$ . This is a valid assumption as the closest singularity to the system, at a given time step, does not typically vary significantly from the previous time step. This is due to the geometry of a CMG cluster, whose singularities are spatially close to each other. Therefore, by preventing the gimbal motion towards one singularity, the system is also preventing the gimbal motion towards its neighbouring singularities.

# Chapter 6

## Simulation Results

In this chapter, the controller and control allocation algorithms proposed in Chapter 5 are validated in the MATLAB® environment. For visualization purposes, the algorithms are tested on a satellite model that comprises a triangular CMG system for steering in the  $xy$  plane, and a reaction wheel for steering in the  $z$  direction.

Firstly, in Section 6.1 the action governor is tested for different parameters and compared to the benchmark solution provided by the Moore-Penrose pseudoinverse. Then, in Section 6.2 the MPC-based allocation algorithm is analysed for different horizons and optimization weights. The effectiveness of the singularity term on the MPC formulation is also confirmed. In addition, the execution times, for the different horizons and formulations, are presented and compared.

The satellite model is depicted in Fig. 3.1<sup>1</sup> and its parameters are given in Table 6.1. The gains for the quaternion error and angular velocity feedback were defined as  $k_q = 0.5$  and  $k_\omega = 0.8$ , respectively.

Table 6.1: Model parameters used in simulation.

Platform	Actuators	
$m = 180 \text{ kg}$ $I_{xx} = 38.07 \text{ kg m}^2$ $I_{yy} = 38.07 \text{ kg m}^2$ $I_{zz} = 32.94 \text{ kg m}^2$ $I_{xy} = -7.730 \text{ kg m}^2$ $\omega_{\max} = 10^\circ / s$	<b>3 × CMG 4-6S Airbus:</b> <ul style="list-style-type: none"> <li>• <math>h_0 = 4 \text{ Nm s}</math></li> <li>• <math>\dot{h}_{\max} = 6 \text{ Nm}</math></li> <li>• <math>\dot{\gamma}_{\max} = 1.5 \text{ rad/s}</math></li> <li>• <math>\ddot{\gamma}_{\max} = 1.5 \text{ rad/s}^2</math></li> </ul>	<b>1 × OCE-RW1000:</b> <ul style="list-style-type: none"> <li>• <math>h_{\omega_{\max}} = 11 \text{ Nm}</math></li> <li>• <math>\dot{h}_{\omega_{\max}} = 1 \text{ Nm}</math></li> <li>• <math>I_\omega = 0.05678 \text{ kg m}^2</math></li> <li>• <math>\omega_{\omega_{\max}} = 1200 \text{ rpm}</math></li> </ul>

The NLP solver *fmincon* was used for solving the allocation problems. This function relies on an iterative method, referred to as sequential quadratic programming, that solves a sequence of optimization subproblems, where each optimizes a quadratic model of the cost function subject to a linearization of the constraints.

The simulations were carried out for 90 s with a sampling time of 0.1 s. For the simulations using the

<sup>1</sup>Image taken from <https://www.satcatalog.com/component/s-200-scaled-microsatellite-platform/>.

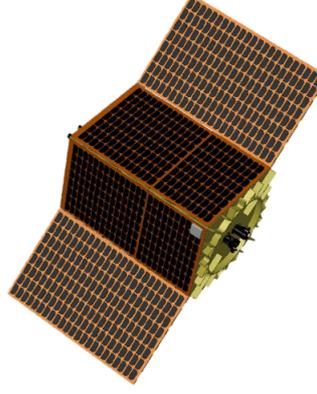


Fig. 6.1: Satellite model.

triangular CMG system, an attitude trajectory consisting of constant references was defined as

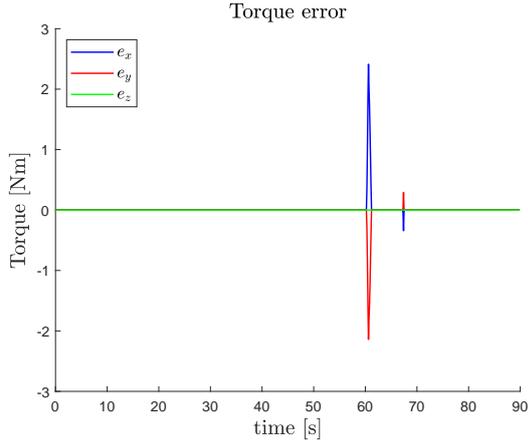
$$(\phi, \theta, \psi) = \begin{cases} (0, 0, 0) \text{ [rad]} & , 0 \leq t \leq 30, \\ (-25, 20, 0) \frac{\pi}{180} \text{ [rad]} & , 30 < t \leq 60, \\ (0, 0, 0) \text{ [rad]} & , 60 < t \leq 90. \end{cases} \quad (6.1)$$

The attitude references were converted to a quaternion representation as follows [43]:

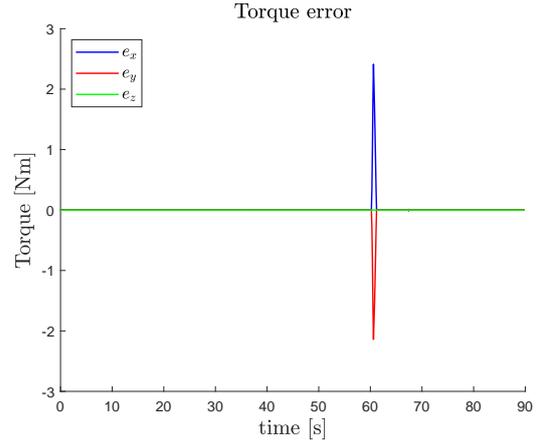
$$\bar{q}_r = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \cos(\phi/2) \cos(\theta/2) \cos(\psi/2) + \sin(\phi/2) \sin(\theta/2) \sin(\psi/2) \\ \sin(\phi/2) \cos(\theta/2) \cos(\psi/2) - \cos(\phi/2) \sin(\theta/2) \sin(\psi/2) \\ \cos(\phi/2) \sin(\theta/2) \cos(\psi/2) + \sin(\phi/2) \cos(\theta/2) \sin(\psi/2) \\ \cos(\phi/2) \cos(\theta/2) \sin(\psi/2) - \sin(\phi/2) \sin(\theta/2) \cos(\psi/2) \end{pmatrix}. \quad (6.2)$$

## 6.1 Action Governor Results

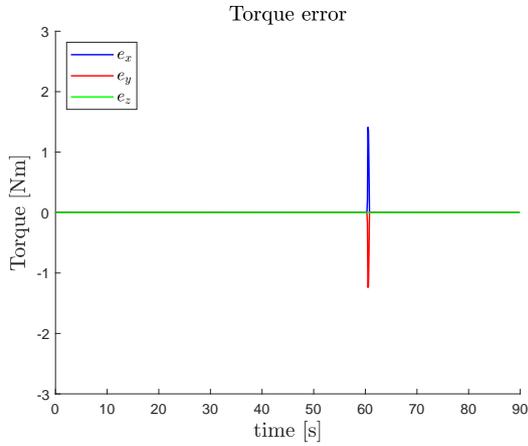
The action governor described by (5.35) was tested for different singularity parameters. In the following simulations, the optimization weights  $S = \text{diag}(1, 1, 1)$  and  $\rho = 10^3$ , were used. The performance of the AG was then evaluated for several  $\kappa$ . The initial orientation of the satellite was set to  $(\phi, \theta, \psi) = (10^\circ, -20^\circ, 0)$  and the gimbals were initialized at  $\gamma = (\pi/4, \pi/3, 0)^T$ . Fig. 6.2 shows the difference between the control torques and the torques produced by the CMG system. Without the use of the AG (Fig. 6.2(a)), it seems clear that the system approaches a singularity at  $t \approx 61$  s and  $t \approx 68$  s. When the minimum required distance to the singularity increases up until  $\kappa = 0.75$  rad the torque error decreases. In fact, when  $\kappa$  increases, the system is further away from the singularity, thus being less affected. Fig. 6.3 depicts the evolution of the gimbal rates. One may notice that the instant the system is not able to match the control torque, the gimbal rates saturate. This is related to what was reported in Section 4.4. When the system crosses the singularity, the gimbal rates become larger in an effort to meet the control torques. However, since the gimbal rates are limited by  $\dot{\gamma}_{\max}$ , the system is not able to cope with the control torques due to saturation of the gimbals. Indeed, when the system meets the singularity, the gimbal rates compensate the lack of conditioning of the Jacobian due to the existence of a singular



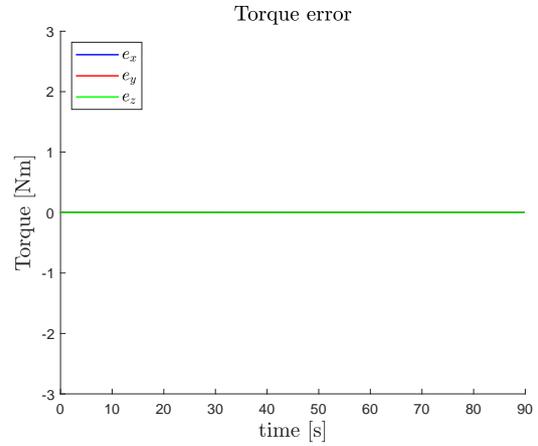
(a) Without AG



(b) With AG :  $\kappa = 0.25 \text{ rad}$



(c) With AG :  $\kappa = 0.50 \text{ rad}$

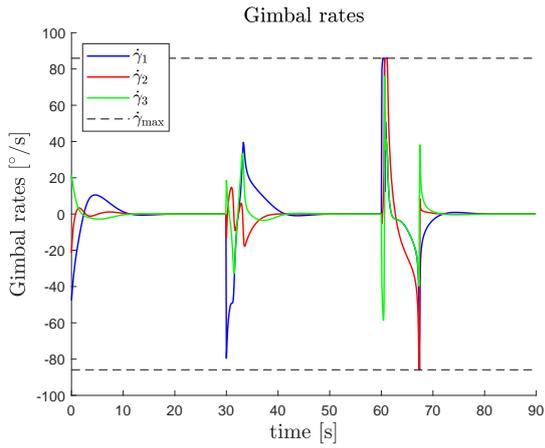


(d) With AG :  $\kappa = 0.75 \text{ rad}$

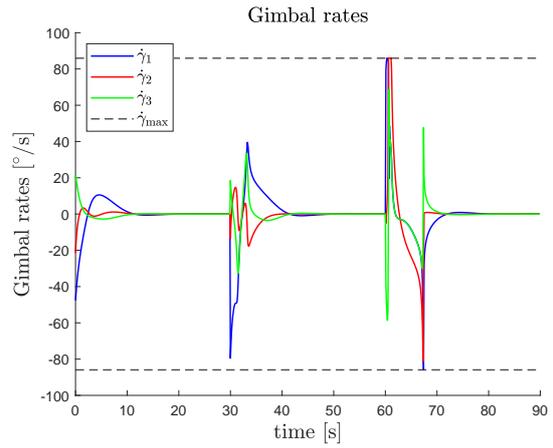
Fig. 6.2: Action governor - torque error for different threshold values.

direction, by becoming larger. On the other hand, when  $\kappa$  increases, the system requires smaller inputs to achieve the same torque. Note that even though the solutions computed with the Moore-Penrose are theoretically optimal in terms of control energy, the gimbal rates after modification display a much lower control energy. This is due to the existence of singularities, that are not accounted in the Moore-Penrose pseudoinverse.

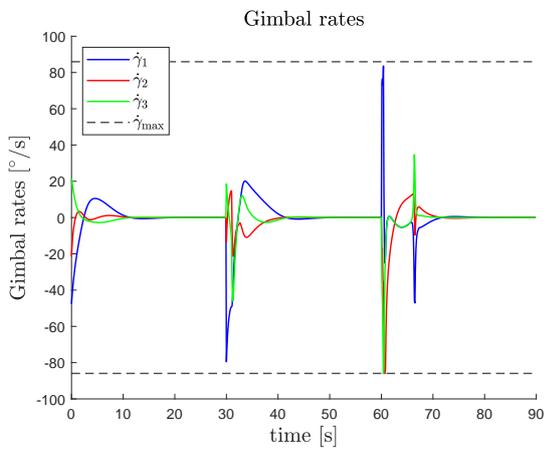
Figs. 6.4-6.6, illustrate the distance to the singularity, the manipulability index, and the condition number, respectively. Without the use of the AG, the distance to the closest singularity approaches zero at  $t \approx 61 \text{ s}$  and  $t \approx 68 \text{ s}$ . With the addition of the AG, the system is bounded to  $d > 0.75 \text{ rad}$ , hence never approaching the singularity. The manipulability index plot in Fig. 6.5 has the same behaviour as the distance plot. In fact, they are both related, as when the determinant of  $JJ^T$  decreases, the distance to the singularity decreases accordingly. The effect of the singularity is also obvious in Fig. 6.6. When the system crosses the singularity, the condition number diverges. This happens because  $\sigma_{\min}$  tends to zero due to the proximity of the singular direction. For  $\kappa = 0.75 \text{ rad}$ , the conditioning of the Jacobian is kept within a reasonable range.



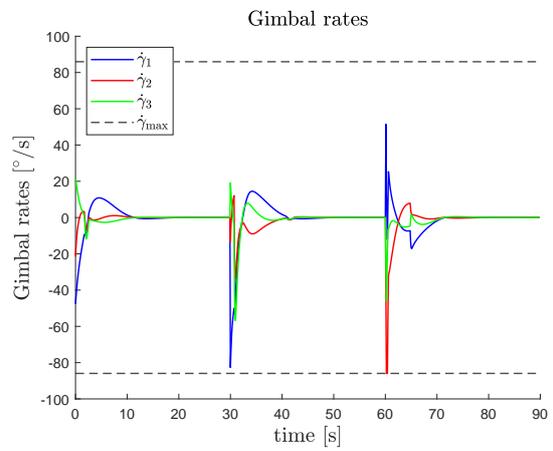
(a) Without AG



(b) With AG :  $\kappa = 0.25 \text{ rad}$

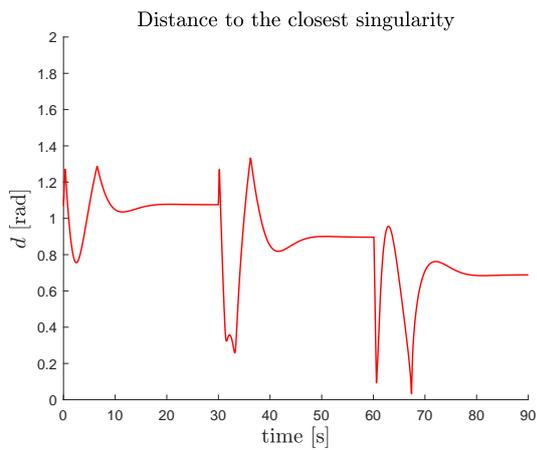


(c) Without AG :  $\kappa = 0.50 \text{ rad}$

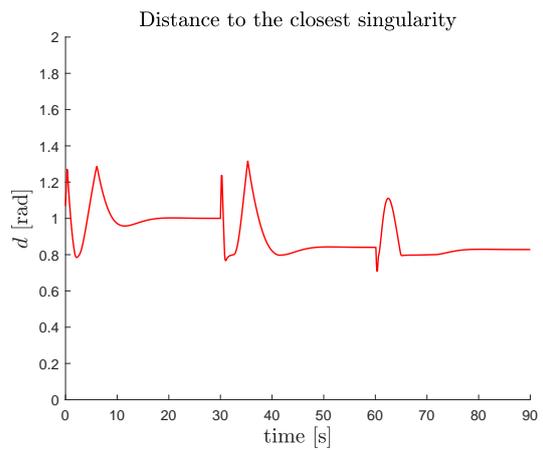


(d) With AG :  $\kappa = 0.75 \text{ rad}$

Fig. 6.3: Action governor - gimbal rates for different threshold values.



(a) Without AG



(b) With AG :  $\kappa = 0.75 \text{ rad}$

Fig. 6.4: Distance to the closest singularity: with AG vs. without AG.

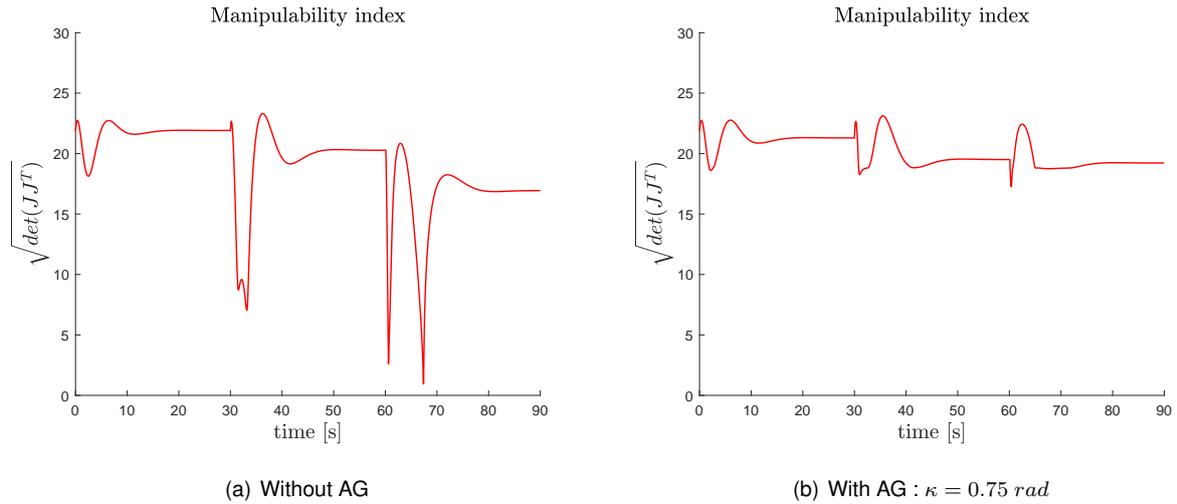


Fig. 6.5: Manipulability index: with AG vs. without AG.

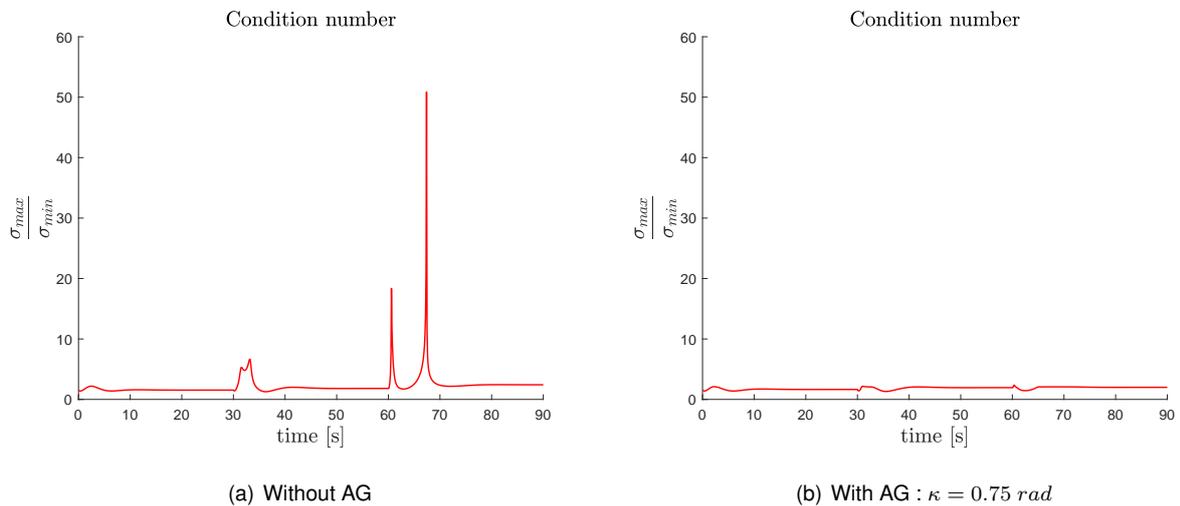


Fig. 6.6: Condition number: with AG vs. without AG.

The nullspace scaling component is represented in Fig. 6.7. When  $\lambda$  is null, the action governor applies the optimal-energy gimbals rates  $\hat{\gamma}_0$  to the system. When  $\lambda$  is different from zero, the action governor modifies  $\hat{\gamma}_0$  in order to drive the system away from the singularity exclusion zone. This shows that the system is capable of avoiding singularities by using the set of redundant solutions provided by the use of an extra CMG.

Note that the triangular CMG system is only meant for steering in the  $xy$  plane. Therefore, the  $z$  direction is not affected by singularities, which is why there is no torque error along the  $z$  direction in Fig. 6.2. This one is being controlled by a RW, which speed is shown in Fig. 6.8.

Finally, Fig. 6.9 shows the gimbal trajectory for the simulation without the AG. It is clear that the system enters the neighbourhood of two different 1h singularity lines. These are the instants where torque matching is not verified and the gimbal rates saturate. On the other hand, Fig. 6.10 shows that the AG prevents the system from entering the neighbourhood of the singularities.

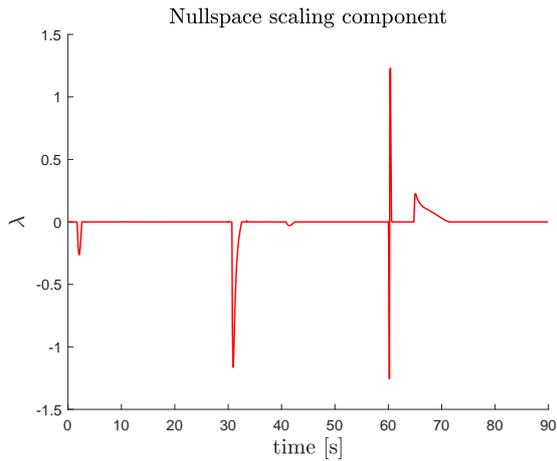


Fig. 6.7: Nullspace scaling component.

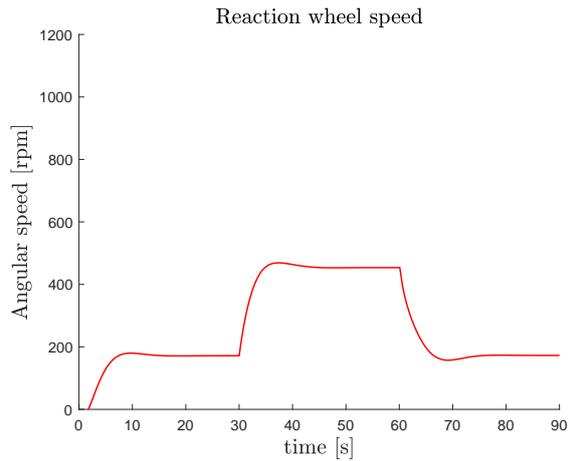
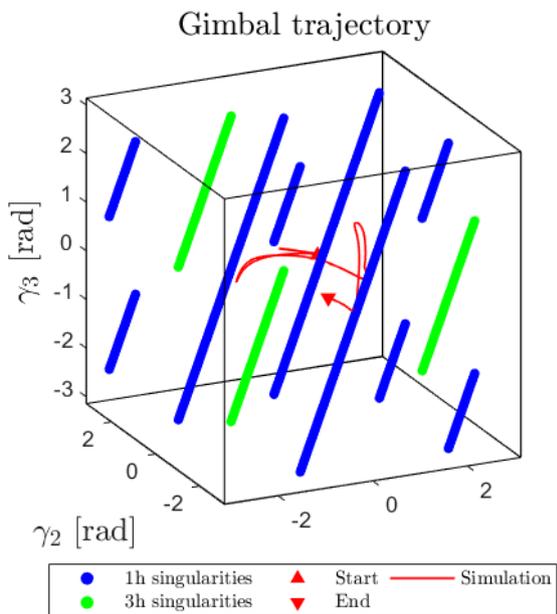
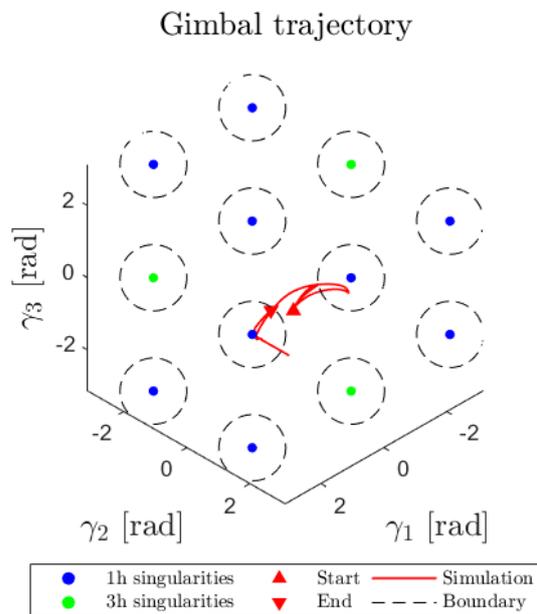


Fig. 6.8: Reaction wheel speed.



(a) 3D View



(b) Plane of view:  $(1, 1, 1)^T$

Fig. 6.9: Gimbal trajectory without AG.

Whilst the AG displays promising results by varying the singularity distance  $\kappa$ , singularity avoidance is not always accomplished. The actuation capability of the system may not be enough to avoid a singularity using a one-step ahead prediction strategy. Sometimes, the system dives so deep into a singularity region, that it becomes impossible to recover without having a torque error. This can obviously be handled by increasing  $\kappa$ , however, the motion of the gimbals cannot be arbitrarily constrained.

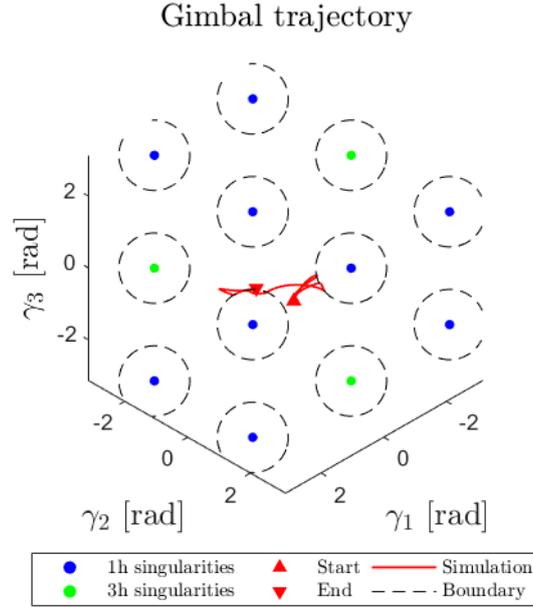


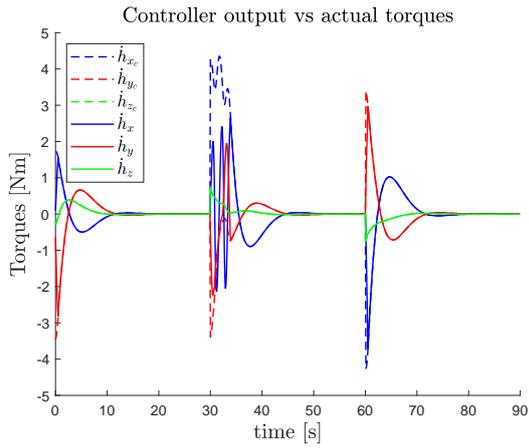
Fig. 6.10: Gimbal trajectory with AG :  $\kappa = 0.75$  rad.

## 6.2 MPC-based Allocation Results

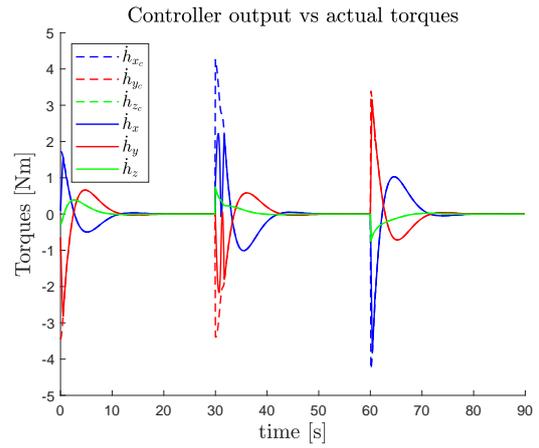
### 6.2.1 Main Results

The MPC-based allocation strategy was tested for different horizons and both with and without the presence of the singularity term. The optimization weights were tuned to  $\mathbf{H} = \text{diag}(1, 1, 1)$ ,  $\mathbf{U} = \text{diag}(0.02, 0.02, 0.02)$ ,  $\mathbf{M} = \text{diag}(0.05, 0.05, 0.05)$ , and  $\rho = 400$ . The distance threshold parameter,  $\kappa$ , was set to  $0.5 \text{ rad}$ . The initial orientation of the satellite was set to  $(\phi, \theta, \psi) = (10^\circ, -20^\circ, 0)$ , and the gimbals were initialized at  $\boldsymbol{\gamma} = (0, \pi/6, -\pi/6)^T$ . In the following simulations, the prediction and the control horizons were set equal. Lastly, the attitude reference in (6.1) was used.

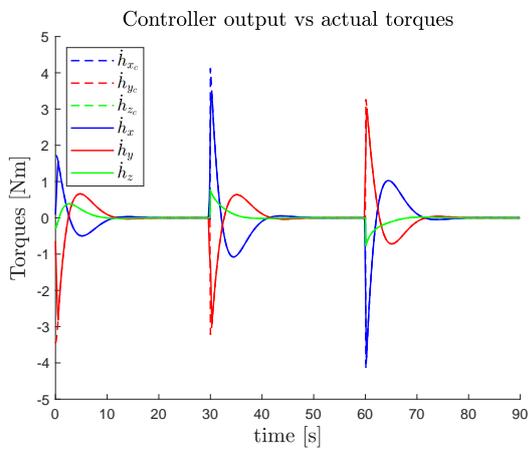
The control torques and the actual torques produced by the system are provided in Fig. 6.11. Moreover, Fig. 6.12 illustrates the reference tracking results. For a horizon of one-time step, and without the use of the singularity term, it is obvious that the system meets a singularity at  $t \approx 30 \text{ s}$ . This translates into a significant torque mismatch that has a direct influence on the attitude tracking, as depicted in Fig. 6.12(a). For the same horizon, but using the singularity term, the torque mismatch is not as large as without the singularity term. Hence, the effectiveness of the singularity term is confirmed. For a horizon of five-time steps, i.e.  $0.5 \text{ s}$ , torque matching is verified for both scenarios, leading to a singularity-free reference tracking as observed in Fig. 6.12(b). In fact, the results are the same for both formulations for a horizon of five-time steps.



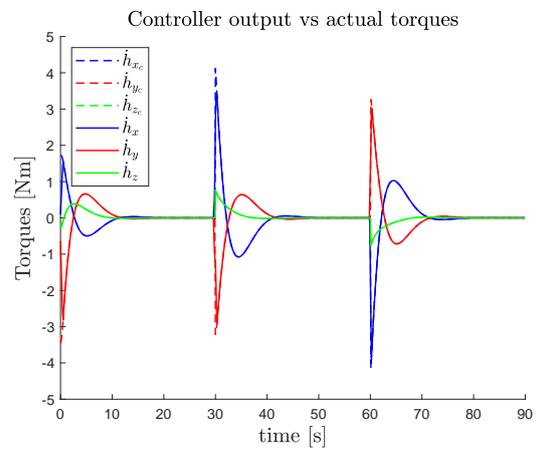
(a) Horizon = 1 (no singularity term)



(b) Horizon = 1 + singularity term

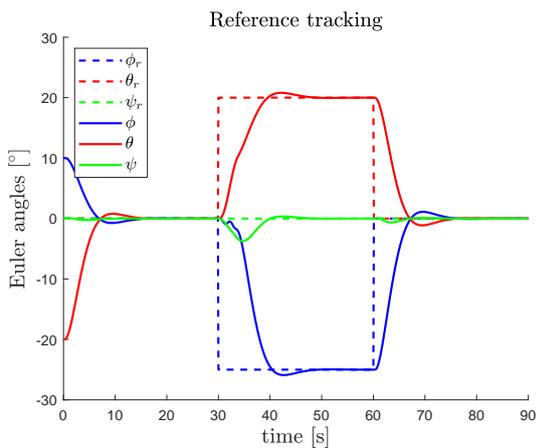


(c) Horizon = 5 (no singularity term)

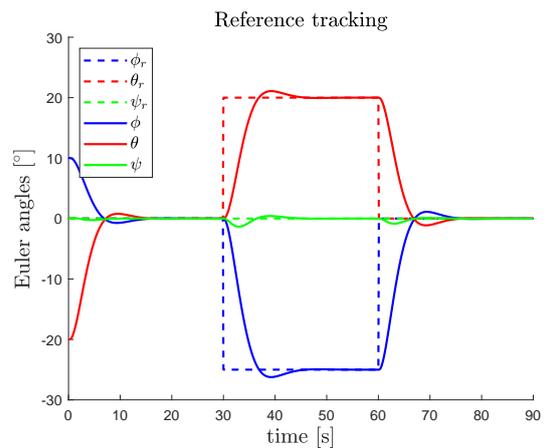


(d) Horizon = 5 + singularity term

Fig. 6.11: MPC-based allocation: control torques vs. actual torques.



(a) Horizon = 1 (no singularity term)



(b) Horizon = 5

Fig. 6.12: Reference tracking using the MPC-based allocation algorithm.

Note that the quaternion estimate  $\bar{q}$  was converted to the Euler angles representation for visualization purposes, as given by [43]

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} \text{atan2}(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \\ \arcsin(2(q_0q_2 - q_3q_1)) \\ \text{atan2}(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \end{pmatrix}. \quad (6.3)$$

The gimbal rates are shown in Fig. 6.13. For a horizon of one-time step, and without using the singularity term, the system requires a huge control effort to overcome the singularity effect. Even though the gimbal rates do not saturate, the gimbals accelerations do, which reflects in a large control energy. When the singularity term is added to the cost function, an overall reduction of the control energy is witnessed. Finally, the gimbal rates for a horizon of five-time steps are minimal.

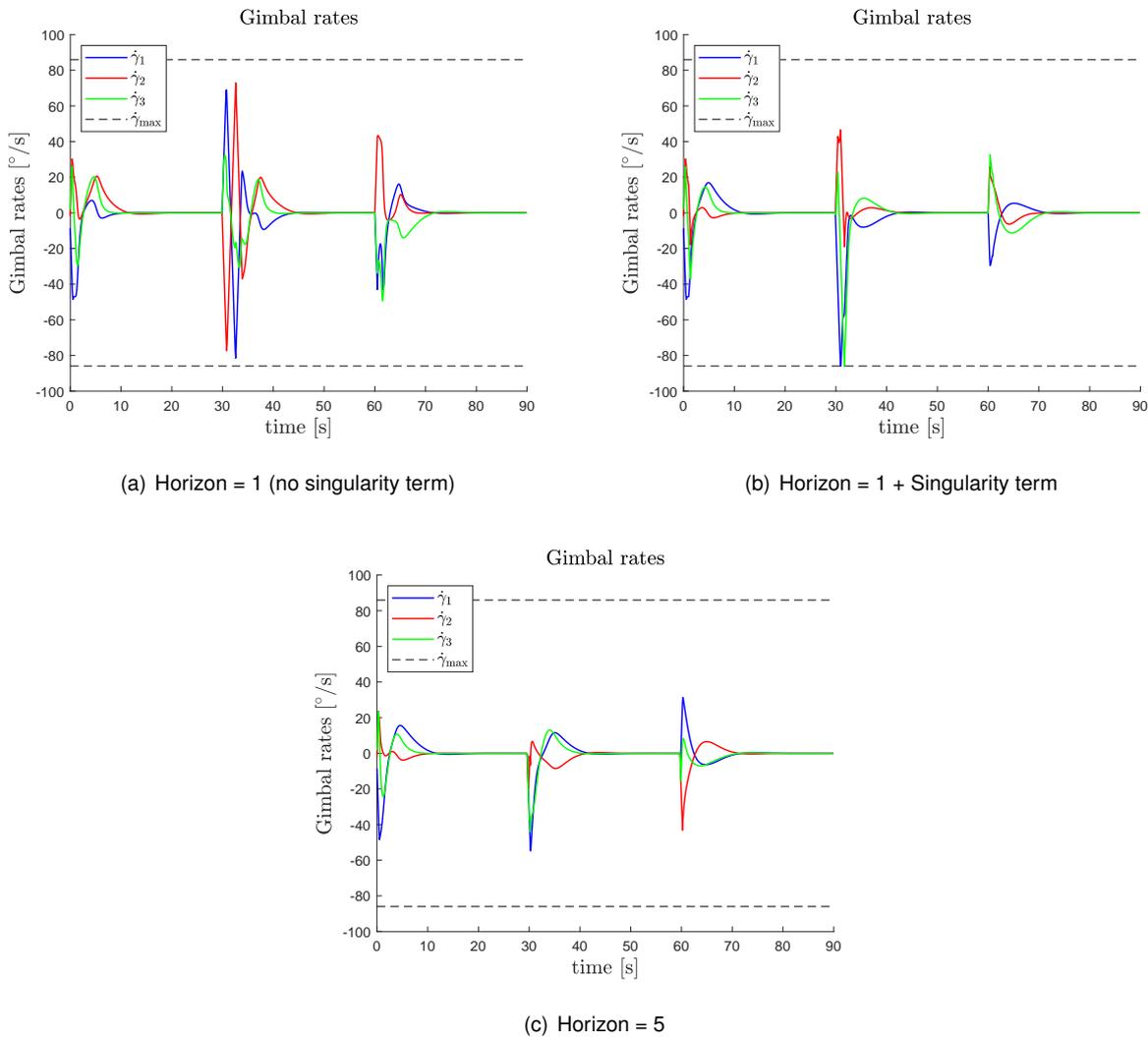


Fig. 6.13: Gimbal rates using the MPC-based allocation algorithm.

The plot in Fig. 6.14 confirms that the system crosses a singularity for  $t \approx 30$  s, when using an horizon of one-time step without the singularity term. On the other hand, when using an horizon of five-time steps, the system stays at least  $0.5$  rad away from the closest singularity, as defined by the

parameter  $\kappa$ . Thereby, the system does not display a jerky behaviour at  $t \approx 30$  s, hence minimizing both the torque error and the control effort.

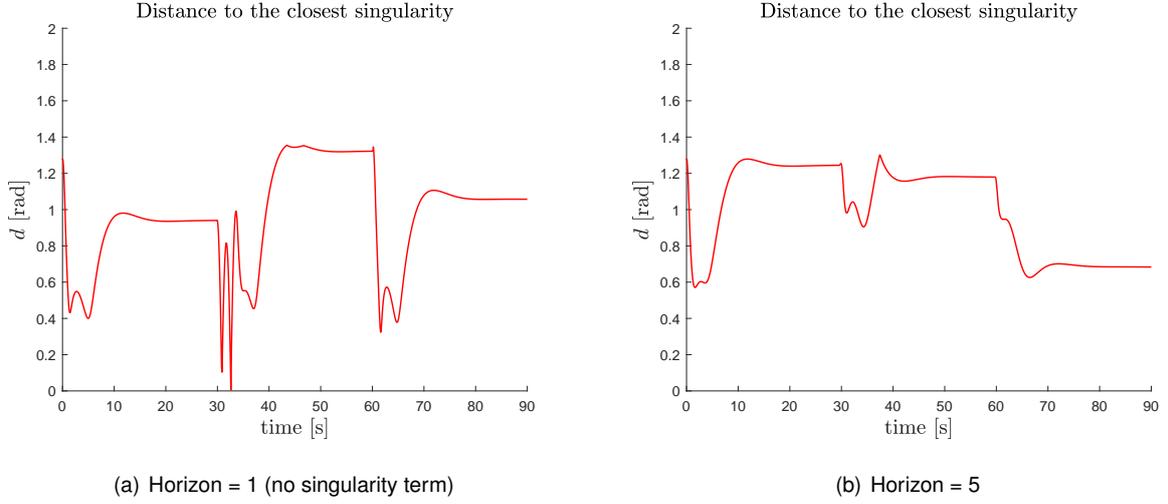


Fig. 6.14: Distance to the closest singularity using the MPC-based allocation algorithm.

The gimbal trajectory shown in Fig. 6.15 validates the aforementioned conclusions. By increasing the MPC horizon, the system is able to predict the system's state a few steps ahead in time, and therefore anticipate the presence of a singularity. The closeness of a singularity is explicitly expressed in the cost function through the singularity term. However, the torque error term and the control energy terms also account for the existence of singularities implicitly. Since torque error is expected under the presence of a singularity, it is likely that by penalizing the torque error in the cost function, the system already performs some kind of singularity avoidance. The same reasoning applies to the control energy terms. Moreover, the hard constraints imposed on the gimbal rates and accelerations implicitly constrain the motion of the gimbals towards the singularities. Nevertheless, the system benefits from the singularity term, as seen before. Finally, Fig. 6.16 shows the torque error energy,  $\mathcal{E}$ , Fig. 6.17 shows the control magnitude energy,  $\mathcal{C}$ , and Fig. 6.18 shows the control variation energy,  $\mathcal{V}$ , obtained for the different horizons.  $\mathcal{E}$ ,  $\mathcal{C}$  and  $\mathcal{V}$  were calculated, respectively, by

$$\mathcal{E} = \int_{t_0}^T \|e\|_2 dt, \quad (6.4)$$

$$\mathcal{C} = \int_{t_0}^T \|u\|_2 dt, \quad (6.5)$$

and

$$\mathcal{V} = \int_{t_0}^T \|\dot{u}\|_2 dt, \quad (6.6)$$

where  $e = \dot{h} - \dot{h}_c$ ,  $u = \dot{\gamma}$ ,  $\dot{u} = \ddot{\gamma}$ ,  $t_0 = 0s$ , and  $T = 90s$ . The results were obtained by numerical integration using the rectangle method.

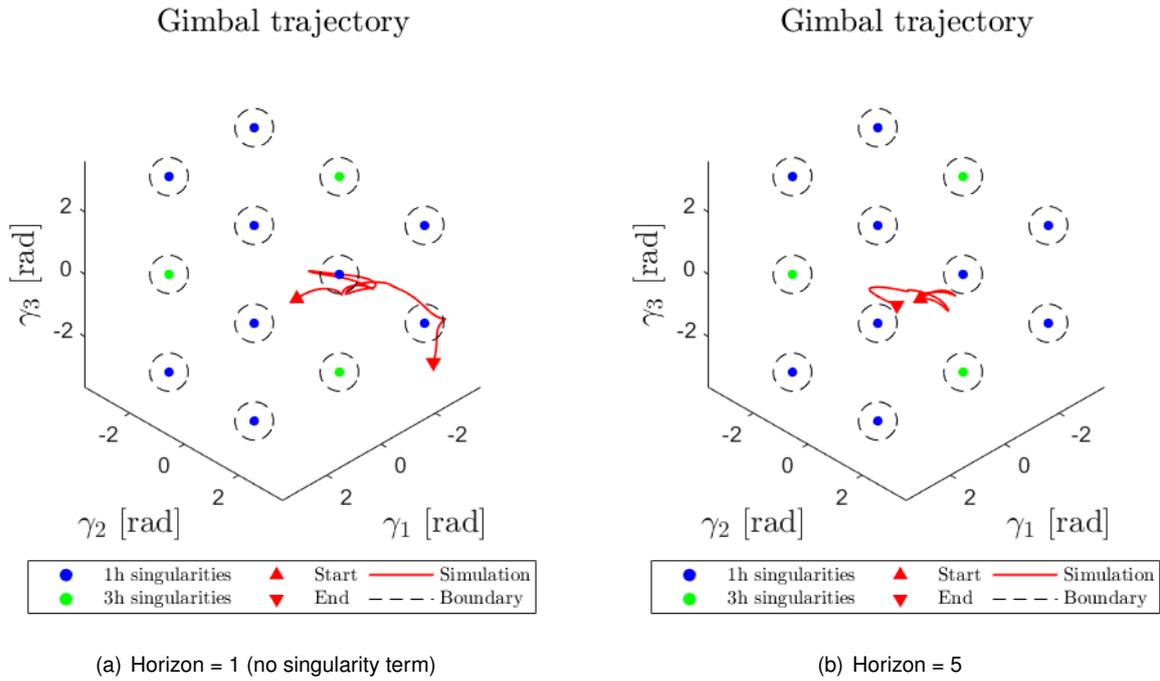


Fig. 6.15: Gimbal trajectory using the MPC-based allocation algorithm.

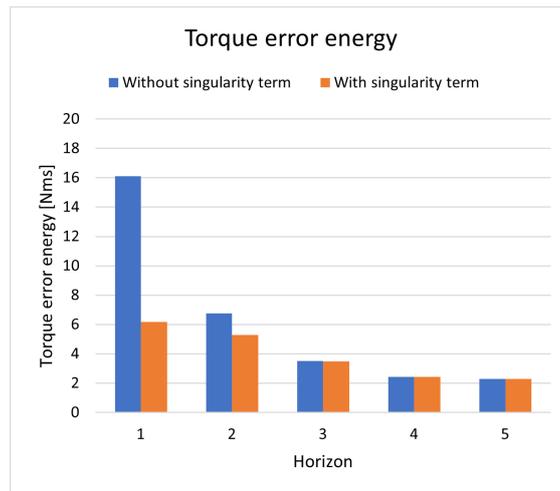


Fig. 6.16: Torque error vs. horizon.

Fig. 6.16 clearly shows a reduction of the torque error as the horizon increases. Besides this, the impact of the singularity term reflects largely for small horizons, and becomes smaller as the horizon increases. In fact, the effect of the singularity term decreases until the point where it has no longer influence. In this case, the torque error is the same for both formulations whenever the horizon exceeds three-time steps, i.e.  $0.3\text{ s}$ . This happens because the system slowly realizes that it has to stay distant from the singularities in order to accomplish torque matching. The control magnitude and control variation plots in Figs. 6.17 and 6.18, respectively, display a decreasing tendency as the horizon increases. Similarly to the torque error, the control energy converges for both formulations when the horizon exceeds three-time steps. The control energy, for the formulation that considers the singularity term, is slightly larger when the horizon equals two-time steps. This happens because the system is exploiting

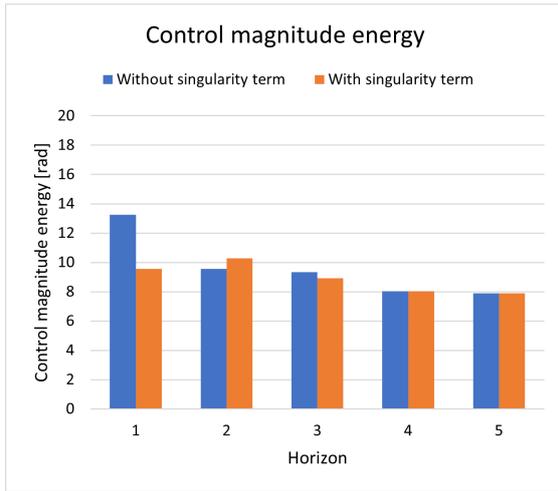


Fig. 6.17: Control energy magnitude vs. horizon.

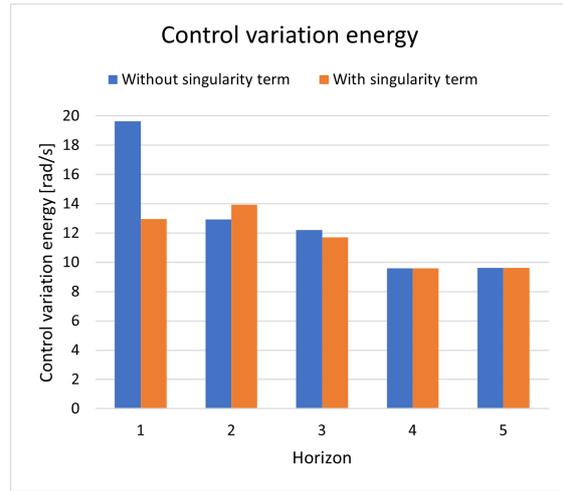


Fig. 6.18: Control energy variation vs. horizon.

the set of redundant solutions, that are not energy-minimal, to escape the singularity effect as fast as possible. Consequently, the torque error is smaller when comparing to the formulation that does not encompass the singularity term.

## 6.2.2 Execution Times

The execution times of the allocation MPC were measured for the different horizons as shown in Fig. 6.19. The impact of the Euclidean distance term was also analysed and compared to formulations that do not encompass a singularity term, and formulations that consider the condition number instead of the Euclidean distance. Note that the execution times are only relative to the execution of the allocation MPC, thus not considering the remaining software in the ACS. The mean execution times were obtained by averaging the entire simulation. To have an idea of the total time required to run a simulation, one has to multiply the mean execution times by the simulation time,  $T$ , and the sampling frequency,  $1/T_s$ . For reference, an 11<sup>th</sup> Gen Intel(R) Core(TM) i7-1165G7 CPU unit, with base frequency 2.8 GHz, was used for this analysis. Besides this, the simulation was carried out with the least number of processes running in parallel in the operating system. Note, however, that the execution times could be significantly faster if the code was optimized and implemented in a real-time systems programming language, such as C/C++. Besides this, high speed solvers, such as Embotech's FORCESPRO<sup>2</sup>, can boost the execution up to 100 times. Fig. 6.19 provides, however, a fair comparison in terms of relative execution times.

The execution times generally increase as the horizon increases. This is expected as when the horizon grows, more terms and constraints are considered in the optimization. Additionally, the Euclidean distance term introduces a slight increase in the execution times. This impact, however, is much more significant when using the condition number. This shows that the Euclidean distance metric does not influence the computational cost significantly. In fact, a convex optimization problem can be achieved with the use of the Euclidean distance, whereas the use of the condition number typically leads to a non-convex optimization problem.

<sup>2</sup><https://www.embotech.com/products/forcespro/overview/>.

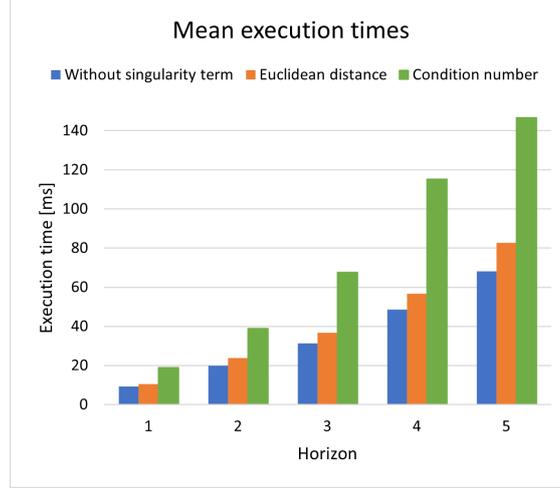


Fig. 6.19: Execution times for the different formulations.

### 6.2.3 Linearization Analysis

To validate the robustness of the linearization of the torque in (5.39), an analysis that compares the torques computed using the nonlinear model and the linear model was carried out. These results were obtained for the parameters presented in Table 6.1 and the attitude reference trajectory in (6.1). Only the part of the simulation where  $30 \text{ s} \leq t \leq 35 \text{ s}$  was used for analysis (where the torque magnitude is large enough along the entire window). The horizon was set to ten time-steps to conclude about the impact of the linearization over a larger prediction window. Table 6.2 shows the results obtained for the worst-case error, between the nonlinear and linear model, witnessed in the aforementioned time-window. This was achieved by finding the MPC prediction whose average torque error (between the nonlinear and linear models) was maximum. Model *I* represents the nonlinear model ( $\dot{\mathbf{h}}(t) = \mathbf{J}(\gamma(t))\dot{\gamma}(t)$ ), model *II* represents the linear model found for a single equilibrium  $(\gamma_\phi, \dot{\gamma}_\phi)$ , and model *III* represents the linear model where the trajectory was linearized along the horizon, using multiple equilibria  $(\gamma_\phi(t), \dot{\gamma}_\phi(t))$ . In model *II*, the equilibrium used corresponds to the gimbals estimate at the time instant  $k$ , whereas the equilibria used in model *III* corresponds to the sequence of gimbal angles and gimbal rates computed by the MPC at the time instant  $k - 1$ , i.e.  $\Gamma(k - 1)$  and  $\dot{\Gamma}(k - 1)$ . It can be verified that the torque error  $e_l = \|(\dot{h}_x, \dot{h}_y)_I - (\dot{h}_x, \dot{h}_y)_{II,III}\|_2$  increases along the horizon for both models. However, the error is larger for model *II* than for model *III*, which confirms the benefits of linearizing the trajectory along the horizon. Furthermore, the torque computed for a horizon of ten time-steps with model *III* is only 0.694% off the torque computed with the nonlinear model. This translates into a small error whose impact on the MPC solution is negligible.

Finally, the error obtained for every time-step along the prediction horizon was further analyzed by averaging the entire time-window. Table 6.3 suggests that the overall error is larger in model *II* than in model *III*. Besides this, the error witnessed in the last time-step in the horizon for model *III* is insignificant when comparing to the actual torques produced by the system, hence proving the effectiveness of model *III* for prediction horizons of up to ten time-steps. Note that the analysis could have been performed for an even larger horizon, nonetheless, this analysis appears to be enough since small hori-

Table 6.2: Worst-case error analysis (values in [Nm]).

Horizon		1	2	3	4	5	6	7	8	9	10
<i>I</i>	$\dot{h}_x$	2.9713	3.9389	3.6058	3.2971	3.0524	2.8317	2.5831	2.3450	2.1186	2.0687
	$\dot{h}_y$	-1.7886	-2.9590	-3.0098	-2.9024	-2.6441	-2.4189	-2.2463	-2.0794	-1.9190	-1.9454
<i>II</i>	$\dot{h}_x$	2.9713	4.0104	3.7675	3.5134	3.2711	3.0314	2.7465	2.4382	2.1453	2.1057
	$\dot{h}_y$	-1.7886	-2.8718	-2.9301	-2.9589	-2.9544	-3.1199	-3.4489	-3.8457	-4.1433	-4.4224
	$e_t$	<b>0</b>	<b>0.1127</b>	<b>0.1803</b>	<b>0.2235</b>	<b>0.3797</b>	<b>0.7289</b>	<b>1.2137</b>	<b>1.7688</b>	<b>2.2244</b>	<b>2.4773</b>
	$e_t(\%)$	<b>0%</b>	<b>2.29%</b>	<b>3.84%</b>	<b>5.09%</b>	<b>9.40%</b>	<b>19.6%</b>	<b>35.5%</b>	<b>56.4%</b>	<b>77.8%</b>	<b>87.2%</b>
<i>III</i>	$\dot{h}_x$	2.9713	3.9389	3.6058	3.2972	3.0524	2.8317	2.5831	2.3449	2.1199	2.0571
	$\dot{h}_y$	-1.7886	-2.9590	-3.0098	-2.9024	-2.6440	-2.4189	-2.2463	-2.0793	-1.9179	-1.9613
	$e_t$	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0.0001</b>	<b>0.0018</b>	<b>0.0197</b>
	$e_t(\%)$	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>0.003%</b>	<b>0.063%</b>	<b>0.694%</b>

zons, of only up to five time-steps, were used in the previous sections. In fact, it was verified empirically that the allocation MPC does not require a long horizon for singularity avoidance.

Table 6.3: Average error analysis (values in [Nm]).

Horizon	1	2	3	4	5	6	7	8	9	10
<i>II</i>	0	0.0095	0.0304	0.0597	0.0967	0.1395	0.1835	0.2250	0.2610	0.2832
<i>III</i>	0	0	0	0	0.0001	0.0001	0.0001	0.0001	0.0003	0.0010

# Chapter 7

## Conclusions and Future Work

### 7.1 Conclusions

The long-standing singularity problem, that occurs in spacecrafts employing control moment gyros clusters, was addressed in this thesis. As a means to understand the impact of singularities in a spacecraft's ACS, a rigorous analysis was carried out in Chapter 4. Herein, the singularity envelope of two different CMG clusters was characterized and analyzed in detail. An SVD analysis of the Jacobian was performed to address the effect of a singularity on the actuation capability of a spacecraft. For non-singular gimbal configurations, a good conditioning of the Jacobian allows the creation of torque in all possible directions. On the other hand, for singular gimbal configurations, the Jacobian contains at least one singular direction that translates into a loss of controllability. In these cases, the spacecraft cannot create torque along the singular direction, resulting in an attitude tracking error.

To tackle this problem, several control allocation algorithms were proposed and tested in a simulation environment. Besides this, a new singularity metric was derived to perceive the proximity of a singularity. Contrary to common literature solutions, which typically require an online computation of the Jacobian's singular values, this solution uses a rather simple function that estimates the Euclidean distance between the gimbals and the closest singularity. The major advantage of this function is that it can be integrated as a linear constraint in a convex optimization problem, resulting in a much lower computational cost. Moreover, the convergence to an optimal solution is guaranteed as convexity is achieved.

The AG consists of a supervisory scheme that promotes singularity avoidance by adding null motion to the solution provided by the Moore-Penrose pseudoinverse. It was noticed that as the minimum required distance to the singularity increased, the earlier the system anticipated its proximity. However, some singularities cannot be avoided by predicting the gimbal state one single step ahead in time. Moreover, the gimbal motion becomes limited by increasing the distance. There is a point where this limitation translates into a generalized increase of the control energy spent to achieve the command torque.

To overcome these limitations, an allocation algorithm that computes the gimbal rates in an integrated way was derived. However, similarly to the previous approach, this algorithm lacks effectiveness

by predicting the state of the gimbals a single step ahead in time. Thereupon, an MPC-based approach was derived to extend the prediction of the gimbals over a fixed horizon. In this way, the system can predict the presence of a singularity a few steps ahead and take control actions that lead to singularity-free gimbal states. Just like the AG, this solution seeks to avoid singularities by exploiting the null motion of the system. One could notice that the behaviour of the system improved as the prediction horizon increased. Besides this, the effectiveness of the novel singularity index was confirmed for small prediction horizons. Furthermore, an analysis has shown that the singularity index based on the Euclidean distance leads to a much lower execution time than that obtained with the condition number. In fact, convexity was achieved for this approach, making it relatively fast compared to some literature solutions that resort to non-convex techniques for singularity avoidance.

Finally, the objectives outlined in Section 1.2 were successfully accomplished. The impact of singularities on spacecraft attitude control was studied in detail. Then, an optimal allocation framework for singularity avoidance was derived. This solution allowed the system to avoid all of the internal singularities of a CMG cluster. Lastly, convexity was achieved considering reasonable assumptions.

## 7.2 Future Work

The allocation algorithms should be validated in the presence of perturbations. These should include internal unmodelled perturbations, such as sensor noise and flexible vibration modes, and external perturbations, such as the gravity gradient, solar radiation pressure, atmospheric drag, and third-body perturbations.

The proposed singularity index does not apply to every CMG system. In some cases, the singularity envelope cannot be described by lines or planes, hence yielding to nonlinear representations that might be costly to store in an on-board computer. Therefore, convexification techniques to approximate the singularity envelope of a CMG array by a convex set should also be investigated.

Since the prediction of the control torques relies on a model of the system, there is a chance of witnessing a mismatch between the model and the real system. Therefore, the robustness of the proposed methods must be evaluated in a real satellite.

Finally, the proposed algorithms should be implemented in a real-time systems programming language to better understand the overhead caused by their integration into the satellite's ACS.

# Bibliography

- [1] A. Gaude and V. Lappas. "Design and Structural Analysis of a Control Moment Gyroscope (CMG) Actuator for CubeSats". *Aerospace*, 7(5):55, May 2020. DOI: 10.3390/aerospace7050055.
- [2] H. Evain, T. Solatges, A. Brunet, A. Ribeiro, L. Sipile, M. Rognant, D. Alazard, and J. Mignot. "Design and Control of a Nano-Control Moment Gyro Cluster for Experiments in a Parabolic Flight Campaign". In *20<sup>th</sup> IFAC World Congress*, Toulouse, France, July 2017.
- [3] R. Votel and D. Sinclair. "Comparison of Control Moment Gyros and Reaction Wheels for Small Earth-Observing Satellites". In *26th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, USA, August 2012.
- [4] B. Wie. "Space Vehicle Dynamics and Control". American Institute of Aeronautics and Astronautics (AIAA), 2<sup>nd</sup> edition, 2008. ISBN:978-1563479533.
- [5] F. Leve, B. Hamilton, and M. Peck. "Spacecraft Momentum Control Systems". Springer, 1<sup>st</sup> edition, 2015. ISBN:978-3319225623.
- [6] S. Boyd and L. Vandenberghe. "Convex Optimization". Cambridge University Press, 1<sup>st</sup> edition, 2008. ISBN:978-0521833783.
- [7] N. Bedrossian, J. Paradiso, E. Bergmann, and D. Rowell. "Steering law design for redundant single gimbal control moment gyroscopes". *Journal of Guidance, Control and Dynamics*, 13(6), November-December 1990. DOI: 10.2514/3.20582.
- [8] K. Ford and C. Hall. "Singular direction avoidance steering for control-moment gyros". *Journal of Guidance, Control and Dynamics*, 23(4):648–656, July-August 2000. DOI: 10.2514/2.4610.
- [9] B. Wie, D. Bailey, and C. Heiberg. "Singularity robust steering logic for redundant single-gimbal control moment gyros". *Journal of Guidance, Control and Dynamics*, 24(5):865–872, September 2001. DOI: 10.2514/2.4799.
- [10] A. Pechev. "Feedback-based steering law for control moment gyros". *Journal of Guidance, Control and Dynamics*, 30(3):848–855, May 2007. DOI: 10.2514/1.27351.
- [11] T. Sands. "Fine pointing of military spacecraft". PhD thesis, Naval Postgraduate School, Monterey, California, 2007.

- [12] D. Cornick. "Singularity avoidance control laws for single gimbal control moment gyros". In *Guidance and Control Conference*, Boulder, CO, USA, August 1979.
- [13] M. Kuhns and A. Rodriguez. "A preferred trajectory tracking steering law for spacecraft with redundant CMGs". In *Proceedings of 1995 American Control Conference - ACC'95*, volume 5, pages 3111–3115, Seattle, WA, USA, June 1995. DOI: 10.1109/ACC.1995.532089.
- [14] H. Kurokawa. "Survey of theory and steering laws of single-gimbal control moment gyros". *Journal of Guidance, Control and Dynamics*, 30(5):1331–1340, September 2007. DOI: 10.2514/1.27316.
- [15] H. Leeghim, C. Lee, J. Jin, and D. Kim. "A Singularity-free Steering Law of Roof Array of Control Moment Gyros for Agile Spacecraft Maneuver". *International Journal of Control Automation and Systems*, 18(7):1679–1690, January 2020. DOI: 10.1007/s12555-019-0008-6.
- [16] H. Kurokawa. "Constrained steering law of pyramid-type control moment gyros and ground tests". *Journal of Guidance, Control and Dynamics*, 20(3), May-June 1997. DOI: 10.2514/2.4095.
- [17] F. Leve and N. Fitz-Coy. "Hybrid steering logic for single-gimbal control moment gyroscopes". *Journal of Guidance, Control and Dynamics*, 33(4):1202–1212, July 2010. DOI: 10.2514/1.46853.
- [18] H. Leeghim, H. Bang, and J. Park. "Singularity avoidance of control moment gyros by one-step ahead singularity index". *Acta Astronautica*, 64:935–945, May-June 2009. DOI: 10.1016/j.actaastro.2008.11.004.
- [19] K. Takada and H. Kojima. "Receding horizon control on steering of control moment gyro for fast attitude maneuvers". *Transactions of the Japan Society for Aeronautical and Space Sciences*, 52 (175):1–10, May 2009. DOI: 10.2322/tjsass.52.1.
- [20] K. Takada, H. Kojima, and N. Matsuda. "Control moment gyro singularity-avoidance steering control based on singular-surface cost function". *Journal of Guidance, Control and Dynamics*, 33(5):1442–1450, September 2010. DOI: 10.2514/1.48381.
- [21] H. Kojima, R. Nakamura, and S. Keshtkar. "Model predictive steering control law for double gimbal scissored-pair control moment gyros". *Acta Astronautica*, 183(26), April 2021. DOI: 10.1016/j.actaastro.2021.03.023.
- [22] Y. Ikeda, T. Nakajima, and Y. Chida. "Attitude Control of Spacecraft by NMPC with Consideration of Singularity Avoidance of CMG". In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, Maui, HI, USA, December 2012. DOI: 10.1109/CDC.2012.6426827.
- [23] J. Ramírez and L. Hewing. "Sequential Convex Programming for Optimal Line of Sight Steering in Agile Missions". June 2022. DOI: 10.48550/arXiv.2206.06061.
- [24] S. Viswanathan, A. Sanyal, F. Leve, and N. McClamroch. "Dynamics and Control of Spacecraft with a Generalized Model of Variable Speed Control Moment Gyroscopes". *Journal of Dynamics Systems Measurement and Control*, 137(7), November 2014. DOI: 10.1115/1.4029626.

- [25] Y. Yang. "Singularity-Free Model Predictive Spacecraft Attitude Regulation Using Variable-Speed Control Moment Gyroscope Model". *IEEE Transactions on Aerospace and Electronic Systems*, 54 (99), June 2018. DOI: 10.1109/TAES.2018.2796878.
- [26] C. Papakonstantinou, V. Lappas, and V. Kostopoulos. "A Gimballed Control Moment Gyroscope Cluster Design for Spacecraft Attitude Control". *Aerospace*, 8(9):273, September 2021. DOI: 10.3390/aerospace8090273.
- [27] A. Mony. "RRT-Based Steering Law for Singularity Avoidance of Control Moment Gyros used for Spacecraft Target Acquisition". In *Indian Control Conference 2019*, IIT Hyderabad, December 2019. DOI: 10.1109/ICC47138.2019.9123199.
- [28] C. Papakonstantinou, V. Lappas, H. Schaub, and V. Kostopoulos. "Global Steering for Control Moment Gyroscope Clusters Using Heuristic Variable Search Techniques". *Journal of Spacecraft and Rockets*, 58(4):1–12, July-August 2021. DOI: 10.2514/1.A34850.
- [29] W. MacKunis, K. Dupree, S. Bhasin, and W. Dixon. "Adaptive Neural Network Satellite Attitude Control in the Presence of Inertia and CMG Actuator Uncertainties". In *2008 American Control Conference*, Seattle, WA, USA, June 2008.
- [30] C. Papakonstantinou, I. Daramouskas, V. Lappas, V. Moulianitis, and V. Kostopoulos. "A Machine Learning Approach for Global Steering Control Moment Gyroscope Clusters". *Aerospace*, 9(3):164, March 2022. DOI: 10.3390/aerospace9030164.
- [31] Y. Yang. "Spacecraft Modeling, Attitude Determination and Control: Quaternion-based Approach". CRC Press, 1<sup>st</sup> edition, 2019. ISBN:978-1138331501.
- [32] T. Kaskalis and K. Margaritis. "Minimum singular value estimation of bipolar matrices". In *Proceedings of the 10th WSEAS International Conference on computers*, Vouliagmeni, Athens, Greece, July 2006.
- [33] K. Lange. "Numerical Analysis for Statisticians". Springer, 2<sup>nd</sup> edition, 2010. ISBN:978-1441959447.
- [34] B. Datta. "Numerical Linear Algebra and Applications". Society for Industrial and Applied Mathematics, 2<sup>nd</sup> edition, 2010. ISBN:978-0898716856.
- [35] K. Kolev. "Convexity in Image-Based 3D Surface Reconstruction". PhD thesis, Department of Computer Science, Technical University of Munich, Germany, 2012.
- [36] N. Li, K. Han, A. Girard, H. Tseng, D. Filev, and I. Kolmanovsky. "Action Governor for Discrete-Time Linear Systems With Non-Convex Constraints". *IEEE Control Systems Letters*, 5(1):121–126, June 2020. DOI: 10.1109/LCSYS.2020.3000198.
- [37] Y. Li, N. Li, H. Tseng, A. Girard, D. Filev, and I. Kolmanovsky. "Robust Action Governor for Discrete-Time Piecewise Affine Systems With Additive Disturbances". *IEEE Control Systems Letters*, 6: 950–955, June 2021. DOI: 10.1109/LCSYS.2021.3087627.

- [38] C. Bordons, F. Garcia-Torres, and M. Ridao. *"Model Predictive Control of Microgrids"*. Springer, 1<sup>st</sup> edition, 2020. ISBN:978-3030245702.
- [39] M. Schwenzer, M. Ay, T. Bergs, and D. Abel. *"Review on model predictive control: an engineering perspective"*. *The International Journal of Advanced Manufacturing Technology*, 117:1327–1349, August 2021. DOI: 10.1007/s00170-021-07682-3.
- [40] B. Cowan. *"Classical Mechanics"*. Springer, 1<sup>st</sup> edition, 1984. ISBN:978-9401091718.
- [41] O. Knill. *"Cauchy–Binet for pseudo-determinants"*. *Linear Algebra and its Applications*, 459:522–547, October 2014. DOI: 10.1016/j.laa.2014.07.013.
- [42] E. Fresk and G. Nikolakopoulos. *"Full Quaternion Based Attitude Control for a Quadrotor"*. In *2013 European Control Conference (ECC)*, Zurich, Switzerland, July 2013. DOI: 10.23919/ECC.2013.6669617.
- [43] D. Henderson. *"Euler Angles, Quaternions, And Transformation Matrices Working Relationships"*. In *Shuttle Program*. National Aeronautics and Space Administration (NASA), July 1977.

## Appendix A

# CMGs and RWs Specifications

Table A.1: Control moment gyros specifications.

Control Moment Gyros			
Name	Mass [Kg]	Torque [Nm]	Source
M50 Honeywell	28	75	<a href="https://www.satcatalog.com/component/m50-cmg/">https://www.satcatalog.com/component/m50-cmg/</a>
MicroCMG Veoware	2.75	1.1	<a href="https://satsearch.co/products/veoware-space-micro-cmg">https://satsearch.co/products/veoware-space-micro-cmg</a>
MiniCMG Veoware	3.25	4.4	<a href="https://satsearch.co/products/veoware-space-mini-cmg">https://satsearch.co/products/veoware-space-mini-cmg</a>
MicroCMG Astrofein	8	14	<a href="https://www.astrofein.com/microcmg/">https://www.astrofein.com/microcmg/</a>
Microsat CMG Honeybee	0.6	0.112	<a href="https://www.honeybeerobotics.com/wp-content/uploads/2019/12/Honeybee-Micro-Sat-CMG-Spec-Sheet.pdf">https://www.honeybeerobotics.com/wp-content/uploads/2019/12/Honeybee-Micro-Sat-CMG-Spec-Sheet.pdf</a>
CMG 15-45S Airbus	18.4	45	<a href="https://www.airbus.com/sites/g/files/jlcbta136/files/2022-02/AVIONICS-CMG15-45-v5-2022.pdf">https://www.airbus.com/sites/g/files/jlcbta136/files/2022-02/AVIONICS-CMG15-45-v5-2022.pdf</a>
CMG 40-60S Airbus	38	60	<a href="https://mediaassets.airbus.com/pm_38_552_552268-1acx8572rs.pdf">https://mediaassets.airbus.com/pm_38_552_552268-1acx8572rs.pdf</a>
CMG 4-6S Airbus	13	6	<a href="https://www.satcatalog.com/component/cmg-4-6s/">https://www.satcatalog.com/component/cmg-4-6s/</a>
CMG 75-75S Airbus	69	75	<a href="https://www.airbus.com/sites/g/files/jlcbta136/files/2022-02/AVIONICS-CMG75-75S-v6_2022.pdf">https://www.airbus.com/sites/g/files/jlcbta136/files/2022-02/AVIONICS-CMG75-75S-v6_2022.pdf</a>
CMG-12 Blue Canyon	18	12	<a href="https://satsearch.co/products/bluecanyontech-cmg-12">https://satsearch.co/products/bluecanyontech-cmg-12</a>
CMG-8 Blue Canyon	10	8	<a href="https://satsearch.co/products/bluecanyontech-cmg-8">https://satsearch.co/products/bluecanyontech-cmg-8</a>

Table A.2: Reaction wheels specifications.

Reaction wheels			
Name	Mass [Kg]	Torque [Nm]	Source
RW-1.0 Rocket Lab	1.380	0.1	<a href="https://www.rocketlabusa.com/assets/Uploads/RL-RW-1.0-Data-Sheet.pdf">https://www.rocketlabusa.com/assets/Uploads/RL-RW-1.0-Data-Sheet.pdf</a>
RW-0.06 Rocket Lab	0.226	0.02	<a href="https://www.rocketlabusa.com/assets/Uploads/RL-RW-0.06-Data-Sheet.pdf">https://www.rocketlabusa.com/assets/Uploads/RL-RW-0.06-Data-Sheet.pdf</a>
RW-0.03 Rocket Lab	0.185	0.002	<a href="https://www.rocketlabusa.com/assets/Uploads/RL-RW-0.03-Data-Sheet.pdf">https://www.rocketlabusa.com/assets/Uploads/RL-RW-0.03-Data-Sheet.pdf</a>
RW-0.003 Rocket Lab	0.050	0.001	<a href="https://www.rocketlabusa.com/assets/Uploads/RL-RW-0.003-Data-Sheet.pdf">https://www.rocketlabusa.com/assets/Uploads/RL-RW-0.003-Data-Sheet.pdf</a>
VRW-B-02	1	0.02	<a href="https://www.vectronic-aerospace.com/reaction-wheels/#rwsection">https://www.vectronic-aerospace.com/reaction-wheels/#rwsection</a>
VRW-C-1	2.3	0.02	<a href="https://www.vectronic-aerospace.com/reaction-wheels/#rwsection">https://www.vectronic-aerospace.com/reaction-wheels/#rwsection</a>
VRW-D-2	2	0.05	<a href="https://www.vectronic-aerospace.com/reaction-wheels/#rwsection">https://www.vectronic-aerospace.com/reaction-wheels/#rwsection</a>
VRW-D-4	2.5	0.05	<a href="https://www.vectronic-aerospace.com/reaction-wheels/#rwsection">https://www.vectronic-aerospace.com/reaction-wheels/#rwsection</a>
VRW-D-6	3	0.09	<a href="https://www.vectronic-aerospace.com/reaction-wheels/#rwsection">https://www.vectronic-aerospace.com/reaction-wheels/#rwsection</a>
VRW-A-1	1.9	0.09	<a href="https://www.vectronic-aerospace.com/reaction-wheels/#rwsection">https://www.vectronic-aerospace.com/reaction-wheels/#rwsection</a>
OCE-RW150	7	0.15	<a href="https://satsearch.co/products/oce-technology-rw150-reaction-wheel-15nms">https://satsearch.co/products/oce-technology-rw150-reaction-wheel-15nms</a>
OCE-RW250	10	0.2	<a href="https://satsearch.co/products/oce-technology-rw250-reaction-wheel-25nms">https://satsearch.co/products/oce-technology-rw250-reaction-wheel-25nms</a>
OCE-RW1000	11.5	1	<a href="https://www.satcatalog.com/component/rw1000/">https://www.satcatalog.com/component/rw1000/</a>
RW400 AAC Clyde Space	0.155	0.012	<a href="https://satsearch.co/products/aac-clyde-rw400-cubesat-reaction-wheels#specs">https://satsearch.co/products/aac-clyde-rw400-cubesat-reaction-wheels#specs</a>
RW-10NMs Spaceteq	5	0.190	<a href="https://satsearch.co/products/spaceteq-reaction-wheel-rw-10nms#specs">https://satsearch.co/products/spaceteq-reaction-wheel-rw-10nms#specs</a>
RWP500 Blue Canyon	0.75	0.025	<a href="https://satsearch.co/products/bluecanyontech-rwp500">https://satsearch.co/products/bluecanyontech-rwp500</a>
CubeWheel $S^+$	0.09	0.0023	<a href="https://satsearch.co/products/cubespace-cube-wheel-small-plus">https://satsearch.co/products/cubespace-cube-wheel-small-plus</a>
CubeWheel M	0.15	0.001	<a href="https://satsearch.co/products/cubespace-cube-wheel-medium">https://satsearch.co/products/cubespace-cube-wheel-medium</a>
CubeWheel L	0.225	0.0023	<a href="https://satsearch.co/products/cubespace-cube-wheel-large">https://satsearch.co/products/cubespace-cube-wheel-large</a>