



POLITECNICO DI TORINO

Master degree course in Mechatronic Engineering

Master Degree Thesis

MPC Design for CMG based Testbed

Academic Supervisors

Prof. Elisa Capello
Prof. Satoshi Satoh

Candidate

Matteo FACCHINO
s255241

ACADEMIC YEAR 2019-2020

Abstract

The growing interest in space exploration is focusing the research in the development of space architectures able to be independent and self-sustainable. To this end, different control strategies have been studied for different types of spacecrafts. The following thesis has the goal to design a linear Model Predictive Control (MPC) approach for the attitude control of a spacecraft testbed equipped with 4 Control Moment Gyroscopes (CMGs) in pyramidal configuration. The CMGs are torque generators used for the attitude control of spacecrafts, and the principal problem in the large diffusion of these devices is that, in a certain condition, the CMGs cannot provide torque along a specific direction. This condition is called singularity, and it compromises the stability of the control system, therefore it must be avoided. The MPC strategy has been adopted because we can directly avoid the singularity condition implementing this control approach. Contrary to other control strategies, the MPC can directly calculate the values to apply to the motors of the CMGs, without calculating the desired value of the total output torque; in such a way, a singularity avoidance steering law is not necessary. In the following work have been performed different simulations investigating different MPC approaches with additional characteristics. After, the control algorithm has been converted in C++ programming language, and used to perform experiments on a testbed in pyramidal configuration located at the Yamada's laboratory at Osaka University.

Contents

List of Figures	III
List of Tables	V
1 Introduction	1
1.1 Spacecraft attitude control general problem	1
1.2 Thesis motivation and previous works	7
1.3 Thesis outline	8
2 Model Predictive Control description	9
2.1 General system definition and Classic control techniques	9
2.2 Introduction to Model Predictive Control	11
2.3 MPC with online optimization of Virtual Reference Signal and Con- troller state	16
2.3.1 Plant definition and constraints	17
2.3.2 Error System and cost function definition	19
2.3.3 Predictions and Constraints with the Error System	20
2.3.4 LMI problem for terminal control law	21
2.3.5 MPC algorithm with integrator resets	22
2.4 Introduction to Stochastic MPC	24
2.4.1 Explication of matrix A with uncertainties	25
2.4.2 Explication of matrix B with uncertainties	26
3 Spacecraft Dynamics and Model of CMG-based testbed	29
3.1 Reference Frames	30
3.1.1 Inertial Reference Frame	30
3.1.2 Body Reference Frame	31
3.2 Introduction to Dynamics	31
3.3 Spacecraft Dynamic equation of motion	34
3.4 Spacecraft Kinematic equation of motion	36
3.5 Mathematical Model of the system	36
3.6 Control Moment Gyroscopes description	37

4	Simulations in Matlab environment	42
4.1	Optimization function	42
4.1.1	Virtual reference term of the cost function	43
4.2	MPC Implementation	44
4.2.1	Classic MPC implementation	44
4.2.2	Mode 1 implementation	45
4.2.3	Mode 2 implementation	47
4.3	Parameters used in the simulation	48
4.4	Simulation Results	49
4.4.1	Classic MPC results	49
4.4.2	MPC with Mode 2 results	53
4.4.3	MPC with Mode 1 and 2 results	53
5	Testebed characteristics and Experimental results	60
5.1	Experimental setup	60
5.2	Experimental results and C++ conversion	63
5.2.1	Conversion of the control algorithm in C++	63
5.2.2	Experimental results	63
6	Conclusions and possible future works	67
	Bibliography	69

List of Figures

1.1	ARVD manoeuvre [1]	2
1.2	Reaction Wheel [2]	3
1.3	CMG platform (credit: Gyroscope.com)	4
1.4	CMGs are used for the attitude control of the International Space Station (credit: NASA)	4
2.1	Block diagram representation of a generic system with input and output signals [3]	9
2.2	Block diagram representation of an open loop control system [3] . .	10
2.3	Block diagram representation of a closed loop control system [3] . .	10
2.4	MPC is like playing chess [4]	12
2.5	Block diagram of an MPC control [4]	12
2.6	Receding Horizon concept for MPC (credit: Wikipedia)	13
3.1	Earth-Centered Inertial Reference frame [5]	30
3.2	Body Reference Frame for a spacecraft with pyramidal CMG system [6]	31
3.3	CMG with a single gimbal [7]	38
3.4	Physical vectorial quantities representation [7]	39
3.5	Testbed equipped with four CMGs in pyramidal configuration . . .	40
4.1	Vectorial part of the quaternion of the classic MPC	50
4.2	Angular velocity components of the classic MPC	50
4.3	Input signals of the classic MPC	51
4.4	Acceleration of the input signals of the classic MPC	51
4.5	Gimbals evolution of the classic MPC	52
4.6	Quaternion components of the classic MPC	52
4.7	Vectorial part of the quaternion of the MPC with Mode 2	53
4.8	Angular velocity components of the MPC with Mode 2	54
4.9	Input signals of the MPC with Mode 2	54
4.10	Acceleration of the input signals of the MPC with Mode 2	55
4.11	Gimbals evolution of the MPC with Mode 2	55

4.12	Quaternion components of the MPC with Mode 2	56
4.13	Vectorial part of the quaternion of the MPC with Mode 1 and 2 . .	56
4.14	Angular velocity components of the MPC with Mode 1 and 2	57
4.15	Input signals of the MPC with Mode 1 and 2	57
4.16	Acceleration of the input signals of the MPC with Mode 1 and 2 . .	58
4.17	Gimbals evolution of the MPC with Mode 1 and 2	58
4.18	Quaternion components of the MPC with Mode 1 and 2	59
5.1	Gravity effect on the testbed [22]	61
5.2	Experimental setup located at Osaka University [22]	61
5.3	Experimental setup located at Osaka University	62
5.4	Vectorial part of the quaternion obtained from the experiment . . .	64
5.5	Angular velocity components of the testbed obtained from the ex- periment	65
5.6	Control input signals obtained from the experiment	65
5.7	Gimbal angles evolution obtained from the experiment	66
5.8	Quaternion components obtained from the experiment	66

List of Tables

4.1	Parameters used for the simulations	49
-----	---	----

Chapter 1

Introduction

1.1 Spacecraft attitude control general problem

What is the meaning of space exploration? EASAC has provided the following definition as reported in [8]: "Space exploration represents the extension of human research beyond the Earth's atmosphere using spacecraft to access unknown terrains and environments, and to acquire knowledge about space, planets, stars, or other celestial bodies by human and robotic means."

The space exploration began in the 1950s, when United States and Soviet Union launched the first spacecrafts. After a short time Europe and Japan started to sent their first small satellites in the space too. From that moment the space exploration has changed a lot with the parallel development of technologies and techniques.

Nowadays, the interest in space field is continuously growing. There are different motivations for this interest in space: first, the scientific reason to try to understand always more the Solar System and the Universe in general; but the space exploration reflects also on societal and cultural aspects. It strives to develop the scientific research, setting new goals and it raises the interest of people towards scientific careers and studies.

Nasa has highlighted as "Robotics, Tele-Robotics, and Autonomous System" shall be considered as high-priority fields of study in consolidating and broadening access to space and incrementing the human presence in the Solar System [1] [9]. The major space agencies all agree about this opinion: the advancement of the space missions must be performed creating space architectures able to be totally independent and self-sustainable, hence not more controlled by the Earth. One of the main problems ever encountered in space missions is the Autonomous Rendezvous and Docking (ARVD) manoeuvre among different systems in space. This problem must be treated with always more sophisticated techniques, because with the growing complexity of the space missions, the ARVD manoeuvres must be treated as better as possible. With the advancement of the autonomous systems and control techniques, space architectures always more able to deal with all the operational

and safety constraints of these ARVD manoeuvres in an independent way will be created. In this way, these space architectures will be able to support the crew in the space and will be also possible to perform these rendezvous manoeuvres without a crew on the system, considerably relaxing the safety constraints for these operations.

In ARVD manoeuvres there is a chaser, the system who leads towards, and the target that is the system that must be achieved. One of the main aspect of these operations, is that the two involved systems have a specific docking point through which they must be attached. The attitude control of these space systems to perform ARVD manoeuvres is fundamental for the correct orientation of the two system one respect to the other, according to the docking points. But in general a good attitude control of spacecrafts is the starting point to perform a lot of different operations, and hence to obtain the success of the space mission.

In Figure 1.1 is reported an ARVD manoeuvre where a chaser (craft on the right) tries to achieve the target (craft on the left). This picture presents a V-bar approach typically considered in these operations.

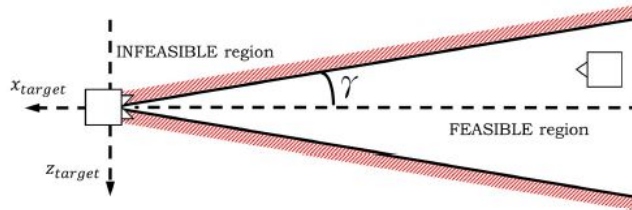


Figure 1.1: ARVD manoeuvre [1]

In general the attitude control can be defined as the process aimed to stabilize and adjust the orientation of spacecrafts during space manoeuvres [10]. Nowadays, space missions for satellites require a capability to rapidly change the attitude with high slew rate, both to ensure the success of the mission and to avoid catastrophic impacts among crafts [11]. The environment in which the attitude control of the spacecrafts must be performed, is particularly challenging. Indeed, spacecrafts control is not like aircraft control, in which ailerons, elevators and rudders are used for the attitude control exploiting aerodynamic forces. In the vacuum space, these forces are absent, hence these actuators are useless. For the spacecraft, devices able to generate torques from the variation of the angular momentum are necessary. In this way, through the spinning of the angular momentum vector of these actuators, is possible to control the attitude of the craft.

There exist passive and active control strategies. However, the active ones ensure more precision and reliability, therefore the second ones are preferred for the attitude control of spacecrafts. The most common active control devices for satellites

are the Reaction Wheels, the Control Moment Gyros (CMGs), Magnetic Torque Rods, and Thrusters. Among them, the main devices used in space applications are the Reaction Wheels and the CMGs, due to their high accuracy. The Reaction Wheels are devices that use torque to spin up a wheel; in this way, an equal and opposite amount of torque is imparted to the spacecraft (Figure 1.2). In this devices, the angular velocity of the wheel (the rotor) may change according to the desired attitude.



Figure 1.2: Reaction Wheel [2]

The CMGs have in general a fixed rate of rotation, thus the magnitude of the angular momentum vector is fixed too. A CMG is composed by a wheel and from one or two motorized gimbals that tilt the rotor's angular momentum; the creation of the gyroscopic torque is performed varying the direction of the angular momentum vector through the gimbals. A CMG system is composed by a group of 3 or 4 CMGs in order to ensure the three-dimensional control, and the configuration in which the CMGs are allocated is very important for the definition of the model of the system (Figure 1.3). In general CMGs are preferred because they are able to generate larger output torque compared to the Reaction Wheels. The generated output torque is much greater than the input torque required to drive the gimbal motor; this property is known as torque amplification of CMGs.

In Figure 1.4 is represented the International Space Station, that uses CMGs for its attitude control.

Though CMG systems can provide rapid slew capability and high pointing accuracy, not using any of the limited propellant dedicated to the main propulsion system, an important difficulty is their inherent geometric singularity problem, regardless of how many CMGs are equipped. In certain configuration of the gimbal angles, it is not possible to generate torque along a specific direction. In the last years, different methodologies have been realized to avoid the singularity condition for the CMGs implementation, each one with its strong and weak aspects.

A possible solution is the implementation of a steering law that, once received the

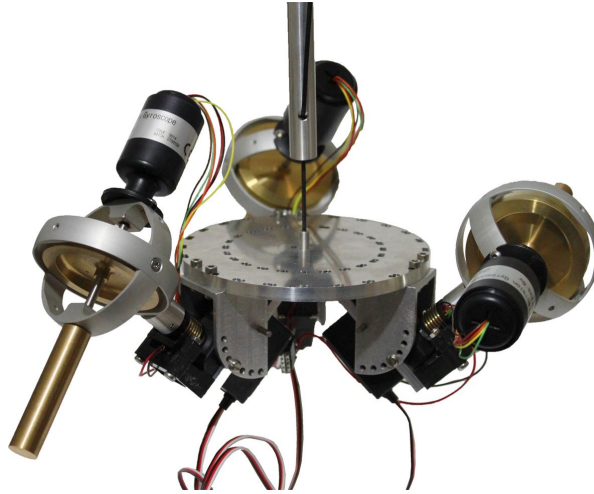


Figure 1.3: CMG platform (credit: Gyroscope.com)

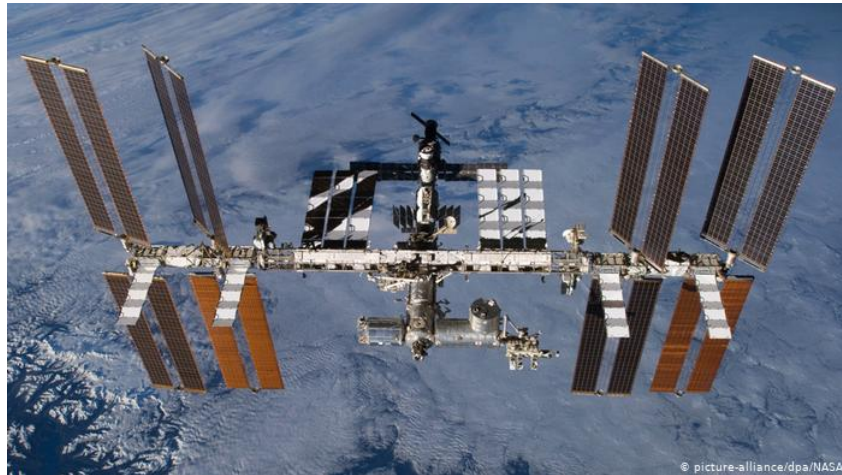


Figure 1.4: CMGs are used for the attitude control of the International Space Station (credit: NASA)

desired output value of the torque from the controller, is able to calculate the value of the gimbal rates in order to avoid the singularity condition.

In paper [12], is presented a configuration with CMGs and steering scheme for rapid maneuvering of Earth observation satellites. The CMG singularity arises when the torque vectors of all the CMGs are aligned on a plane, so that the CMG system cannot produce a control torque in the normal direction of that plane. In this paper, the CMG steering problem is solved using two planar systems in a rooftop CMG configuration. The component of momentum perpendicular to gimbal directions to be shared between the two pair of actuators is used to avoid the singularities or minimize the control degradation.

In [2], different configuration of CMGs systems are compared, utilizing a single PID controller with a steering law based on the Moore-Penrose inverse matrix to avoid the singularity condition. The analysis is particularly interesting for the comparison among different actuator system: the first two are CMG systems in pyramidal configuration with a different skew angles, the third system is a cluster of two CMGs and four Reaction Wheels.

Even if PI and PID controllers are largely used in industrial applications, they are used in simple cases where the plant has not a complex dynamics and where an appropriate proportional and damping gains are able to ensure a stable behaviour of the control system. However, if the system to be controlled is more complex, maybe because more variables in the same time have to be calculated by the controller, the implementation of a optimal control strategy could be better. The optimal control solution has the particularity simple form of linear state feedback for the case of linear systems and quadratic cost functions, and the feedback gains can be computed by solving an equation known as the steady-state Riccati equation. The most successful methodology among the optimal control environment is the Model Predictive Control.

The Model Predictive Control (MPC) uses a dynamic model of the system to be controlled, to predict its future behaviour and choose the best control input to apply to the plant. The MPC computes a sequence of control inputs, but unlike general optimal control, this sequence is recomputed at each sampling time after having measured/estimated the current state of the system. This brings the MPC more suitable for systems that operates in environments with disturbances. The MPC has received a great interest in the last decades for its great compromise between computational speed and accuracy, due to its receding horizon implementation for the system prediction. Particularly interesting is the Non linear MPC (NMPC) that is able to deal with non linear systems and constraints. This particular MPC has received less attention in the past decades with respect to the Linear MPC because it was used for relatively slow dynamic processes which leave time for intense computational control algorithms. However, with the development of the optimization control functions, NMPC is getting more and more considered. In paper [13], the problem of attitude control of spacecraft using CMGs in pyramidal configuration is faced with the implementation of a method to avoid the singularity condition by means of NMPC. Into the cost function that must be minimized, is introduced a singularity avoidance term, in order to generate a control input that moves away from the singularity when the CMG system approaches the neighbourhood of singularity, and to rapidly converge to a desired attitude when the CMG system stays away from the singularity. Even if the implementation of an NMPC is treated, this paper still uses a steering law to calculate the value of the gimbal angular velocities. But a great advantage of using an MPC strategy is the

direct calculation of the gimbal rates by the controller, without previously found out the required output torque value for the spacecraft.

As said, MPC is more effective with respect to general optimal control strategies to deal with disturbances and uncertainties. However, if the problem requires a certain level of robustness against the parameter uncertainties in the system model and against additive disturbances coming from the environment, it could be necessary the implementation of more sophisticated control strategies. The Robust MPC (RMPC) is a particular type of MPC able to reject noise; in particular is assumed that are known just the bounds of the disturbances, but no other information is known. The RMPC is able to guarantee state constraint satisfaction, feasibility and closed-loop stability for any value of the disturbances inside the bounds.

In certain condition, the uncertainties and the disturbances coming from the environment can be modelled as random noises but with a probabilistic distribution. This control problem is called Stochastic MPC (SMPC) and its objective is to ensure that all the constraints are met and the closed-loop stability guaranteed.

The paper [14] is concerned with the inclusion of integral action in an LMI based RMPC formulation for the purpose of ensuring offset-free regulation. In particular, the paper proposes a novel integrator resetting scheme aimed at improving the transient response of the closed-loop system. The resetting procedure is designed to retain the recursive feasibility and asymptotic stability properties of the RMPC formulation.

In paper [15], a sample based SMPC for tracking control problem of fixed-wing UAV is presented. The proposed control exploits recent results on sample-based stochastic MPC, which allow coping in a computationally efficient way with both parametric uncertainties and additive random disturbances. In this essay different scenarios have been studied and the effectiveness of the control strategy is shown by the experimental results.

The implementation of a similar SMPC algorithm for discrete-time linear systems subject to parametric uncertainties and additive disturbance is proposed in [1]. In this paper the control algorithm has been developed and validated for a floating spacecraft experimental testbed, proving that this solution is implementable in real-time. The considered SMPC provides reliability and robustness for the ARVD operations performed by the considered testbed, even if parametric uncertainties and disturbances due to external space environment are simultaneously considered. This paper has developed an off-line sample-based strategy for addressing in a computationally tractable manner SMPC. In the chapter dedicated to the description of the MPC, a section will be devoted to the initial operations that must be performed in order to implement the off-line computation of this SMPC algorithm. Indeed, as will be motivated after, a possible future work related to the following thesis essay, can be the implementation of an SMPC strategy.

In the last years, the problem of feasibility of the on-line optimization and stability of the closed loop system have been extensively studied. The dual-mode MPC

strategy is accepted as one of the most systematic approaches to design an MPC that guarantees stability and feasibility. In this strategy, the open loop optimal control is initially applied, then a terminal control law is utilized after the state variable reaches a positively invariant set. Nowadays, the MPC is particular interesting for tracking control problems; and for these problems a standard way to achieve the goal is the insertion of an integrator into the feedback loop in order to achieve set-point tracking. Inserting the integrator, the overall system on which must be designed the MPC approach is the augmented system that consists of the plant and of the integrator. It would be possible to reset, in certain sampling times, the value of the integrator state in order to improve the tracking performance. In paper [16] is proposed a Linear MPC algorithm that enhance the tracking control performance by introducing a controller initialization technique. This controller is assumed to have an integrator to achieve set-point tracking. By the optimization procedure at each sampling time the open loop optimal control sequence and the integrator state are obtained simultaneously so that a cost function is minimized until the value of the cost function becomes smaller than a threshold value. After it, the integrator state is updated according to the integrator equations in order to achieve zero steady state error.

1.2 Thesis motivation and previous works

The following thesis had the objective to design a linear Model Predictive Control (MPC) strategy for the attitude control of a testbed, which simulates a spacecraft, located at the Yamada's laboratory at Osaka University. The considered testbed is equipped with 4 Control Moment Gyroscopes (CMGs) disposed in pyramidal configuration.

The implementation of an MPC approach for the attitude of a spacecraft is motivated by the fact that an MPC strategy can directly calculate the desired angular velocities of the gimbals without previously calculating the desired output torque for the spacecraft. Furthermore, MPC approach can take into account in the optimization procedure the state and input constraints, optimizing the control input that must be applied to the plant.

For the development of this work, different paper have been studied. In particular, the following essay is a continuation of the work developed by a student of the Politecnico di Torino, exposed in [17], in which a comparison of the results obtained by the implementation of an MPC and of a PID control with Singular Direction Avoidance (SDA) steering law in simulation has been carried out. In this thesis project the same model of the testbed is used and explained in that work has been adopted.

The other important article for this work is [18], because the Linear MPC strategy developed in that paper is used in this thesis project. In this paper is exposed a

linear MPC approach where are introduced additional characteristics with respect to conventional MPC. In this paper is considered an MPC in which are optimized not just the sequence of control inputs to apply to the plant, but from the optimization procedure, are obtained also other variables: the virtual reference signal and the integrator state.

1.3 Thesis outline

Chapter 2 is concerned to explain the basic principles of the MPC. After, the MPC proposed in [18] is analysed, where are also implemented the optimization of the virtual reference signal and of the integrator state.

Chapter 3 deals the definition of the model used to simulate the behaviour of the testbed located at the Yamada's laboratory at Osaka University. Starting from the Euler's equation of rotation and from a quaternion representation for the attitude description is defined a Linear Time-Invariant state space model of the system.

Chapter 4 is the one dedicated to the simulations in Matlab environment. Three different types of MPC have been implemented: the first simulation has been performed with a classic MPC; the second one with an MPC in which also the virtual reference signal optimization is implemented; finally, the third one has been performed with an MPC in which both the virtual reference and the integrator state are optimized. The results of each of these simulations have been reported and compared among them, highlighting the benefits brought by the additional characteristics.

Chapter 5 is dedicated to the experiments performed at the Yamada's laboratory at Osaka University. A description of the physical features of the considered testbed has been given, and the experimental results obtained have been reported showing the differences with respect to the simulation outcomes.

Chapter 6 is devoted to the conclusions of this thesis work.

Chapter 2

Model Predictive Control description

2.1 General system definition and Classic control techniques

In general, it's possible to define a system as a collection of elements that work together to obtain something. A system is composed by inputs (which go in), outputs (which go out) and by some process that collecting the inputs, is able to produce the outputs. In the world of the control theory, inputs and outputs are usually called signals, and the process between them is called plant. In the control field is usually utilized a type of representation called block diagram (Figure 2.1). In this type of representation the signals are pictured as arrows, while the processes are represented by rectangular boxes.

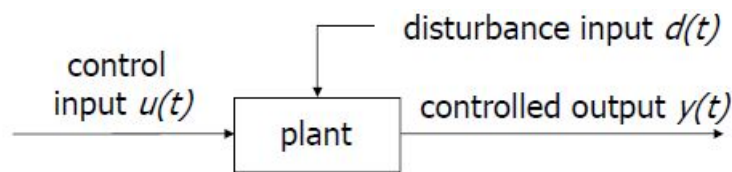


Figure 2.1: Block diagram representation of a generic system with input and output signals [3]

The plant alone is a so-called "equal-opportunity" process that provides an output, given either a random input from the environment or a precisely calculated input. In the space application, the success of the mission heavily relies on the ability to reach a specific output in a certain moment. In these applications a more complex system is needed; for these purposes are more interesting a specific class of systems, the control systems. Control systems are present in every aspect of

our life, from turn up the thermostat in the house to drive a car. In general, a control system can be defined as a system in which the input signals are precisely calculated in order to obtain a specific desired behaviour of the output signals.

In order to calculate the required input signals to obtain the desired outputs, a controller is needed. At this point is important also differentiate between two types of control systems: the open loop control systems and the closed loop control systems. The open loop control systems are systems in which is not possible to dynamically correct the value of the input based on what is actually happening (Figure 2.2). It is clear that this type of systems don't work properly in case in which from the environment come disturbances. The second type are the closed loop control systems, where sensors are used to calculate the actual values of the state and, exploiting this information, the value of the input signals to apply to the plant is recalculated (Figure 2.3). In space applications is clear that are needed the second type of control systems, because they provide more reliability for the success of the mission.

In general the principal elements which constitute a closed loop control system are: a controller, a desired reference signal, a plant, actuators (not represented in the images) and sensors.

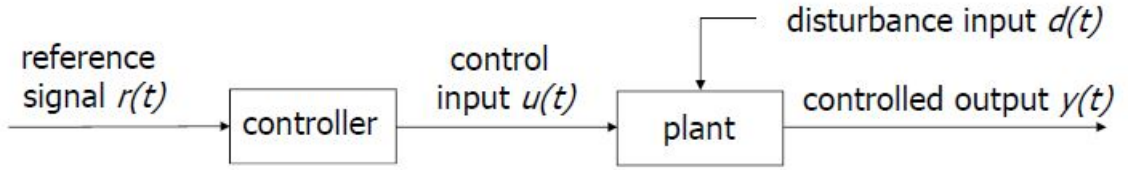


Figure 2.2: Block diagram representation of an open loop control system [3]

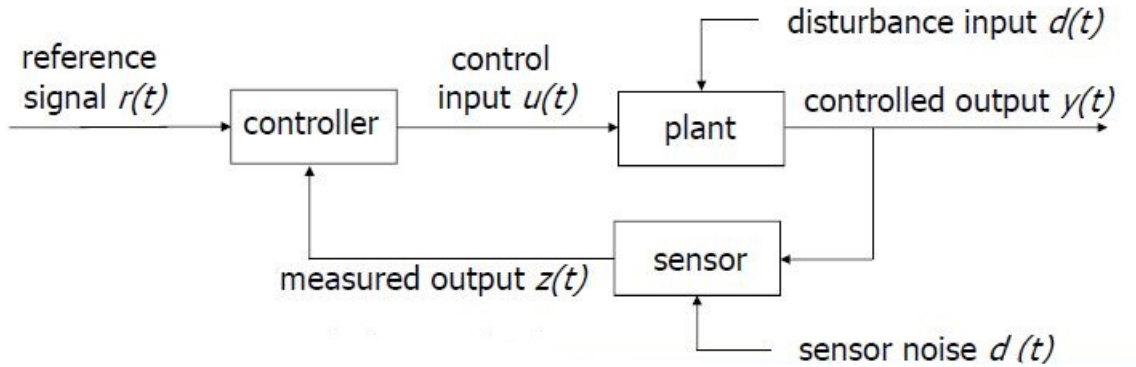


Figure 2.3: Block diagram representation of a closed loop control system [3]

The purpose of the control system in space applications is to obtain a good tracking behaviour of the actual outputs with respect to the desired output references. The controller's job is to generate commands to the actuator in order to

obtain this tracking behaviour of the actual outputs with respect to the references. The most common type of controller is the PID (Proportional, Integrative, Derivative) controller. It uses the difference between the reference state and the measured one, obtained from sensors, to calculate the best control input to apply to the plant, in order to minimize the error between the reference signals and the actual ones. This type of controllers is very useful for applications in which the system to be controlled is simple, and they can give a good result with the minimum effort. But if the problem is more complex, maybe because some input or state constraints have to be considered, or because we have to control more variables in the same time, the use of these control strategies can be inappropriate. Indeed, the tuning of the gain parameters to obtain a good behaviour can be tricky and these controllers can also fail to guarantee the stability of the control system. In PID controllers, the constraints are not considered in the input calculation, as consequence, input values are often saturated and in this case we cannot obtain an optimal control. Simple PID controllers can be also complicated to implement in control systems for space applications where is required to generate different commands for different actuators in the same time. Indeed, the different control systems of each actuator can be connected among them, consequently they can affect each other and all the PID controllers should be calibrated all together.

In this work has been implemented a control strategy called Model Predictive Control, which will be treated better in the next section. One of the great characteristics of this control strategy, is the ability to consider the constraints already in the control input computation; in this way the input signals are efficiently chosen. Utilising the MPC strategy is also easier calculate different commands for different actuators in the same time.

2.2 Introduction to Model Predictive Control

To understand better how the predictive control works, it's common to compare this strategy to the chess game (Figure 2.4). In the chess game, each player on his turn has to decide the best move to do trying to predict the future development of the game, anticipating the future moves of the opponent. If, at the next turn, the opponent reply in an unexpected way, the player has to replan his strategy, trying again to predict the future progress of the game. A good chess player is able to predict the development of the game for a lot of next turns forward in the time, considering different possible scenarios.

Similarly, the Model Predictive Control uses a dynamic model of the system that we want to control, in order to make a prediction of the future behaviour of the system itself and consequently "chooses" the best control input to apply to the plant. It's easy to understand that the definition of the mathematical model of the system is a really important part for the success of this control strategy. From one hand, the model has to be descriptive enough to capture all the most important

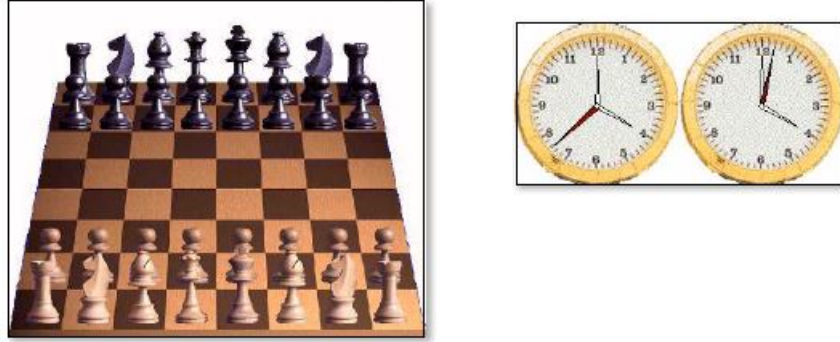


Figure 2.4: MPC is like playing chess [4]

characteristics of the system, in order to make predictions as much as possible near to the real evolution of the process. But on the other hand, if the model is too much descriptive, the complexity is increased too, and the result is that the control system has a high computational cost, that can be processed just by expensive hardware devices [4] [19]. This was the principal problem for the development of this control strategy some decades ago. Indeed, at the beginning of its progress, it was used just for industrial and chemical processes, systems in which the dynamic can be really slow. In Figure 2.5 is reported the block diagram of a generic MPC control.

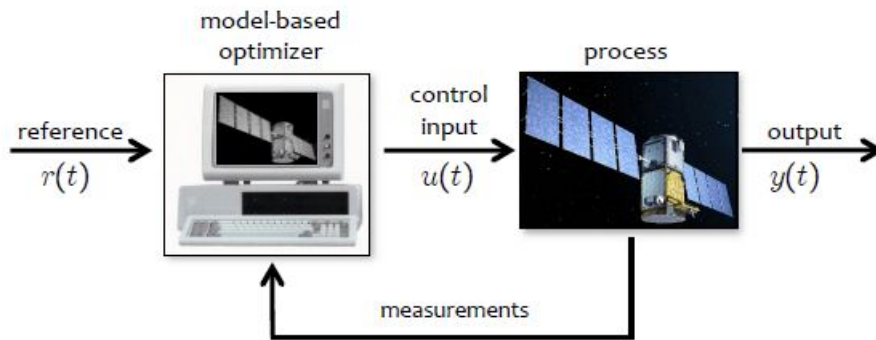


Figure 2.5: Block diagram of an MPC control [4]

At each sample time, the controller computes the sequence of control inputs for the actual and next sample times in open loop fashion, but just the first one of this sequence is applied to the plant. At the next sample time the procedure is repeated from the beginning, without considering the other control input values obtained in the previous sample times.

The number of control inputs collected in the control sequence that are obtained by

the controller at each sampling time, depend on the capability of the controller itself to "look forward" in the future. This characteristic is defined by a parameter called Receding Horizon, and is a design parameter that is possible to tune according to the performance expected by the specific MPC. The choice of this value must be done taking into account a good trade-off between the prediction capability of the controller and the computational cost we want to meet. In practise, this parameter has to be as high as possible, in order to improve the optimal control input obtained from the controller, but always respecting the stabilized computational limits.

In Figure 2.6 are reported the variables included in an MPC problem. From the picture is important to notice that the MPC solves the optimization problem exploiting the entire control sequence calculated at time k to reach the reference signal; but in reality, only the first value of the control sequence, circled in green, will be utilized.

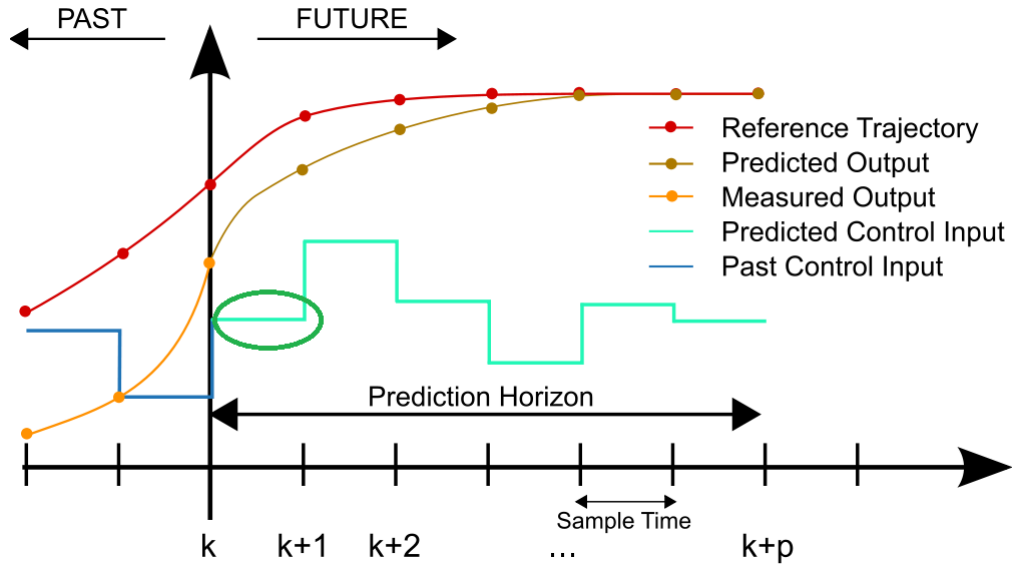


Figure 2.6: Receding Horizon concept for MPC (credit: Wikipedia)

In this work has been implemented a Linear MPC, so just this type of MPC will be treated. But in the literature there exist a lot of different types of MPC, such as Non Linear MPC, Linear Time-Varying MPC, Explicit MPC and the Stochastic MPC, the one on which the research is focusing on for its great characteristics to deal with disturbances and uncertainties.

Starting the formulation of the problem, the first thing to define is the Linear dynamic model of the system in state space formulation:

$$x(k+1|k) = Ax(k|k) + Bu(k|k) \quad (2.1)$$

$$y(k|k) = Cx(k|k) + Du(k|k) \quad (2.2)$$

The first equation is the state equation, where appear the state of the system $x \in \mathbb{R}^{n_p}$, and the input applied at time k , $u \in \mathbb{R}^m$. The second equation is the output equation, from which is obtained the value of the output $y \in \mathbb{R}^{n_y}$.

The matrices of the state space system have the following dimensions: $A \in \mathbb{R}^{n_p \times n_p}$, $B \in \mathbb{R}^{n_p \times m}$, $C \in \mathbb{R}^{n_y \times n_p}$ and $D \in \mathbb{R}^{n_y \times m}$.

The notation $x(k+1|k)$ indicates the value of the state x at time $k+1$ calculated at time k .

Using the defined model of the system is possible to obtain the prediction matrices that will be used to calculate the optimal sequence of control inputs.

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{H_s} \end{bmatrix} = \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{H_s-1}B & A^{H_s-2}B & \cdots & B \end{bmatrix}}_S \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{H_s-1} \end{bmatrix} + \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_s} \end{bmatrix}}_T x_0 \quad (2.3)$$

The nomenclature x_{H_s} is the abbreviation of the term $x(k+H_s|k)$; similarly, the variable x_0 represents the state $x(k|k)$.

The previous equation is composed by two terms: the first one collects all the future inputs that compose the input sequence U , obtained by the optimization procedure; this sequence is multiplied by a matrix called S . The second term is defined by the multiplication of a matrix T by the state x_0 , which is the only variable known before the optimization procedure, because it is the actual value of the state obtained by the sensors.

The objective of the controller is to find the optimal sequence of control inputs. For this reason, each sample time, a convex optimization problem must be solved. In particular is defined a Quadratic Program in which a cost function must be minimized. The cost function is defined at time k :

$$J(k) = x'_{H_s} P x_{H_s} + \sum_{k=0}^{H_s-1} x'_k Q x_k + u'_k R u_k \quad (2.4)$$

Where:

- H_s is the prediction horizon;
- $x_0 = x(k|k)$ is the initial state;
- $U = U(k)$ is the control sequence;
- Q, R and P are the weighting matrices and are defined as follows:

$$R = R' > 0$$

$$Q = Q' \geq 0$$

$$P = P' \geq 0$$

The goal is to find the sequence $U^* = \begin{bmatrix} u_0^* \\ u_1^* \\ \vdots \\ u_{H_s-1}^* \end{bmatrix}$ that steers the output to the

desired reference "optimally".

The cost function can be rewritten as:

$$\begin{aligned} J(k) = x_0' Q x_0 + & \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{H_s-1} \\ x_{H_s} \end{bmatrix}' \underbrace{\begin{bmatrix} Q & 0 & 0 & \cdots & 0 \\ 0 & Q & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & Q & 0 \\ 0 & 0 & \cdots & 0 & P \end{bmatrix}}_{\bar{Q}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{H_s-1} \\ x_{H_s} \end{bmatrix} \\ & + \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{H_s-1} \end{bmatrix}' \underbrace{\begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & R \end{bmatrix}}_{\bar{R}} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{H_s-1} \end{bmatrix} \end{aligned} \quad (2.5)$$

It is important to notice the presence of the P matrix as last term in \bar{Q} . This matrix is the terminal controller, and its presence ensures feasibility and stability to the MPC. Combining the prediction equation (2.3) and the previous one (2.5) we obtain the following final version of the cost function:

$$\begin{aligned} J(k) &= x_0' Q x_0 + (SU + Tx_0)' \bar{Q} (SU + Tx_0) + U' \bar{R} U \\ &= \frac{1}{2} U' \underbrace{2(\bar{R} + S' \bar{Q} S)}_H U + x_0' \underbrace{2T' \bar{Q} S}_F U + \frac{1}{2} x_0' \underbrace{2(Q + T' \bar{Q} T)}_Y x_0 \end{aligned} \quad (2.6)$$

One of the great features of the MPC is the fact that, in the optimization problem is possible to include input and state constraints. Moreover, it's possible to demonstrate that the previous formulation of a quadratic program without constraints, it's nothing else than a linear state feedback law.

The input and state constraints can be defined:

$$u_{min} \leq u_k \leq u_{max} \quad k = 0, \dots, H_s - 1 \quad (2.7)$$

$$x_{min} \leq x_k \leq x_{max} \quad k = 1, \dots, H_s \quad (2.8)$$

Finally we can define the constrained optimal control problem (Quadratic Programming):

$$\begin{aligned} \min_U & x'_{H_s} P x_{H_s} + \sum_{k=0}^{H_s-1} x'_k Q x_k + u'_k R u_k \\ & \text{subject to (2.7), (2.8) and (2.3)} \end{aligned} \quad (2.9)$$

The algorithm of the Linear MPC is, starting at time k :

1. Measure the current state $x_k = x(k|k)$;
2. Solve the Quadratic problem:
From which we obtain the solution: $U(k)^* = [u^*(k|k)^T; \dots u^*(k + H_s - 1|k)^T]^T$;
3. Apply to the plant the first input of the sequence $u(k) = u^*(k|k)$ and eliminate the remaining part of the control sequence obtained;
4. Set $k + 1 \rightarrow k$ and repeat from the beginning.

In the following thesis project an MPC has been designed for the attitude control of a testbed equipped with 4 CMGs. The designed MPC is a Linear MPC but with additional characteristics. Indeed, each sample time, from the optimization problem, it's not just obtained the control sequence U , but also other variables that help to obtain a better tracking performance of the system. From the Quadratic Programming problem is obtained also the virtual reference signal, and the integrator state. The virtual reference signal is a variable that replaces the real reference of the system. It has to be as much as possible equal to the real one, but in certain moments it can deviate from the real one in order to guarantee the satisfaction of the input constraints. The second variable, the integrator state, it can be or not optimized at each sample time. So at each sampling time the controller has to decide if this variable will be or not optimized.

2.3 MPC with online optimization of Virtual Reference Signal and Controller state

In the previous section the basic Linear MPC has been described, in which from the optimization procedure, the optimal value of the control input is the only variable obtained. In this thesis project a more sophisticated Linear MPC has been used to control the attitude of the testbed. In this section is presented the MPC described in paper [18]. In this control algorithm from the optimization procedure is obtained

the sequence of the control inputs, the actual value of the virtual reference signal and can be also optimized the value of the integrator state.

This two new optimized variables bring benefits to the control problem. Indeed, the virtual reference signal can be seen as a variable that substitutes the real reference signal in the control algorithm. The virtual reference has to be as much as possible equal to the real reference but in some occasions it can assume different values with respect to the real reference. In certain conditions it could be difficult or maybe even impossible for the system follow the real reference. In this critical situation the virtual reference deviates from the original one, in order to provide to the control system an easier reference to achieve, and thus trying to improve the general tracking behaviour of the control system.

The use of the virtual reference is particularly useful in control problems in which are included input constraints. Indeed, if the output of the system is too far from the real reference, a too high control input could be required; in this situation the virtual reference is optimized, deviating from the real reference signal, taking into account the input constraints.

The other variable is the controller state. As will be described later, the paper considers to include into the feedback system an integrator as a servo compensator, in order to eliminate the steady-state error. The controller has to decide at each sample time if the integrator state x_c must be optimized or not. If x_c is not optimized, it evolves according to the integrator equations in order to guarantee the steady-state error elimination; instead, if x_c is included in the optimization problem, the optimized value of this variable gives a rapid decrement of the cost function value.

2.3.1 Plant definition and constraints

The system is described in the state space formulation as follows [16]:

$$x_p(k+i+1|k) = A_p x_p(k+i|k) + B_p u(k+i|k) \quad (2.10)$$

$$y(k+i|k) = C_p x_p(k+i|k) + D_p u(k+i|k) \quad (2.11)$$

Where $x_p \in \mathbb{R}^{n_p}$ is the state of the plant, and the notation $x_p(k+i|k)$ indicates the plant state at time $k+i$ calculated at time k .

In the considered paper the matrix D_p is considered null.

In the formulation of this MPC, both input and state constraints are considered. The input constraints are defined as:

$$|u(k+i|k)| \leq \bar{u}, \quad \forall i \geq 0 \quad (2.12)$$

Which becomes:

$$-\bar{u} \leq u(k+i|k) \leq \bar{u}, \quad \forall i \geq 0 \quad (2.13)$$

Where the maximum value $\bar{u} \in \mathbb{R}^m$.

For the implementation of these constraints in the MPC problem, it is more suitable a matrix formulation:

$$\Psi_u u(k+i|k) \leq \theta_u, \quad \forall i \geq 0 \quad (2.14)$$

Where the matrices $\Psi_u \in \mathbb{R}^{2m \times m}$ and $\theta_u \in \mathbb{R}^{2m}$.

For example, if we consider an input u with dimensions $m = 3$, the matrices become:

$$\Psi_u = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\theta_u = \begin{bmatrix} \bar{u}_{(1)} \\ \bar{u}_{(1)} \\ \bar{u}_{(2)} \\ \bar{u}_{(2)} \\ \bar{u}_{(3)} \\ \bar{u}_{(3)} \end{bmatrix}$$

In order to obtain the inequality:

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} u_{(1)}(k+i|k) \\ u_{(2)}(k+i|k) \\ u_{(3)}(k+i|k) \end{bmatrix} \leq \begin{bmatrix} \bar{u}_{(1)} \\ \bar{u}_{(1)} \\ \bar{u}_{(2)} \\ \bar{u}_{(2)} \\ \bar{u}_{(3)} \\ \bar{u}_{(3)} \end{bmatrix}, \quad \forall i \geq 0$$

The state constraints are defined:

$$|x_p(k+i+1|k)| \leq \bar{x}_p, \quad \forall i \geq 0 \quad (2.15)$$

Similarly to the input constraints, also in this case is possible to derive a matrix formulation:

$$\Psi_{x_p} x_p(k+i+1|k) \leq \theta_x, \quad \forall i \geq 0 \quad (2.16)$$

Where $\theta_x \in \mathbb{R}^{2n_p}$ and $\Psi_{x_p} \in \mathbb{R}^{2n_p \times n_p}$.

2.3.2 Error System and cost function definition

As said previously, the studied paper inserts an integrator into the control system to obtain the steady-state error elimination. The equations of the integrator are:

$$x_c(k+i+1|k) = x_c(k+i|k) + e(k+i|k) \quad (2.17)$$

$$e(k+i|k) = w(k|k) - y(k+i|k) \quad (2.18)$$

Where $x_c \in \mathbb{R}^{n_c}$ is the controller state, $w \in \mathbb{R}^{n_y}$ is the virtual reference signal and $e \in \mathbb{R}^{n_y}$ is the error obtained by the difference between w and y .

Combining the equations of the plant and of the integrator it is possible to obtain:

$$x(k+i+1|k) = Ax(k+i|k) + Bu(k+i|k) + Ew(k|k) \quad (2.19)$$

$$e(k+i|k) = Cx(k+i|k) + D_w w(k|k) \quad (2.20)$$

Where $x = [x_p^T, x_c^T]^T$, $C = [-C_p, 0]$, $D_w = I$,

$$A = \begin{bmatrix} A_p & 0 \\ -C_p & I \end{bmatrix}, \quad B = \begin{bmatrix} B_p \\ 0 \end{bmatrix}, \quad E = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

The state constraints can be rewritten as:

$$\Psi_x x(k+i+1|k) \leq \theta_x, \quad \forall i \geq 0 \quad (2.21)$$

Where $\Psi_x = [\Psi_{x_p}, 0]$.

At this point the paper made the following assumptions:

- The first one is that there exist matrices $\Pi \in \mathbb{R}^{(n_p+n_c) \times n_y}$ and $\Gamma \in \mathbb{R}^{m \times n_y}$ that satisfy the following equations:

$$\Pi = A\Pi + B\Gamma + E$$

$$0 = C\Pi + D_w$$

- the second useful assumption is that the steady-state value of the reference signal \bar{r} satisfies $\Psi_u \Gamma \bar{r} \leq \theta_u$ and $\Psi_x \Pi \bar{r} \leq \theta_x$.

The above assumptions ensure respectively the resolvability of the tracking control problem in case of linear systems, and that the control input u and the state x satisfy the constraints (2.14) and (2.21). The proof for the previous assumptions is called in the studied paper.

Using the previous assumptions is possible to obtain the values of Π and Γ , and with these matrices it's constructed the following error system:

$$\xi(k+i+1|k) = A\xi(k+i|k) + B\nu(k+i|k) \quad (2.22)$$

$$e(k+i|k) = C\xi(k+i|k) \quad (2.23)$$

Where:

$$\xi(k+i|k) = x(k+i|k) - \Pi w(k|k) \quad (2.24)$$

$$\nu(k+i|k) = u(k+i|k) - \Gamma w(k|k) \quad (2.25)$$

The obtained error system will be used to construct the MPC problem, hence from the optimization procedure will be obtained the value of ν , and not the value of u .

In the considered paper, the following cost function is implemented:

$$\begin{aligned} J(k) = & \sum_{i=0}^{H_s-1} \{ \|\xi(k+i+1|k)\|_Q^2 + \|\nu(k+i|k)\|_R^2 \} + \\ & + \|\xi(k+H_s|k)\|_P^2 + \|\Pi(w(k|k) - r(k|k))\|_M^2 \end{aligned} \quad (2.26)$$

Where Q , R , M , and P are positive-definite matrices. The first three matrices are design matrices, instead the matrix P is found solving an LMI problem, as will be explained later. H_s is the prediction horizon.

The novelty in the defined cost function is found out in the last term, $\|\Pi(w(k|k) - r(k|k))\|_M^2$, in which appears the Virtual Reference Signal, in order to include this new variable in the optimization. This term is optimized if $w(k|k) = r(k|k)$, that is what we want, but as already said in certain condition can be useful to have a small difference between the two variables, to increase the general tracking performance of the control system.

It's important to notice that the controller state x_c , that is the other variable that can be included in the optimization procedure, appears in the cost function in the terms associated to ξ .

2.3.3 Predictions and Constraints with the Error System

The state predictions of the error system up to the horizon H_s is:

$$\tilde{\xi} = T\xi(k|k) + S\tilde{\nu} \quad (2.27)$$

Where:

$$\tilde{\xi} = \begin{bmatrix} \xi(k+1|k) \\ \vdots \\ \xi(k+H_s|k) \end{bmatrix} \quad \text{and} \quad \tilde{\nu} = \begin{bmatrix} \nu(k|k) \\ \vdots \\ \nu(k+H_s-1|k) \end{bmatrix}$$

$\tilde{\xi}$ is the vector of the future state, instead $\tilde{\nu}$ is the optimized vector of the error system inputs, obtained by the optimization procedure.

The matrices T and S have been defined previously in the equation (2.3) and are constructed using the matrices A and B of the error system.

Now it is necessary to rewrite the constraints in function of the variables used in the error system. The input constraints must be rewritten, during $0 \leq i \leq H_s - 1$:

$$\tilde{\Psi}_u(\tilde{\nu} + \tilde{\Gamma}w(k|k)) \leq \tilde{\theta}_u \quad (2.28)$$

Where $\tilde{\Psi}_u = \text{block} - \text{diag}[\Psi_u, \dots, \Psi_u]$, $\tilde{\theta}_u = [\theta_u^T, \dots, \theta_u^T]^T$ and $\tilde{\Gamma} = [\Gamma^T, \dots, \Gamma^T]^T$.

Where the notation **block – diag** defines a diagonal matrix composed by the elements in the brackets.

The state constraints must be rewritten, during $1 \leq i \leq H_s$:

$$\tilde{\Psi}_x(T\xi(k|k) + S\tilde{\nu} + \tilde{\Pi}w(k|k)) \leq \tilde{\theta}_x \quad (2.29)$$

Where $\tilde{\Psi}_x = \text{block} - \text{diag}[\Psi_x, \dots, \Psi_x]$, $\tilde{\theta}_x = [\theta_x^T, \dots, \theta_x^T]^T$ and $\tilde{\Pi} = [\Pi^T, \dots, \Pi^T]^T$.

2.3.4 LMI problem for terminal control law

In the studied paper the dual-mode MPC is considered, because is generally accepted as a design approach that guarantees the feasibility. In the dual-mode strategy, the open loop optimization problem is applied, but then a terminal controller is used to reach a positive invariant set. This terminal controller consists in the derivation of a matrix P that substitute the last matrix Q in the optimization problem. This matrix is obtained through the resolution of an LMI problem. The calculation of P is performed off-line, bringing an important benefit in terms of real-time implementability.

The paper consider a theorem for the definition the LMI problem. In the following will be reported just the definition of the problem, without entering into the details. The reader interested in the proof of the following theorem can consult the paper [18].

The theorem that defines the LMI problem states that, considering the defined

error system, and for given positive-definite matrices Q, R and positive scalars η_2, τ , there exist a matrix Y and a positive-definite matrix \mathbb{S} that satisfy:

$$\begin{bmatrix} \mathbb{S} & * & * & * \\ A\mathbb{S} + BY & \mathbb{S} & * & * \\ Y & 0 & R^{-1} & * \\ A\mathbb{S} + BY & 0 & 0 & Q^{-1} \end{bmatrix} > 0 \quad (2.30)$$

$$\begin{bmatrix} \frac{1}{\eta_2} \mathbb{S} & * \\ Y^{(l)} & \rho^{(l)2} \end{bmatrix} \geq 0, \quad \forall l = 1, \dots, m \quad (2.31)$$

$$\begin{bmatrix} \tau \mathbb{S} & * & * \\ 0 & \theta_x^{(j)2} - \tau \eta_2 & * \\ \Psi_x^{(j)}(A\mathbb{S} + BY) & \Psi_x^{(j)} \Pi \nu_i & 1 \end{bmatrix} \geq 0, \quad \forall i = 1, \dots, n_s, \quad \forall j = 1, \dots, n_x \quad (2.32)$$

Where $P = \mathbb{S}^{-1}$.

It is very important to highlight that the resolution of the previous LMI problem and the consequently implementation of the P matrix in the MPC problem, ensure the closed-loop stability of control system.

Furthermore, another theorem states that, if the control algorithm is feasible at time k , the same algorithm is feasible for all $t > k$.

The closed-loop stability and feasibility conditions are demonstrated by theorems explained in [18].

2.3.5 MPC algorithm with integrator resets

In these section the control algorithm will be explained. As already said, in the considered MPC, the integrator state can be or not optimized. In order to choose if x_c has to be optimized or not, a threshold η_s must be defined: the integrator state x_c is reset at each sampling time so that the cost function value $J(k)$ is minimized while $J(k)$ is greater then η_s . The control algorithm with integral action is used to achieve zero steady-state error when $J(k)$ is less than or equal to η_s .

The algorithm is defined in this way:

- *Step 1* (Initialization): Set $k = 0$, η_s and all the design parameters, matrix P computation.
- *Step 2*: Measure the state $x_p(k|k)$.
- *Step 3*: If $J(k - 1) \leq \eta_s$, go to *Step 5*.
- *Step 4* (MPC with integrator reset) **Mode 1**:

$$\min_{\nu, w, x_c} J(k)$$

subject to

$$\begin{aligned}\tilde{\Psi}_u(\tilde{\nu} + \tilde{\Gamma}\tilde{w}) &\leq \tilde{\theta}_u \\ \tilde{\Psi}_x(T\tilde{\xi}(\tilde{x}_c, \tilde{w}) + S\tilde{\nu} + \tilde{\Pi}\tilde{w}) &\leq \tilde{\theta}_x \\ |\Gamma\tilde{w}| &\leq \theta_u - \rho\end{aligned}$$

Where $\tilde{\xi}(\tilde{x}_c, \tilde{w}) = [x_p(k|k)^T, \tilde{x}_c^T]^T - \Pi\tilde{w}$. Then set $x_c(k|k) = \tilde{x}_c$, $w(k|k) = \tilde{w}$ and go to *Step 6*.

- *Step 5* (MPC with integral action) **Mode 2**:

$$\begin{aligned}\min_{\tilde{\nu}, \tilde{w}} J(k) \\ \text{subject to} \\ \tilde{\Psi}_u(\tilde{\nu} + \tilde{\Gamma}\tilde{w}) &\leq \tilde{\theta}_u \\ \tilde{\Psi}_x(T\tilde{\xi}(x_c(k|k), \tilde{w}) + S\tilde{\nu} + \tilde{\Pi}\tilde{w}) &\leq \tilde{\theta}_x \\ |\Gamma\tilde{w}| &\leq \theta_u - \rho\end{aligned}$$

Where $\tilde{\xi}(x_c(k|k), \tilde{w}) = [x_p(k|k)^T, x_c(k|k)^T]^T - \Pi\tilde{w}$. Then set $w(k|k) = \tilde{w}$.

- *Step 6*: Apply $u(k|k) = \nu(k|k) + \Gamma w(k|k)$ to the plant system.
- *Step 7*: Compute $x_c(k+1|k)$ using the integrator equations.
- *Step 8*: Set $k+1 \rightarrow k$ and go to *Step 2*

In the algorithm appears a further constraint, $|\Gamma\tilde{w}| \leq \theta_u - \rho$, which limit the maximum amplitude of the virtual reference signal. It is possible to observe that in this constraint appears the variable ρ , that was introduced in the problem for the first time in the LMI problem to calculate the value of matrix P . So ρ needs to be determined before solving the LMI problem. From its definition, ρ must satisfy the condition $0 < \rho \leq \bar{u}$.

The effect of ρ in the LMI problem is to restrict the maximum value of the control signal of the error system. So, in order to have a large value of the control input, and hence a better tracking performance, ρ should be choose ad large as possible. On the other hand, analysing the new constraint defined in the control algorithm, ρ has the objective to limit the maximum value of $w(k|k)$. Hence to track a large reference, the value of ρ must be decreased.

From the previous considerations, it has been pointed out the importance of the value of ρ . The value has to be determined considering a trade-off, between the tracking performance and the maximum amplitude of the virtual reference signal.

Another parameter that must be chosen carefully is η_s . As said, this parameter is compared to the actual value of the cost function and, depending on whether it is minor or major, Mode 1 or Mode 2 is utilized. Consequently, η_s can assume a minimum value equal to 0, as the cost function. If $\eta_s = 0$, the Mode 1 is always called in the control algorithm. As a consequence, a rapid decrement of the cost function is observed and the error also converges to zero. However, if some disturbance occurs, the error increases because the integral action is not used. On the other hand, if η_s assumes a too large value, Mode 2 can be always called and an improvement of the robustness against disturbances is observed, but, in the same time, the convergence of the cost function is slower, with a decrement of the tracking performances.

Hence, the parameter η_s must be chosen taking into consideration a trade-off between robustness and tracking performance.

2.4 Introduction to Stochastic MPC

In paper [1] is treated the implementation of a Stochastic MPC, and possible future works after this thesis essay are to perform simulations and experiments with this stochastic control strategy. For this reason a preliminary step in the implementation of this control strategy is here presented.

In this paper, the implementation of an integrator is not considered, hence the system matrices A and B correspond to the plant matrices A_p and B_p that are defined in (3.20).

In the implementation of a Stochastic MPC, the following state space system subject to external disturbances and stochastic uncertainty is considered:

$$x(k+1|k) = A(q(k))x(k|k) + B(q(k))u(k|k) + w(k) \quad (2.33)$$

Where the state $x_k \in \mathbb{R}^{n_p}$, the input $u_k \in \mathbb{R}^m$, the additive disturbance $w_k \in \mathbb{R}^{m_w}$ and the parametric uncertainties $q_k \in \mathbb{R}^{n_q}$. In the following will be explained how the uncertainty q_k affects the matrices A and B of the system. The main problem of Stochastic control strategies is their slowness, but the considered Stochastic MPC has the great advantage to reduce the on-line computational time using a matrix K , calculated off-line, to obtain the optimal control sequence. Indeed the control input is obtained by the following equation:

$$u(k+l|k) = Kx(k+l|k) + v(k+l|k), \quad \forall l = 0, \dots, H_s - 1 \quad (2.34)$$

Where $x(k|k)$ is known, and the vector $\mathbf{v}(k) = [v(k|k)^T, \dots, v(k+H_s-1|k)^T]^T$ is a vector obtained solving an optimization problem.

For the off-line definition of the matrix K is needed the explication of the matrices $A(q(k))$ and $B(q(k))$ and is also important to highlight how the uncertainties affect these matrices.

2.4.1 Explication of matrix A with uncertainties

Starting considering the matrix A without uncertainties, its explicit form is:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{2}q_{d4} & -\frac{1}{2}q_{d3} & \frac{1}{2}q_{d2} \\ 0 & 0 & 0 & 0 & \frac{1}{2}q_{d3} & \frac{1}{2}q_{d4} & -\frac{1}{2}q_{d1} \\ 0 & 0 & 0 & 0 & -\frac{1}{2}q_{d2} & \frac{1}{2}q_{d1} & \frac{1}{2}q_{d4} \\ 0 & 0 & 0 & 0 & -\frac{1}{2}q_{d1} & -\frac{1}{2}q_{d2} & -\frac{1}{2}q_{d3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Substituting with the value of the final desired quaternion, $q_d = [0; 0; 0.2164; 0.9763]$, the matrix becomes:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0.4881 & -0.1082 & 0 \\ 0 & 0 & 0 & 0 & 0.1082 & 0.4882 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.4882 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.1082 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Considering the uncertainties on the angular velocity terms , the matrix A becomes:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0.4881 & -0.1082 & 0 \\ 0 & 0 & 0 & 0 & 0.1082 & 0.4882 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.4882 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.1082 \\ 0 & 0 & 0 & 0 & q_\omega & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_\omega & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q_\omega \end{bmatrix}$$

Where q_ω is the parametric uncertainty relative to the angular velocity.

2.4.2 Explication of matrix B with uncertainties

The explication of matrix B is more complicated with respect to the one of matrix A . The difficulty arises from the fact that the inertia tensor in this term $-h_\omega J^{-1}A(\theta)$ appears inverted; consequently, more mathematical passages are required. It's necessary starting by the definition of the inertia tensor J :

$$J = \begin{bmatrix} 0.9684 & -0.0062 & -0.0087 \\ -0.0062 & 0.9768 & -0.0074 \\ -0.0087 & -0.0074 & 1.300 \end{bmatrix}$$

Using MATLAB, the following inverse matrix is obtained:

$$J^{-1} = \begin{bmatrix} 1.0327 & 0.0066 & 0.0069 \\ 0.0066 & 1.0238 & 0.0059 \\ 0.0069 & 0.0059 & 0.7693 \end{bmatrix} = \begin{bmatrix} \dot{j}_{11} & \dot{j}_{12} & \dot{j}_{13} \\ \dot{j}_{21} & \dot{j}_{22} & \dot{j}_{23} \\ \dot{j}_{31} & \dot{j}_{32} & \dot{j}_{33} \end{bmatrix}$$

The angular momentum of each wheel is: $h_\omega = 0.0361 \text{ kg m}^2/\text{s}$ as explained in (4.1).

The matrix $A(\theta)$ is defined in (3.16).

So we can define the explicit form of the B matrix without parametric uncertainties as:

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -h_\omega[-j_{11} \cos(\theta_1)c - j_{12} \sin(\theta_1) + j_{13} \cos(\theta_1)s] & -h_\omega[j_{11} \sin(\theta_2) - j_{12} \cos(\theta_2)c + j_{13} \cos(\theta_2)s] \\ -h_\omega[-j_{21} \cos(\theta_1)c - j_{22} \sin(\theta_1) + j_{23} \cos(\theta_1)s] & -h_\omega[j_{21} \sin(\theta_2) - j_{22} \cos(\theta_2)c + j_{23} \cos(\theta_2)s] \\ -h_\omega[-j_{31} \cos(\theta_1)c - j_{32} \sin(\theta_1) + j_{33} \cos(\theta_1)s] & -h_\omega[j_{31} \sin(\theta_2) - j_{32} \cos(\theta_2)c + j_{33} \cos(\theta_2)s] \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -h_\omega[j_{11} \cos(\theta_3)c + j_{12} \sin(\theta_3) + j_{13} \cos(\theta_3)s] & -h_\omega[-j_{11} \sin(\theta_4) + j_{12} \cos(\theta_4)c + j_{13} \cos(\theta_4)s] \\ -h_\omega[j_{21} \cos(\theta_3)c + j_{22} \sin(\theta_3) + j_{23} \cos(\theta_3)s] & -h_\omega[-j_{21} \sin(\theta_4) + j_{22} \cos(\theta_4)c + j_{23} \cos(\theta_4)s] \\ -h_\omega[j_{31} \cos(\theta_3)c + j_{32} \sin(\theta_3) + j_{33} \cos(\theta_3)s] & -h_\omega[-j_{31} \sin(\theta_4) + j_{32} \cos(\theta_4)c + j_{33} \cos(\theta_4)s] \end{bmatrix}$$

Using $\theta = \theta_0 = [0; 0; 0; 0]$ we obtain the following matrix:

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.0262 & -0.0000 & -0.0265 & -0.0003 \\ 0.0000 & 0.0260 & -0.0003 & -0.0263 \\ -0.0195 & -0.0195 & -0.0198 & -0.0198 \end{bmatrix}$$

Considering uncertainties just on the diagonal of J:

$$J(q_p) = \begin{bmatrix} q_p + 0.9684 & -0.0062 & -0.0087 \\ -0.0062 & q_p + 0.9768 & -0.0074 \\ -0.0087 & -0.0074 & q_p + 1.300 \end{bmatrix}$$

The corresponding inverse matrix is:

$$J^{-1}(q_p) = \begin{bmatrix} \frac{2500(250000000q_p^2 + 56920000q_p + 31744631)}{62500000000q_p^3 + 202825000000q_p^2 + 217157764375q_p + 76845957632} & \frac{1250(310000q_p + 406219)}{62500000000q_p^3 + 202825000000q_p^2 + 217157764375q_p + 76845957632} & \frac{2500(217500q_p + 213601)}{62500000000q_p^3 + 202825000000q_p^2 + 217157764375q_p + 76845957632} \\ \frac{1250(310000q_p + 406219)}{62500000000q_p^3 + 202825000000q_p^2 + 217157764375q_p + 76845957632} & \frac{6250(74000q_p + 72201)}{62500000000q_p^3 + 202825000000q_p^2 + 217157764375q_p + 76845957632} & \frac{2500(250000000q_p^2 + 48630000q_p + 23647367)}{62500000000q_p^3 + 202825000000q_p^2 + 217157764375q_p + 76845957632} \\ \frac{2500(217500q_p + 213601)}{62500000000q_p^3 + 202825000000q_p^2 + 217157764375q_p + 76845957632} & \frac{2500(250000000q_p^2 + 48630000q_p + 23647367)}{62500000000q_p^3 + 202825000000q_p^2 + 217157764375q_p + 76845957632} & \frac{6250(74000q_p + 72201)}{62500000000q_p^3 + 202825000000q_p^2 + 217157764375q_p + 76845957632} \end{bmatrix}$$

Also if the uncertainty q_p appear just in the diagonal of the $J(q_p)$, the uncertainty propagates to all the terms of the inverse matrix $J^{-1}(q_p)$. But we are interested in a formulation where each term of the previous matrix is defined as follow:

$$\begin{aligned} J^{-1}(q_J) &= \begin{bmatrix} q_{J11} + \dot{j}_{11} & q_{J12} + \dot{j}_{12} & q_{J13} + \dot{j}_{13} \\ q_{J21} + \dot{j}_{21} & q_{J22} + \dot{j}_{22} & q_{J23} + \dot{j}_{23} \\ q_{J31} + \dot{j}_{31} & q_{J32} + \dot{j}_{32} & q_{J33} + \dot{j}_{33} \end{bmatrix} \\ &= \begin{bmatrix} q_{J11} + 1.0327 & q_{J12} + 0.0066 & q_{J13} + 0.0069 \\ q_{J21} + 0.0066 & q_{J22} + 1.0238 & q_{J23} + 0.0059 \\ q_{J31} + 0.0069 & q_{J32} + 0.0059 & q_{J33} + 0.7693 \end{bmatrix} = q_J + J^{-1} \end{aligned}$$

where J^{-1} is the inverse of the J matrix without uncertainties.

At this point to obtain the values of the q_{Jii} it's sufficient to put each term of the $J^{-1}(q_J)$ matrix, equal to the respective term of the $J^{-1}(q_p)$.

For example to obtain the value of q_{J11} :

$$q_{J11} + 1.0327 = \frac{2500(25000000q_p^2 + 56920000q_p + 31744631)}{62500000000q_p^3 + 202825000000q_p^2 + 217157764375q_p + 76845957632}$$

$$q_{J11} = \frac{2500(25000000q_p^2 + 56920000q_p + 31744631)}{62500000000q_p^3 + 202825000000q_p^2 + 217157764375q_p + 76845957632} - 1.0327$$

It's important to notice that the terms q_{Jii} are different among them.

Further more, performing simulation in MATLAB, I noticed that the values of the q_{Jii} not in the diagonal are significantly smaller with respect to the terms of the diagonal q_{J11}, q_{J22} and q_{J33} . So it's possible to make consideration in order to neglect the terms not in the diagonal.

For simplicity, the resultant B matrix, is written in the case in which all the terms q_{Jii} not in the diagonal are neglected:

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -h_\omega [-(q_{J11} + j_{11}) \cos(\theta_1)c - j_{12} \sin(\theta_1) + j_{13} \cos(\theta_1)s] & -h_\omega [(q_{J11} + j_{11}) \sin(\theta_2) - j_{12} \cos(\theta_2)c + j_{13} \cos(\theta_2)s] \\ -h_\omega [-j_{21} \cos(\theta_1)c - (q_{J22} + j_{22}) \sin(\theta_1) + j_{23} \cos(\theta_1)s] & -h_\omega [j_{21} \sin(\theta_2) - (q_{J22} + j_{22}) \cos(\theta_2)c + j_{23} \cos(\theta_2)s] \\ -h_\omega [-j_{31} \cos(\theta_1)c - j_{32} \sin(\theta_1) + (q_{J33} + j_{33}) \cos(\theta_1)s] & -h_\omega [j_{31} \sin(\theta_2) - j_{32} \cos(\theta_2)c + (q_{J33} + j_{33}) \cos(\theta_2)s] \\ \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -h_\omega [(q_{J11} + j_{11}) \cos(\theta_3)c + j_{12} \sin(\theta_3) + j_{13} \cos(\theta_3)s] & -h_\omega [-(q_{J11} + j_{11}) \sin(\theta_4) + j_{12} \cos(\theta_4)c + j_{13} \cos(\theta_4)s] \\ -h_\omega [j_{21} \cos(\theta_3)c + (q_{J22} + j_{22}) \sin(\theta_3) + j_{23} \cos(\theta_3)s] & -h_\omega [-j_{21} \sin(\theta_4) + (q_{J22} + j_{22}) \cos(\theta_4)c + j_{23} \cos(\theta_4)s] \\ -h_\omega [j_{31} \cos(\theta_3)c + j_{32} \sin(\theta_3) + (q_{J33} + j_{33}) \cos(\theta_3)s] & -h_\omega [-j_{31} \sin(\theta_4) + j_{32} \cos(\theta_4)c + (q_{J33} + j_{33}) \cos(\theta_4)s] \end{bmatrix}$$

Next steps for the implementation of this Stochastic MPC are the calculation of matrix K and to perform simulations and experiments.

Chapter 3

Spacecraft Dynamics and Model of CMG-based testbed

It's common practise to consider the testbed as a rigid body, in such a way that it's possible to apply for the spacecraft the Euler's equations of motion to describe the dynamics of the body. In particular, we are interested in the attitude motion, so the rotational equation of motion is the starting point for the derivation of the model.

The attitude is the orientation of a craft with respect to a reference frame. In the following work the attitude has been described through the use of the four-parameter quaternion representation. There exist also other attitude representation techniques such as the the Euler angles, Gibbs vector, Cayley Rodrigues vector and Modified Rodrigues parameter.

In this work has been studied a testbed equipped with 4 CMGs in pyramidal configuration. The CMG is a torque generator device, and it can be used to answer the attitude control requirements of future spacecrafts. The great advantage of this device is that it can improve the torque and momentum capability of the spacecraft and consequently their agility. On the other hand, the large diffusion of this device is limited by one great problem, the singularity condition. In certain combination of the gimbal angles is not possible to provide a value of torque in a particular direction. After the definition of the dynamic and kinematic equations of the testbed, the Linear Time-Invariant (LTI) mathematical model is derived in a state space formulation.

3.1 Reference Frames

For the study of spacecraft dynamics are usually utilized two reference frames: an Inertial Reference Frame and a Body Reference Frame.

3.1.1 Inertial Reference Frame

The inertial reference frame considered in this study is the Earth-Centered Inertial (ECI) that is also called Inertial Geocentric Reference Frame. It is commonly used to study the motion of a body orbiting around Earth. It has its origin in the center of mass of Earth. Thus, the x axis is on the equatorial plane, oriented towards the mean vernal equinox at J2000 epoch, the z axis is aligned with Earth's axis of rotation with positive sign towards the celestial North Pole, and the y axis completes the right-handed frame (Figure 3.1). The reference frame can be defined by:

$$RF_{ECI} = (x_{ECI}, y_{ECI}, z_{ECI}) \quad (3.1)$$

In Figure 3.1 is represented the Earth-Centered Inertial Reference frame.

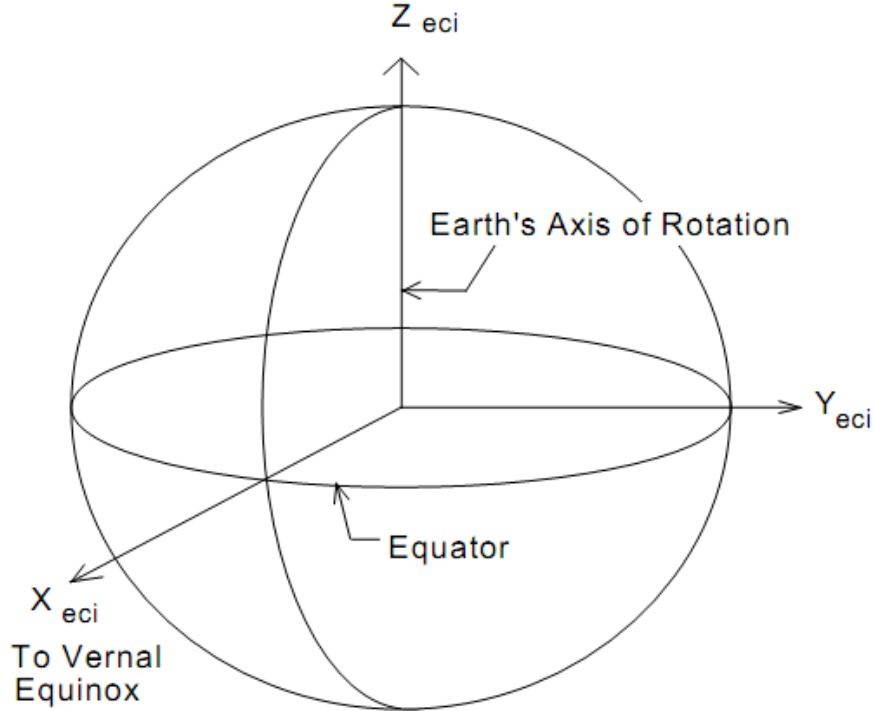


Figure 3.1: Earth-Centered Inertial Reference frame [5]

3.1.2 Body Reference Frame

The Body Reference Frame has been adopted to describe the attitude dynamics once introduced the CMGs to the equations of motion. The origin of the Body Reference Frame is in the center of mass of the spacecraft. If we consider principal axis of inertia, the Body Reference Frame has the axis oriented along the spacecraft principal axis of inertia. The z axis direction is positive from the lower to the upper side of the spacecraft, the x axis is orthogonal to z and the y axis completes the right-handed triad (Figure 3.2). The Reference frame can be defined as:

$$RF_B = (x_B, y_B, z_B) \quad (3.2)$$

In Figure 3.2 is represented the Body Reference Frame of the spacecraft.

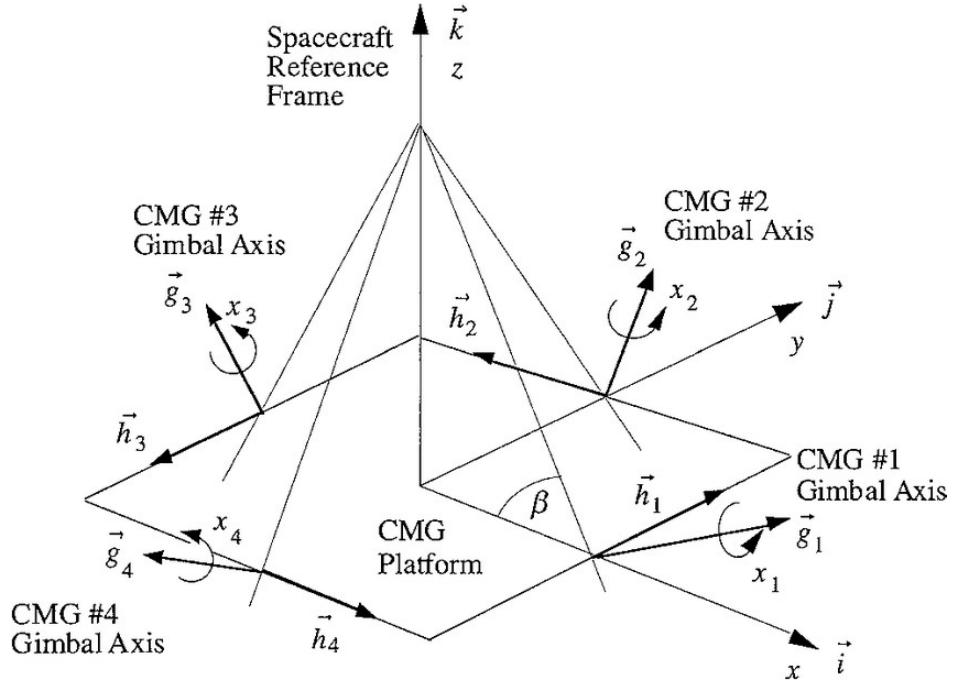


Figure 3.2: Body Reference Frame for a spacecraft with pyramidal CMG system [6]

3.2 Introduction to Dynamics

The dynamics of a general rigid body may be described in terms of its linear momentum \mathbf{l} and its angular momentum \mathbf{h} according to the following equations

[7]:

$$\mathbf{l} = m\mathbf{v} \quad (3.3)$$

$$\mathbf{h} = \mathbf{r} \times \mathbf{l} = \mathbf{r} \times (m\mathbf{v}) \quad (3.4)$$

In which \mathbf{v} and \mathbf{r} is the velocity and position with respect to an arbitrary point O, and m is the mass of the body. The former equation describes the trajectory of the body, the latter one the attitude.

Since we want to control the attitude of the spacecraft, the angular momentum \mathbf{h} will be studied.

The angular momentum decomposed in its Cartesian components is:

$$\mathbf{h} = h_1\hat{\mathbf{x}} + h_2\hat{\mathbf{y}} + h_3\hat{\mathbf{z}} \quad (3.5)$$

For a generic body, the inertia tensor directly relates the angular velocity with the angular momentum. The angular velocity is indicated with $\boldsymbol{\omega}$. The angular momentum \mathbf{h} of a rigid body, referred to its Center-of-Mass, can be defined:

$$\mathbf{h} = \mathbf{J}\boldsymbol{\omega} \quad (3.6)$$

In general the inertia tensor \mathbf{J} is defined as:

$$\mathbf{J} = \begin{bmatrix} J_x & -J_{xy} & -J_{xz} \\ -J_{xy} & J_y & -J_{yz} \\ -J_{xz} & -J_{yz} & J_z \end{bmatrix}$$

Where J_x , J_y and J_z are the moments of inertia with respect to the Center-of-Mass. The other terms in the matrix are the products of inertia and they represent the lack of inertia symmetry. The angular momentum Cartesian components can be defined as:

$$\begin{aligned} h_1 &= J_x w_1 - J_{xy} w_2 - J_{xz} w_3 \\ h_2 &= -J_{xy} w_1 + J_y w_2 - J_{yz} w_3 \\ h_3 &= -J_{xz} w_1 - J_{yz} w_2 + J_z w_3 \end{aligned}$$

Where w_1 , w_2 and w_3 are the angular velocity components.

From the following, $\hat{\mathbf{I}}$, $\hat{\mathbf{J}}$, $\hat{\mathbf{K}}$ is the notation used for the inertial reference frame, and $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, $\hat{\mathbf{k}}$ is the notation used for the body reference frame.

The angular momentum of a body, can be seen as the torque impulse needed to create the rotational motion of the body itself. As a consequence, the rotational

motion can be interrupted with the application of a torque impulse equal and opposite to the angular momentum \mathbf{h} . When the reference point respect which are calculated the values of velocity and position is the Center-of-Mass or an inertially fixed point, the angular momentum can be changed in two ways:

- By applying an external torque or a force that creates a moment about the reference point.

$$\frac{d^{\circ}\mathbf{h}}{dt} = {}^{\circ}\mathbf{M} \quad (3.7)$$

Where ${}^{\circ}\mathbf{M}$ is the applied torque. The notation ${}^{\circ}$ indicates the inertial reference frame. It's important to highlight that forces and torques exchanged between internal particles of the body, for example through the movement of the fuel, don't change the total angular momentum of the spacecraft.

- The second way to change the total angular momentum is through the emission of particles from the body, whose momenta with respect to the reference point is different from 0.

The angular momentum can be rewritten with respect to the Inertial Reference Frame as:

$${}^{\circ}\mathbf{h} = h_x\hat{\mathbf{I}} + h_y\hat{\mathbf{J}} + h_z\hat{\mathbf{K}} = {}^{\circ}\mathbf{J} {}^{\circ}\boldsymbol{\omega} \quad (3.8)$$

Rewriting the inertia tensor and the angular velocity in terms of the Body Reference Frame using the rotation matrix ${}^{\circ}\mathbf{R}_B$:

$$\begin{aligned} {}^{\circ}\mathbf{J} &= {}^{\circ}\mathbf{R}_B \mathbf{J}_B {}^{\circ}\mathbf{R}_B^T \\ \boldsymbol{\omega}_B &= {}^{\circ}\mathbf{R}_B {}^{\circ}\boldsymbol{\omega} \end{aligned}$$

Where \mathbf{J}_B indicates the Body Reference Frame.

The angular momentum with respect to the Body Reference Frame can be rewritten:

$$\mathbf{h}_B = h_x\hat{\mathbf{i}} + h_y\hat{\mathbf{j}} + h_z\hat{\mathbf{k}} \quad (3.9)$$

At this point, is assumed that the Body Reference Frame is rigidly attached to the body, in this way its angular velocity $\boldsymbol{\Omega}$ is equal to the angular velocity of the spacecraft $\boldsymbol{\omega}_B$. With the previous assumption, the absolute angular acceleration is

equal to the derivative of the angular velocity of the spacecraft, that in general is not true. The Euler's equation of motion can be rewritten:

$$\mathbf{M}_B = \dot{\mathbf{h}}_B + \boldsymbol{\omega}_B \times \mathbf{h}_B \quad (3.10)$$

Where:

$$\begin{aligned} M_x &= J_x \dot{\omega}_x + (J_z - J_y) \omega_y \omega_z \\ M_y &= J_y \dot{\omega}_y + (J_x - J_z) \omega_z \omega_x \\ M_z &= J_z \dot{\omega}_z + (J_y - J_x) \omega_x \omega_y \end{aligned}$$

3.3 Spacecraft Dynamic equation of motion

The initial dynamic equation of motion from which the model of the testbed is obtained, it's the equation that relates the rate of change of the angular momentum with the torque applied to the body (3.11) [13] [17]:

$$\dot{\mathbf{h}}_B = \mathbf{M}_B - \boldsymbol{\omega}_B \times \mathbf{h}_B \quad (3.11)$$

Where:

- \mathbf{M}_B is the torque vector defined as:

$$\mathbf{M}_B = \sum_{i=1}^n \mathbf{r}_i \times \mathbf{F}_i$$

where \mathbf{F}_i is a force that has a momentum with respect to the Center-of-mass, and \mathbf{r}_i is the distance between the considered force and the Center-of-mass;

- $\boldsymbol{\omega}_B$ is the angular velocity vector of the spacecraft.

In the equation (3.11) is present also the term $\boldsymbol{\omega}_B \times \mathbf{h}_B$ on the right side. This term indicates that also if $\mathbf{M}_B = 0$, the angular momentum \mathbf{h}_B , and consequently also the angular velocity $\boldsymbol{\omega}_B$ are not constant in the spacecraft. This phenomenon is called nutation.

Using again the equation that relates the angular momentum with the angular velocity, considering the whole system, it's possible to derive the following equation:

$$\mathbf{h}_B = \mathbf{J}_B \boldsymbol{\omega}_B + \mathbf{h}_B^C \quad (3.12)$$

The first term of the right side represents the angular momentum of the spacecraft, instead the second term indicates the contribution given by the CMG system. This last term can be rewritten, for a testbed equipped with 4 CMG in pyramid configuration, in the following way:

$$\mathbf{h}_B^C = h_\omega \sum_{i=1}^4 \overline{\mathbf{h}}_i(\theta_i) \quad (3.13)$$

Where $h_\omega = J_\omega \omega_\omega$ is the angular momentum of each wheel, and it is constant because in the following work the wheel velocity is set to a defined fix value; θ_i is the gimbal angle, for $i = 1, \dots, 4$.

Combining the previous equations it's possible to rewrite in the following way the rotational equation of motion:

$$\mathbf{M}_B = \boldsymbol{\omega}_B \times (J_B \boldsymbol{\omega}_B + \mathbf{h}_B^C) + J_B \dot{\boldsymbol{\omega}}_B + \dot{\mathbf{h}}_B^C \quad (3.14)$$

For a cluster of four CMGs, the internal angular momentum vector \mathbf{h}_B^C is non linear function of the gimbal angles $\theta_1, \theta_2, \theta_3, \theta_4$ of each CMG, as reported in (3.13). The time derivative of this equation, is possible to get:

$$\dot{\mathbf{h}}_B^C = h_\omega A(\theta) \dot{\boldsymbol{\theta}} \quad (3.15)$$

Where the matrix $A(\theta) \in \mathbb{R}^{3 \times 4}$, considering a fixed skew angle equal to $\beta = 45^\circ$, is defined:

$$A(\theta) = \begin{bmatrix} -\cos(\theta_1) \cos \beta & \sin(\theta_2) & \cos(\theta_3) \cos \beta & -\sin(\theta_4) \\ -\sin(\theta_1) & -\cos(\theta_2) \cos \beta & \sin(\theta_3) & \cos(\theta_4) \cos \beta \\ \cos(\theta_1) \sin \beta & \cos(\theta_2) \sin \beta & \cos(\theta_3) \sin \beta & \cos(\theta_4) \sin \beta \end{bmatrix} \quad (3.16)$$

The columns of matrix $A(\theta)$ expresses the contribution of each CMG to total $\dot{\mathbf{h}}_B^C$. Now the following assumptions are made:

1. The initial angular momentum is zero, and consequently $\boldsymbol{\omega}_B(0) = 0$
2. The initial torque applied to the body is zero, $\mathbf{M}_B = 0$

With these hypothesis the dynamic equation can be rewritten as:

$$\dot{\boldsymbol{\omega}}_B = -h_\omega J_B^{-1} A(\theta) \dot{\boldsymbol{\theta}} \quad (3.17)$$

The last equation is very useful because it relates the angular acceleration of the testbed with respect to the angular rate of the gimbals $\dot{\theta}$, that are the control input vector in this control system. For the definition of the model, the subscript that indicates the Body Reference Frame will be omitted for simplicity.

3.4 Spacecraft Kinematic equation of motion

To describe the attitude of a rigid body in a three-dimensional space are required at least three parameters [20]. There are different parametrization techniques to describe the rotational motion of bodies. In the presented work, the attitude of the testbed has been described using the quaternion representation. Another alternative method is the Euler Angles, that have the great characteristic to be easy to visualize, but in the same time, the problem of having an elevated computational cost; indeed the three angles are used for three successive transformations, hence the computational effort is high. Instead, the quaternion representation permits to describe the attitude just defining a unit vector around which is performed the rotation, having a smaller computational impact. The corresponding kinematic equation is:

$$\frac{d(q)}{dt} = \dot{q} = \frac{1}{2}\Gamma(q)\omega \quad (3.18)$$

With $q = [q_v^T, q_4]^T = [q_1, q_2, q_3, q_4]^T \in \mathbb{R}^4$, where q_v is the vectorial part of the quaternion, and q_4 is the scalar term.

The matrix $\Gamma(q) \in \mathbb{R}^{4 \times 3}$ is defined as:

$$\Gamma(q) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}$$

3.5 Mathematical Model of the system

The state vector for the considered system to represent the attitude dynamics is defined as:

$$x_p = \begin{bmatrix} q \\ \omega \end{bmatrix} \in \mathbb{R}^{n_p}$$

In this work $n_p = 7$.

Combining the dynamic and kinematic equations obtained previously:

$$\dot{x}_p = \begin{bmatrix} \dot{q} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\Gamma(q)\omega \\ -h_\omega J^{-1}A(\theta)\dot{\theta} \end{bmatrix} \quad (3.19)$$

Rearranging the previous equation in a state space formulation, we can obtain the state equation of the system:

$$\dot{x}_p = \begin{bmatrix} 0_{4 \times 4} & \frac{1}{2}\Gamma(q) \\ 0_{3 \times 4} & 0_{3 \times 3} \end{bmatrix} x_p + \begin{bmatrix} 0_{4 \times 4} \\ -h_\omega J^{-1}A(\theta) \end{bmatrix} \dot{\theta} = A_p x_p + B_p \dot{\theta}$$

Where is highlighted that the state of the system is $x_p \in \mathbb{R}^7$ and the control input is $\dot{\theta} \in \mathbb{R}^4$, that is the array with the four values of the angular rates of the gimbals. The system described by the previous equation is a Non Linear Time-Varying model because in matrix A_p is needed the actual value of the quaternion, and in matrix B_p is necessary the actual value of θ for the derivation of $A(\theta)$.

For ensure the stability of the plant, we need to linearise the model. The linearisation is performed assuming that the initial gimbal angles are $\theta_0 = [0, 0, 0, 0]^T$. This fixed value is used to calculate the matrix $A(\theta)$, in order to make constant the matrix B_p . To linearise the matrix A_p , the desired attitude in terms of ω_d and $q_d = [q_{1,d}, q_{2,d}, q_{3,d}, q_{4,d}]^T$ is considered as equilibrium point.

The Linear Time-Invariant model without any parameter dependence is:

$$\dot{x}_p = \begin{bmatrix} 0_{4 \times 4} & \frac{1}{2}\Gamma(q_d) \\ 0_{3 \times 4} & 0_{3 \times 3} \end{bmatrix} x_p + \begin{bmatrix} 0_{4 \times 4} \\ -h_\omega J^{-1}A(\theta_0) \end{bmatrix} \dot{\theta} = A_p x_p + B_p \dot{\theta} \quad (3.20)$$

To have a complete state space formulation, the output equation has been defined as:

$$y = \begin{bmatrix} I_3 & 0_{3 \times 4} \\ 0_{3 \times 4} & I_3 \end{bmatrix} x_p = C_p x_p \quad (3.21)$$

Where I_3 is a identity matrix with dimensions $\mathbb{R}^{3 \times 3}$. From the equation is pointed out that the output y has been chosen as $y = [q_v^T, \omega^T]^T \in \mathbb{R}^6$

3.6 Control Moment Gyroscopes description

The CMGs are devices able to generate torque for the attitude control of satellites. Compared to the reaction wheels, the CMGs are able to provide higher torques as

output and more rapid responses [7].

A CMG is composed by a flywheel rotating at constant or variable speed (in this work the speed is fixed) and by motorized gimbals. The creation of the torque comes from the variation of the angular momentum of the flywheel. Indeed, the rotating flywheel possesses angular momentum that is fixed in magnitude, because we chose to set a constant speed for the flywheel. Through the motors, which drive the gimbals, is possible to change the direction of the angular momentum vector of the flywheel, and thus generates a gyro-effect torque.

When the gimbal is rotating, the rotation axis of the flywheel points towards different directions, causing a change in its angular momentum orientation.

In Figure 3.3 is presented a CMG with a single gimbal, and the physical vector quantities are pointed out.

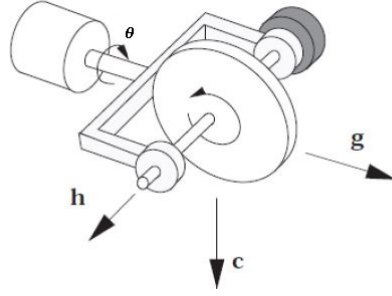


Figure 3.3: CMG with a single gimbal [7]

Three orthogonal vectors are shown, and are defined as:

- **g**: Gimbal vector;
- **c**: Torque vector;
- **h**: Angular momentum vector;

And θ is the gimbal angle. The mathematical relation between them is:

$$\mathbf{c} = \mathbf{g} \times \mathbf{h} \quad (3.22)$$

The gimbal vector is constant, while the other two vectors change orientation according to the gimbal angle θ . In Figure 3.4 is shown an example of how the orientation of the vectors changes with the rotation of the gimbal.

If we assume to know the initial vectors of the previous figure, the other vectors can be found in function of θ_i :

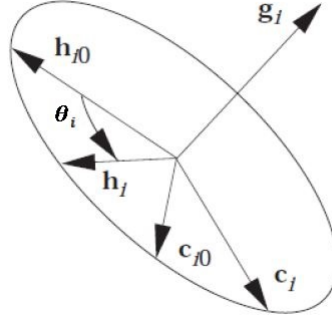


Figure 3.4: Physical vectorial quantities representation [7]

$$h_i = h_{i0} \cos(\theta_i) + c_{i0} \sin(\theta_i) \quad (3.23)$$

$$c_i = -h_{i0} \sin(\theta_i) + c_{i0} \cos(\theta_i) \quad (3.24)$$

Normally the CMGs are used in groups of 3 or 4, in order to ensure three-dimensional attitude control. The total torque provided by the group of CMGs will be the vectorial sum of each torque provided by a single CMG. The objective of the control will be to calculate the angular speed of each gimbal that the motors must provide to the CMGs in order to obtain a certain desired total torque as output. If we consider the angular momentum given by one CMG as h_j , the total angular momentum provided by the system will be:

$$\mathbf{h} = \sum_{j=1}^{N_{CMG}} \mathbf{h}_j = f(\theta) \quad (3.25)$$

Where the variable $\theta = [\theta_1, \dots, \theta_{N_{CMG}}]^T$ is the vector composed by all the gimbal angles which compose the system and N_{CMG} is the total number of CMGs in the system. The previous equation highlights the non-linear relation that holds between the total angular momentum of the system h and the vector of all the gimbal angles θ .

In Figure 3.5 is shown a testbed in pyramidal configuration equipped with 4 CMGs, the same type of system used in the simulations and experiments in this work.

The main problem encountered in the large diffusion of the CMGs is the singularity problem. Indeed, in certain combinations of the gimbal angles, it is not possible to provide torque along specific directions.

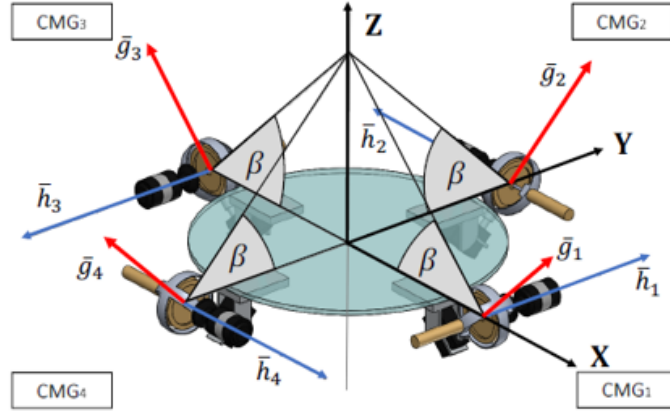


Figure 3.5: Testbed equipped with four CMGs in pyramidal configuration

If each CMG has the same speed and inertia tensor, the singular angular momentum provided by each of these devices will be the same. The total torque is given by the derivative of the angular momentum respect the time.

$$\mathbf{M} = \frac{d(\mathbf{h})}{dt} = \frac{d(\mathbf{h})}{d\theta} \dot{\theta} = A(\theta) \dot{\theta} \quad (3.26)$$

Where in general the matrix $A(\theta) \in \mathbb{R}^{3 \times N_{CMG}}$ is the instantaneous Jacobian matrix. In this work the matrix $A(\theta) \in \mathbb{R}^{3 \times 4}$ and has been already defined in (3.16). Each column of this matrix represents the torque vector of the respective CMG in the system. If the matrix were analysed, it would be noticed that the maximum rank of the matrix is equal to 3. When the rank of this matrix is:

$$\text{rank}(A(\theta)) < 3$$

the previous equation deteriorates. Indeed, each column of the matrix $A(\theta)$ becomes coplanar (one term of each column is null), and there exists a unit vector u orthogonal to that coplanar plane.

Therefore, the total torque M is also orthogonal to u , because is a combination of the columns of matrix $A(\theta)$, and so the system is not able to generate any momentum along the direction of u . This coincides with a singularity condition, and must be avoided at any cost. As said, one of the great advantages to use MPC as control law, is the fact that can directly calculate the value of the angular velocity of the gimbal. Indeed, more common control techniques, like a simple PID controller, calculate the value of the desired total torque to follow the reference signal, and then the values of the angular velocities of the gimbals is obtained from the previous equation in which is used the desired total torque and the matrix $A(\theta)$. This last

passage is the critical one, in which the singularity condition can be encountered. For this reason special singularity avoidance algorithms must be implemented if a common control technique has been chosen.

Chapter 4

Simulations in Matlab environment

After having defined the model of the spacecraft and the considered MPC, the following chapter is intended to explain the implementation in MATLAB of the control system and also to show the obtained results. In MATLAB three different MPC strategies have been implemented: the first one is a classic linear MPC (from the optimization problem is obtained just the optimal sequence of control inputs); in the second presented simulation, is implemented an MPC in which just the Mode 2 is used (optimization of the sequence of the control inputs and of the virtual reference signals); the last one is obtained with an MPC constructed with Mode 1 and 2, as treated in [18]. With the representation of the obtained results, is also highlighted the improvement obtained adding the new characteristics.

4.1 Optimization function

In this work different optimization functions have been used in the simulations on MATLAB and all these functions attempt to solve the quadratic programming problem:

$$\min_{\psi} \frac{1}{2} \psi(k)' H \psi(k) + f' \psi(k) \quad \text{subject to : } A_{con} \psi \leq B_{con}$$

Where the vector $\psi(k)$ is the optimized vector.

For convenience, it is better to rewrite the cost function in the following intermediate way, already seen in (2.6), before to write it in the final format for the implementation in MATLAB:

$$\min_{\psi} \frac{1}{2} \psi(k)' H \psi(k) + \phi(k)' F \psi(k) + \frac{1}{2} \phi(k)' Y \phi(k) \quad \text{subject to : } A_{con} \psi \leq B_{con}$$

Where the vector ϕ is a vector of the variables already known before the optimization procedure. Starting from the formulation of the cost function described in [18], it is necessary to rewrite it in the just presented way to implement the optimization procedure in MATLAB. To this end will be discussed the implementation for the various types of MPC defined previously.

The tested optimization functions were:

- *quadprog*, the interior-point-convex algorithm provided by the MATLAB Optimization Toolbox to solve quadratic programming problem;
- *fmincon*, MATLAB function that can implement four different algorithms: interior-point, SQP, active set, and trust region reflective. Also this solver deals quadratic programming problems.
- *quadwright*, a quadratic programming solver developed by J. Currie presented in [21], able to speed up the computational capabilities for embedded applications.

All these solvers provided good and similar results performing simulations, but for the experiments has been decided to use the solver *quadwright*. Indeed, this function provided the smallest computational times, hence it is the best for real time implementation. Furthermore, this one was the only solver that could be converted by MATLAB Coder Toolbox for the creation of the C++ algorithm.

4.1.1 Virtual reference term of the cost function

Before entering into the details of the implementations, is useful analyse the term of the cost function referred to the virtual reference signal $\|\Pi(w(k|k) - r(k|k))\|_M^2$, because it must be decomposed in different terms to construct the cost function as described above. Analysing this term:

$$\begin{aligned} & [\Pi(w(k|k) - r(k|k))]^T M [\Pi(w(k|k) - r(k|k))] = \\ & (\Pi w(k|k))^T M (\Pi w(k|k)) + (\Pi r(k|k))^T M (\Pi r(k|k)) - 2(\Pi r(k|k))^T M (\Pi w(k|k)) = \\ & w(k|k)^T \Pi^T M \Pi w(k|k) + r(k|k)^T \Pi^T M \Pi r(k|k) - 2r(k|k)^T \Pi^T M \Pi w(k|k) \end{aligned}$$

In this way the term has been split in three parts, and according to which variables are included in each part, the first one will be included in the matrix H , the second one in Y and the third one in F .

4.2 MPC Implementation

4.2.1 Classic MPC implementation

Considering the implementation of a classic MPC using the error system described in paper [18], we can define two vectors, ϕ_0 , is the vector in which are collected all the variables known before the optimization procedure, and ψ_0 , constituted by the variables that will be obtained by the optimization procedure, and hence are unknown before it:

$$\phi_0(k) = \begin{bmatrix} x_p(k|k) \\ x_c(k|k) \\ r(k) \end{bmatrix}, \quad \psi_0(k) = \begin{bmatrix} \tilde{\nu} \end{bmatrix} \quad (4.1)$$

Where $\phi_0 \in \mathbb{R}^{n_p+n_y+n_y}$ and $\psi_0 \in \mathbb{R}^{H_s n_u}$.

The vector ψ_0 is simply the optimized control input sequence, because in this case the implementation of a classic MPC is treated; ϕ_0 is constituted by the error system state and from the reference. It's important to notice that in the following MPC the virtual reference signal w is not present, as consequence, in the definition of the error system state and input, the virtual reference must be substituted by the real reference r :

$$\begin{aligned} \xi(k+i|k) &= \begin{bmatrix} x_p(k+i|k) \\ x_c(k+i|k) \end{bmatrix} - \Pi r(k) = x(k+i|k) - \Pi r(k) \\ \nu(k+i|k) &= u(k+i|k) - \Pi r(k) \end{aligned}$$

At this point, the prediction equation is defined as:

$$\begin{aligned} \begin{bmatrix} \xi(k+1|k) \\ \xi(k+2|k) \\ \vdots \\ \xi(k+H_s|k) \end{bmatrix} &= \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{H_s-1}B & A^{H_s-2}B & \cdots & B \end{bmatrix}}_{S_0} \begin{bmatrix} \nu(k|k) \\ \nu(k+1|k) \\ \vdots \\ \nu(k+H_s-1|k) \end{bmatrix} + \\ &\quad + \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_s} \end{bmatrix}}_{T_0} \xi(k|k) \end{aligned}$$

In this case, the last term of the cost function, related to the virtual reference signal, is null. Consequently the cost function has the same terms of the cost function already seen in (2.4). The definition of the matrices H, F and Y is carried out in the same way as in (2.6)

4.2.2 Mode 1 implementation

Considering the Mode 1, for the implementation in MATLAB, is useful to define the following vectors:

$$\phi_1(k) = \begin{bmatrix} x_p(k|k) \\ r(k) \end{bmatrix}, \quad \psi_1(k) = \begin{bmatrix} \tilde{\nu} \\ w(k|k) \\ x_c(k|k) \end{bmatrix} \quad (4.2)$$

Where $\phi_1 \in \mathbb{R}^{n_p+n_y}$ and $\psi_1 \in \mathbb{R}^{H_s n_u+n_y+n_y}$.

The vector ϕ_1 define the variables known before the optimization procedure; in particular $r(k|k)$ is known a priori and $x_p(k|k)$ is given by sensors. The vector ψ_1 is the vector of all the variables optimized; it's important to notice that in Mode 1, the integrator state $x_c(k|k)$ is included as variable in the optimization problem.

Remembering that $\xi(k|k) = \begin{bmatrix} x_p(k|k) \\ x_c(k|k) \end{bmatrix} - \Pi w(k|k)$, the prediction equation for the error system becomes:

$$\begin{aligned} \begin{bmatrix} \xi(k+1|k) \\ \xi(k+2|k) \\ \vdots \\ \xi(k+H_s|k) \end{bmatrix} &= \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{H_s-1}B & A^{H_s-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} \nu(k|k) \\ \nu(k+1|k) \\ \vdots \\ \nu(k+H_s-1|k) \end{bmatrix} + \\ &+ \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_s} \end{bmatrix}}_{\bar{A}} \left(\begin{bmatrix} x_p(k|k) \\ x_c(k|k) \end{bmatrix} - \Pi w(k|k) \right) \end{aligned}$$

At this point, defining the matrix $\bar{A} = [\bar{A}_{x_p} \quad \bar{A}_{x_c}]$:

$$\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_s} \end{bmatrix} = \begin{bmatrix} A_{x_p} & A_{x_c} \\ A_{x_p}^2 & A_{x_c}^2 \\ \vdots & \vdots \\ A_{x_p}^{H_s} & A_{x_c}^{H_s} \end{bmatrix}$$

Using the obtained new nomenclature for the matrix \bar{A} , and rewriting the prediction equation in function of ψ_1 :

$$\begin{bmatrix} \xi(k+1|k) \\ \xi(k+2|k) \\ \vdots \\ \xi(k+H_s|k) \end{bmatrix} = \begin{bmatrix} B & 0 & \cdots & 0 & -A\Pi & A_{x_c} \\ AB & B & \cdots & 0 & -A^2\Pi & A_{x_c}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ A^{H_s-1}B & A^{H_s-2}B & \cdots & B & -A^{H_s}\Pi & A_{x_c}^{H_s} \end{bmatrix} \underbrace{\begin{bmatrix} \tilde{\nu} \\ w(k|k) \\ x_c(k|k) \end{bmatrix}}_{\psi_1} +$$

$$+ \begin{bmatrix} A_{x_p} \\ A_{x_p}^2 \\ \vdots \\ A_{x_p}^{H_s} \end{bmatrix} x_p(k|k)$$

Including also the reference $r(k)$ in the previous equation, we obtain a formulation in which are included just ψ_1 and ϕ_1 :

$$\begin{bmatrix} \xi(k+1|k) \\ \xi(k+2|k) \\ \vdots \\ \xi(k+H_s|k) \end{bmatrix} = \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 & -A\Pi & A_{x_c} \\ AB & B & \cdots & 0 & -A^2\Pi & A_{x_c}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ A^{H_s-1}B & A^{H_s-2}B & \cdots & B & -A^{H_s}\Pi & A_{x_c}^{H_s} \end{bmatrix}}_{S_1} \psi_1(k) +$$

$$+ \underbrace{\begin{bmatrix} A_{x_p} & 0 \\ A_{x_p}^2 & 0 \\ \vdots & \vdots \\ A_{x_p}^{H_s} & 0 \end{bmatrix}}_{T_1} \phi_1(k)$$

At this point, using the definition of the matrices \bar{Q} and \bar{R} explicated in (2.5), and

defining for simplicity the state prediction $\tilde{\xi} = \begin{bmatrix} \xi(k+1|k) \\ \xi(k+2|k) \\ \vdots \\ \xi(k+H_s|k) \end{bmatrix}$, the cost function

(2.26) becomes:

$$J(k) = \underbrace{\tilde{\xi}' \bar{Q} \tilde{\xi}}_{\psi_1(k)'} + \underbrace{\begin{bmatrix} \tilde{\nu} \\ w(k|k) \\ x_c(k|k) \end{bmatrix}'}_{\psi_1(k)'} \underbrace{\begin{bmatrix} \bar{R} & 0 & 0 \\ 0 & \Pi' M \Pi & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\bar{R}_1} \underbrace{\begin{bmatrix} \tilde{\nu} \\ w(k|k) \\ x_c(k|k) \end{bmatrix}}_{\psi_1(k)} +$$

$$+ r(k|k)' \Pi' M \Pi r(k|k) - 2r(k|k)' \Pi' M \Pi w(k|k)$$

In the previous cost function definition, the term $w(k|k)' \Pi' M \Pi w(k|k)$ has been already included in the definition of matrix \bar{R}_1 .

If the last terms are rewritten in terms of $\phi_1(k)$ and $\psi_1(k)$, and if the prediction equation is inserted, the cost function becomes:

$$J(k) = (S_1 \psi_1(k) + T_1 \phi_1(k))' \bar{Q} (S_1 \psi_1(k) + T_1 \phi_1(k)) + \phi_1(k)' \bar{R}_1 \phi_1(k) +$$

$$+\phi_1(k)' \begin{bmatrix} 0 & 0 \\ 0 & \Pi' M \Pi \end{bmatrix} \phi_1(k) + \phi_1(k)' \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2\Pi' M \Pi & 0 \end{bmatrix} \psi_1(k)$$

After some mathematical calculation:

$$\begin{aligned} J(k) = & \frac{1}{2} \psi_1(k)' \underbrace{(2(S_1' \overline{Q} S_1 + \overline{R}_1))}_{H_1} \psi_1(k) + \phi_1(k)' \underbrace{\left(\begin{bmatrix} 0 & 0 \\ 0 & \Pi' M \Pi \end{bmatrix} + T_1' \overline{Q} T_1 \right)}_{Y_1} \phi_1(k) + \\ & + \phi_1(k)' \underbrace{\left(2T_1' \overline{Q} S_1 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2\Pi' M \Pi & 0 \end{bmatrix} \right)}_{F_1} \psi_1(k) \end{aligned}$$

The previous definition of the cost function is the one useful for the optimization functions utilised in MATLAB.

4.2.3 Mode 2 implementation

Considering the implementation of Mode 2, it's necessary to reformulate the prediction equation, this time considering the integrator state x_c as a known variable before the optimization procedure. Consequently, the vectors used in the optimization problem are defined as:

$$\phi_2(k) = \begin{bmatrix} x_p(k|k) \\ x_c(k|k) \\ r(k) \end{bmatrix}, \quad \psi_2(k) = \begin{bmatrix} \tilde{v} \\ w(k|k) \end{bmatrix} \quad (4.3)$$

Where the integrator state is obtained updating the integrator equations. The considerations for the other variables are the same explicated previously for the Mode 1 implementation.

Remember that $\xi(k|k) = \begin{bmatrix} x_p(k|k) \\ x_c(k|k) \end{bmatrix} - \Pi w(k|k) = x(k|k) - \Pi w(k|k)$, the prediction equation for the error system becomes:

$$\begin{aligned} \begin{bmatrix} \xi(k+1|k) \\ \xi(k+2|k) \\ \vdots \\ \xi(k+H_s|k) \end{bmatrix} = & \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{H_s-1}B & A^{H_s-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} \nu(k|k) \\ \nu(k+1|k) \\ \vdots \\ \nu(k+H_s-1|k) \end{bmatrix} + \\ & + \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_s} \end{bmatrix} \left(\begin{bmatrix} x_p(k|k) \\ x_c(k|k) \end{bmatrix} - \Pi w(k|k) \right) \end{aligned}$$

Rewriting the previous equation in function of ψ_2 and ϕ_2 :

$$\begin{bmatrix} \xi(k+1|k) \\ \xi(k+2|k) \\ \vdots \\ \xi(k+H_s|k) \end{bmatrix} = \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 & -A\Pi \\ AB & B & \cdots & 0 & -A^2\Pi \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A^{H_s-1}B & A^{H_s-2}B & \cdots & B & -A^{H_s}\Pi \end{bmatrix}}_{S_2} \psi_2(k) + \underbrace{\begin{bmatrix} A & 0 \\ A^2 & 0 \\ \vdots & \vdots \\ A^{H_s} & 0 \end{bmatrix}}_{T_2} \phi_2(k)$$

At this point it is possible to obtain the required structure of the cost function for the optimization procedure following the same steps explained for the implementation of Mode 1, substituting S_1 and T_1 with S_2 and T_2 .

4.3 Parameters used in the simulation

In Table 4.1 are defined the principal physical quantities utilised to perform the simulations and also used for the experiments. The value of the inertia tensor of the testbed is:

$$J \text{ [kg/m}^2\text{]} = \begin{bmatrix} 0.9684 & -0.0062 & -0.0087 \\ -0.0062 & 0.9768 & -0.0074 \\ -0.0087 & -0.0074 & 1.300 \end{bmatrix} \quad (4.4)$$

Furthermore, in all the simulations the initial conditions of the state x and of the gimbal angles θ are:

$$\theta(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad x(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.5)$$

The other decision variables such as the weight matrices Q, R and M have been chosen with different values for each simulation case.

Variable	Nomenclature	Numerical value	Units
Angular momentum of each wheel	h_ω	0.0361	kg m ² /s
Input constraint (gimbal motors speed max value)	$\bar{\dot{\theta}}$	4	rad/s
Angular velocity constraint of the testbed	$\bar{\omega}$	5	deg/s
Acceleration constraints on the CMG motors	$\bar{\ddot{\theta}}$	2	rad/s ²
Horizon of prediction	H_s	5	-

Table 4.1: Parameters used for the simulations

4.4 Simulation Results

The simulations have a total duration of 30 seconds. The angular velocity reference has a triangular shape for the third component, the references for the other two velocity components are equal to zero for the entire duration of the simulation. This means that, for the first 5 seconds, the testbed should increase its angular velocity around its vertical axis; then, for the next 5 seconds, the velocity should decrease in such a way that a value equal to zero is reached at the 10 second. From that moment, all the components of the angular velocity reference of the testbed are zero. The reference signals for the vectorial part of the quaternion have been suitably calculated utilizing the angular velocity reference signals.

For the next plots, the actual values of each quantity is indicated with a blue line, the reference signals are indicated in red and the virtual reference signal is represented in yellow.

4.4.1 Classic MPC results

The first simulation has been performed implementing a classic MCP with the integrator as servo compensator. In Figure 4.1 are reported the vectorial part of the quaternion. In Figure 4.2 are represented the values of the testbed angular velocity. The vectorial part of the quaternion and the angular velocity of the testbed are the output of the control system. From the images, it is possible to notice a growing oscillatory behaviour along the time. This compromise the stability of the control system. In Figure 4.3 are reported the values of the angular speeds of the motors which control the CMG gimbals; also in this variable is observed a oscillatory behaviour.

In Figure 4.4 are plotted the derivatives of the control inputs. To respect the

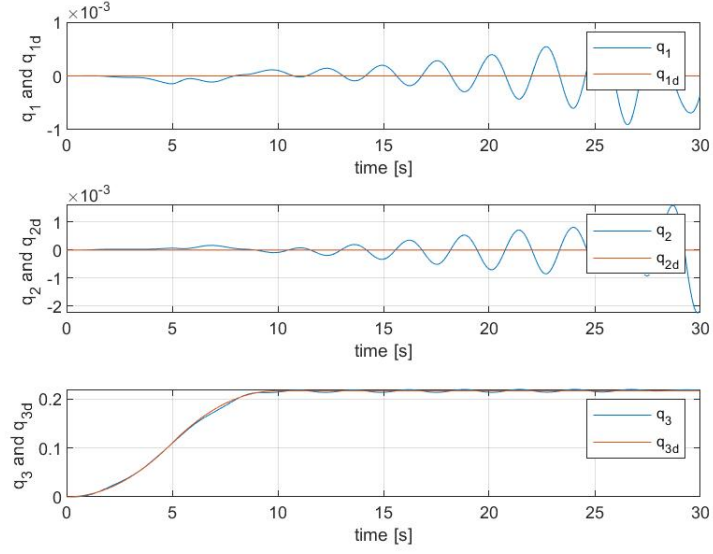


Figure 4.1: Vectorial part of the quaternion of the classic MPC

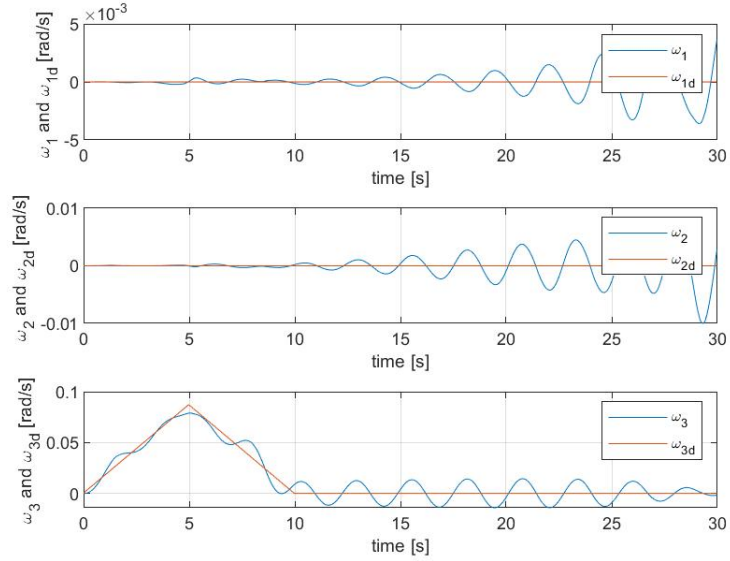


Figure 4.2: Angular velocity components of the classic MPC

hardware constraint of the testbed, a further acceleration constraint has been added for this control algorithm, to bound the maximum absolute value of the acceleration speed of the gimbals equal to 2 rad/s^2 .

In Figure 4.5 are plotted the values of the gimbal angles during the simulation.

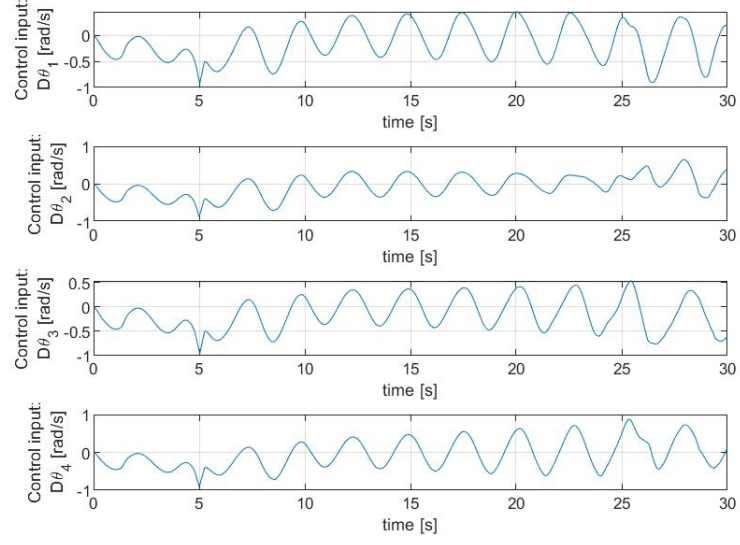


Figure 4.3: Input signals of the classic MPC

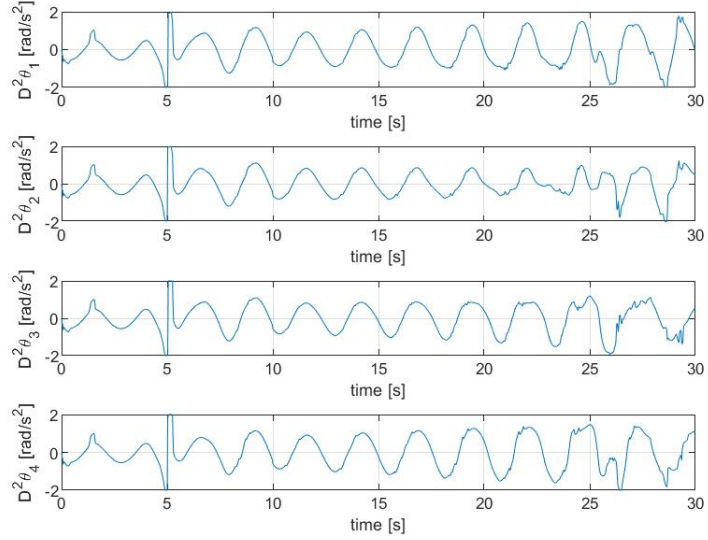


Figure 4.4: Acceleration of the input signals of the classic MPC

In Figure 4.6 are reported the values of the 4 components of the quaternion. In particular is interesting to notice that the fourth component change, despite it is not an output of the system and hence it has not a reference to follow. Indeed, the scalar component change in order to ensure that the quaternion, in each instant of

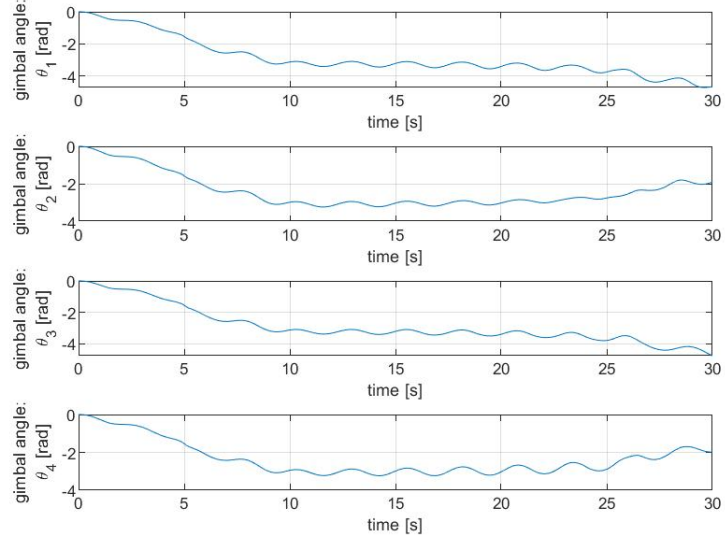


Figure 4.5: Gimbals evolution of the classic MPC

the simulation, is a unit quaternion, with absolute value equal to one.

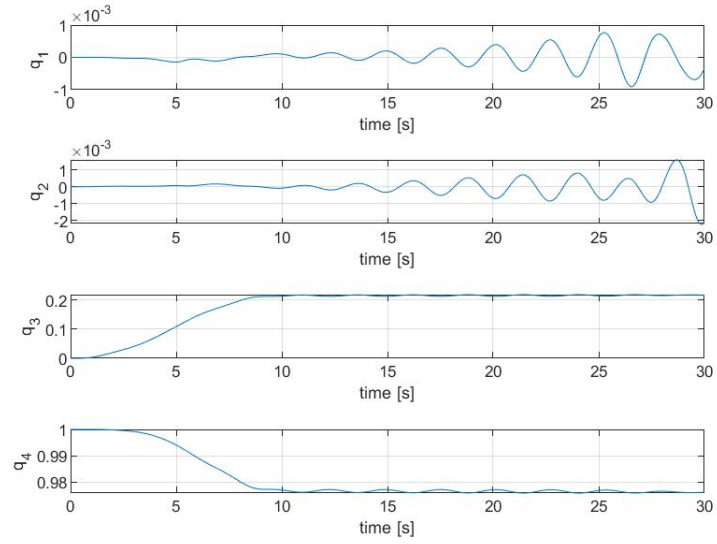


Figure 4.6: Quaternion components of the classic MPC

4.4.2 MPC with Mode 2 results

In this second simulation, an MPC with virtual reference signal optimization has been implemented. In this case, the integrator state is never involved in the optimization procedure, so it is simply updated by the integrator equations (2.17) and (2.18). In Figure 4.7 is plotted the actual values of the vectorial components of the quaternion, with the respective reference signals. In Figure 4.8 are reported the

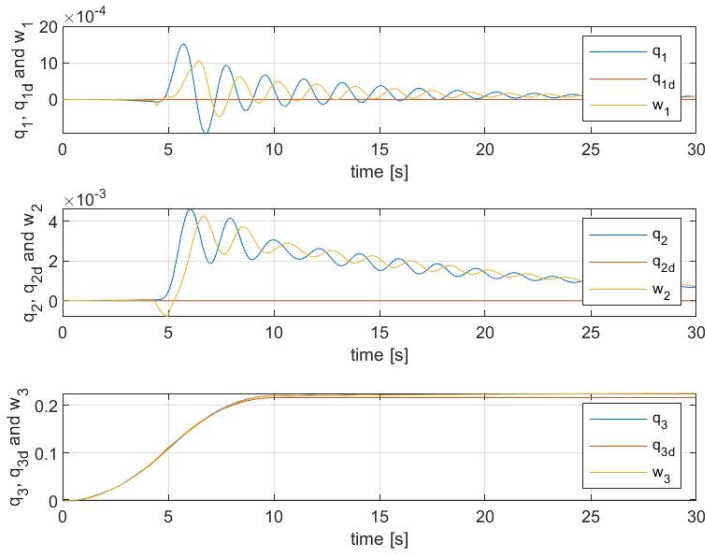


Figure 4.7: Vectorial part of the quaternion of the MPC with Mode 2

angular velocity components of the testbed. In this simulation appears the virtual reference signal in yellow, in the graphs of the output variables. Respect to the first simulation, in this case, utilizing the virtual reference, is possible to obtain a decreasing oscillatory behaviour, hence the stability of the control system is improved with respect the previous case. In Figure 4.9 are represented the control inputs of the system. In Figure 4.10 are reported the value of the acceleration of the control input. In this case the acceleration constraints have not been implemented cause the accelerations are always inside the permitted range of values. In Figure 4.11 are reported the time evolutions of the gimbal angles. In Figure 4.12 are plotted the four components of the quaternion. Also in this case the scalar term varies to ensure the unit quaternion condition.

4.4.3 MPC with Mode 1 and 2 results

In the third simulation, an MPC with virtual reference signal and integrator state optimization has been implemented. In this case, if the integrator state is involved

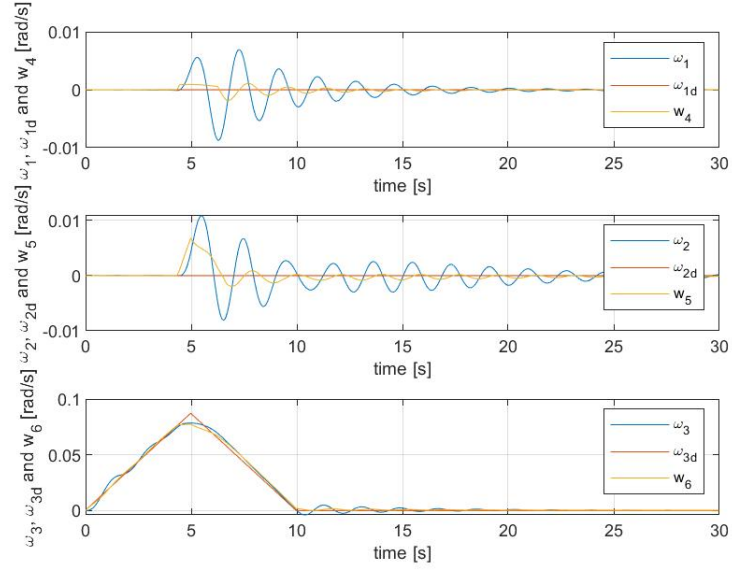


Figure 4.8: Angular velocity components of the MPC with Mode 2

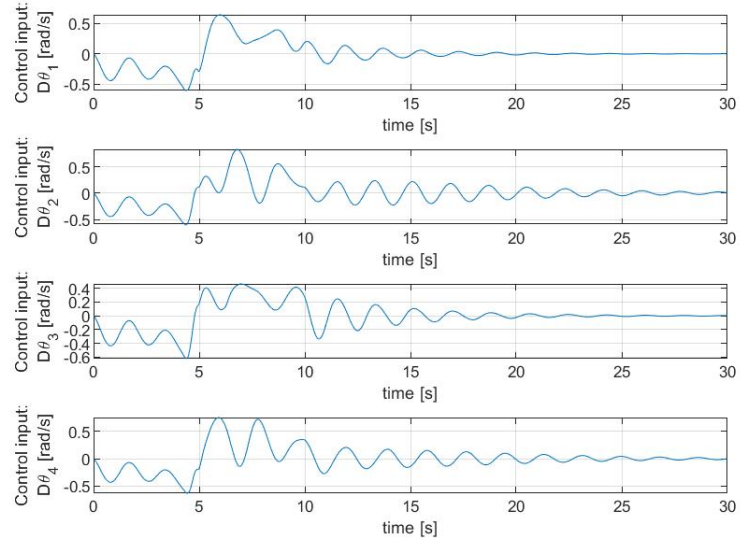


Figure 4.9: Input signals of the MPC with Mode 2

in the optimization procedure, its value is rapidly decreased; if it is not considered in the optimization, it is simply updated by the integrator equations (2.17) and (2.18). In Figure 4.13 is plotted the actual values of the vectorial components of the quaternion, with the respective reference signals and virtual reference signals. In

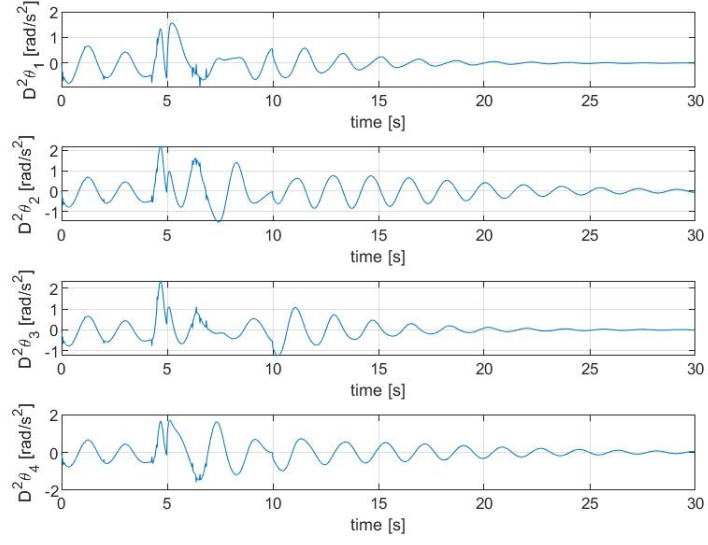


Figure 4.10: Acceleration of the input signals of the MPC with Mode 2

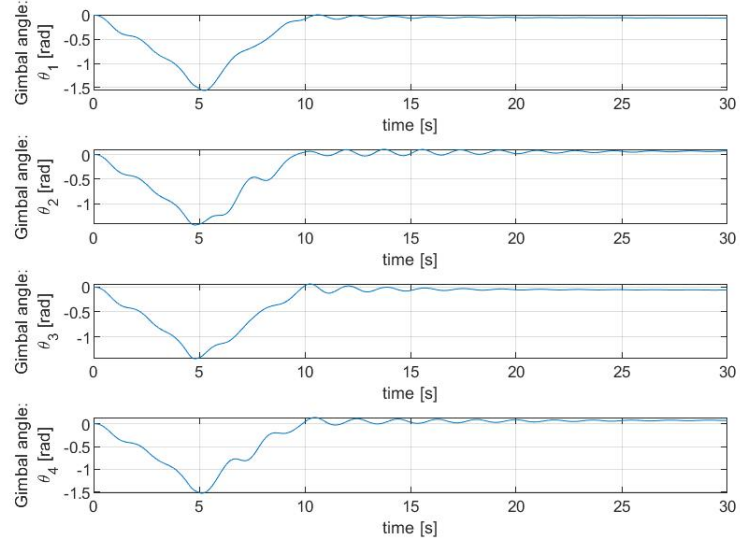


Figure 4.11: Gimbals evolution of the MPC with Mode 2

Figure 4.14 are reported the angular velocity components of the testbed. With the integrator state reset, in this third simulation, the outputs have been improved with respect the previous simulation. Indeed, the output signals show an attenuation of the oscillations, hence the system response is better with respect the previous

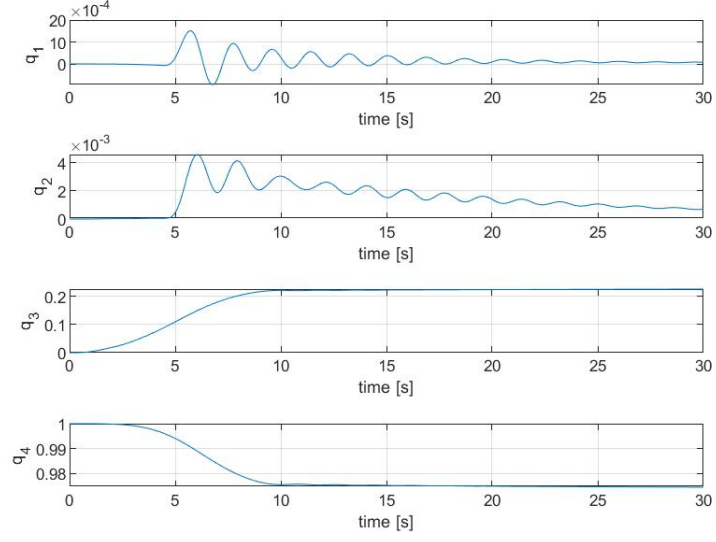


Figure 4.12: Quaternion components of the MPC with Mode 2

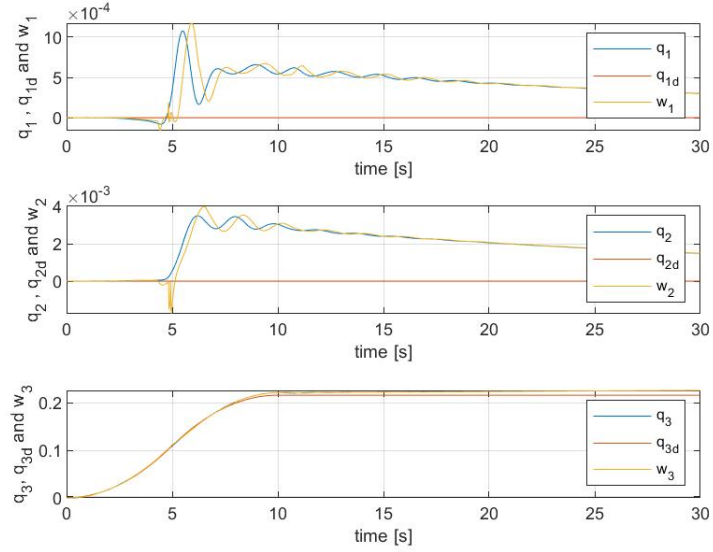


Figure 4.13: Vectorial part of the quaternion of the MPC with Mode 1 and 2

cases. In Figure 4.15 are represented the control inputs of the system. In Figure 4.16 are reported the value of the acceleration of the control input. In this case the acceleration constraints has been implemented because the acceleration had too high values. In Figure 4.17 are reported the time evolutions of the gimbal angles.

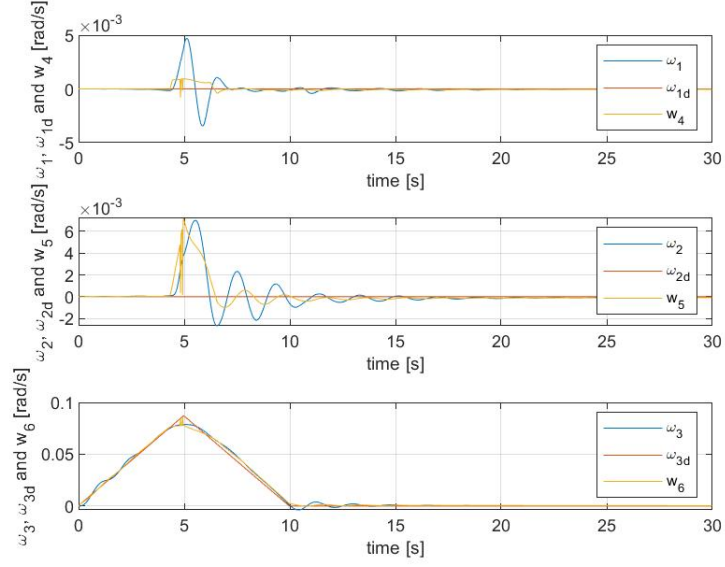


Figure 4.14: Angular velocity components of the MPC with Mode 1 and 2

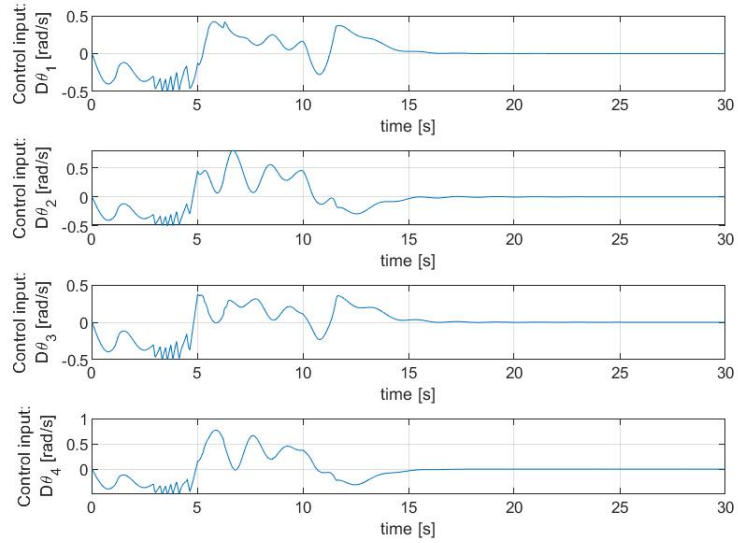


Figure 4.15: Input signals of the MPC with Mode 1 and 2

In Figure 4.18 are plotted the four components of the quaternion. Also in this case the scalar term varies to ensure the unit quaternion condition.

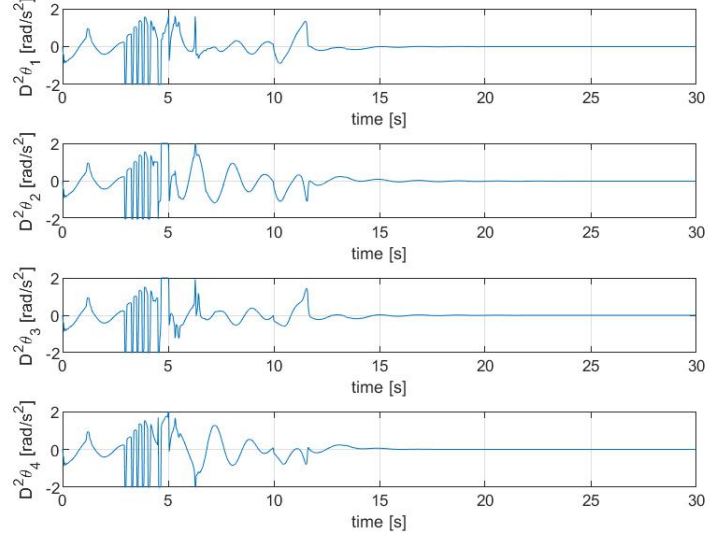


Figure 4.16: Acceleration of the input signals of the MPC with Mode 1 and 2

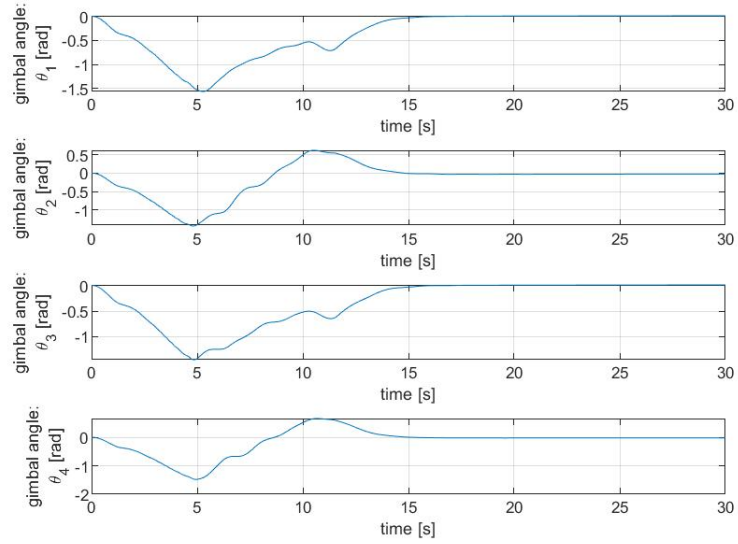


Figure 4.17: Gimbals evolution of the MPC with Mode 1 and 2

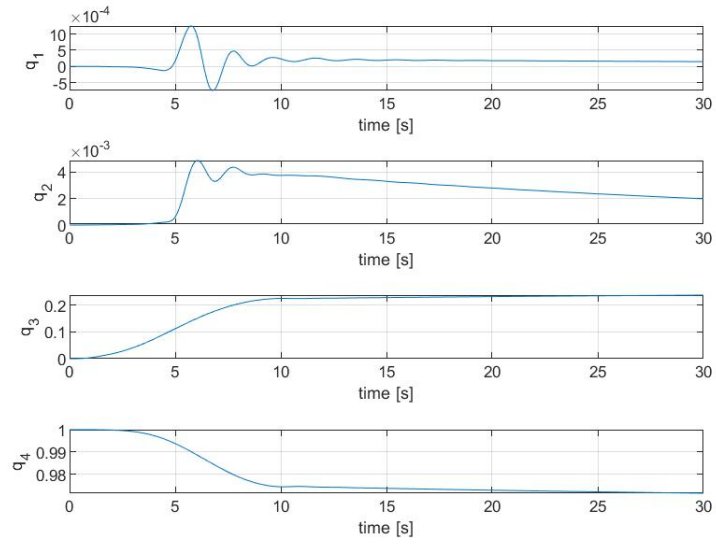


Figure 4.18: Quaternion components of the MPC with Mode 1 and 2

Chapter 5

Testbed characteristics and Experimental results

5.1 Experimental setup

The testbed used in this work, located at the laboratories of Osaka University (Japan) [22], simulates a spacecraft using an air floating table as represented in Figure 5.2 and Figure 5.3.

The problem of this experimental setup is that some misalignments between the center of rotation and the center of gravity can happen. In this situation gravity torques arise even if there is a slightly misalignment as shown in Figure 5.1. To minimize this gravity effect, the experimental setup is equipped with three counter weights, which are attached to the testbed by ball screws (Figure 5.2). The position of the three counter weights must be adjusted manually, hence it is very complicated to obtain a perfect alignment of the center of gravity and of the center of rotation. In adjusting the counter weights, the period of the platform is empirically maximized if the center of rotation coincides with the centre of gravity as much as possible. The simulation parameters treated in Chapter 4 have been chosen because they represent the physical characteristics of the described testbed utilized for the experiments.

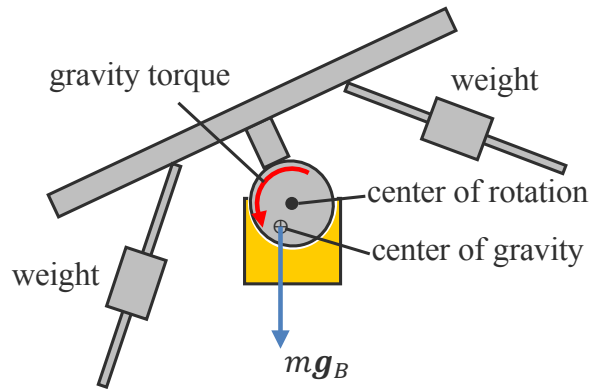


Figure 5.1: Gravity effect on the testbed [22]

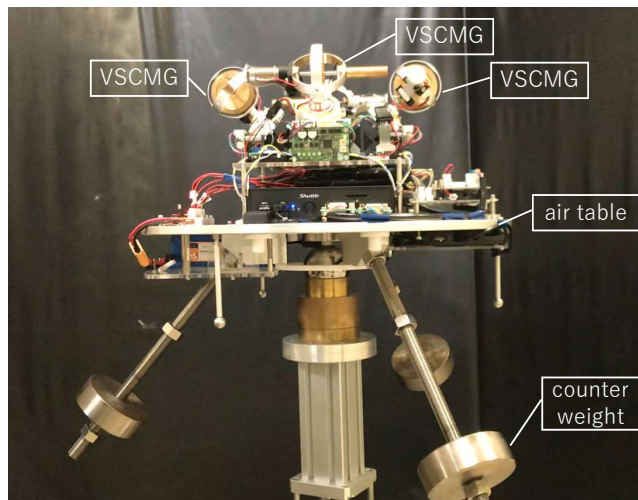


Figure 5.2: Experimental setup located at Osaka University [22]

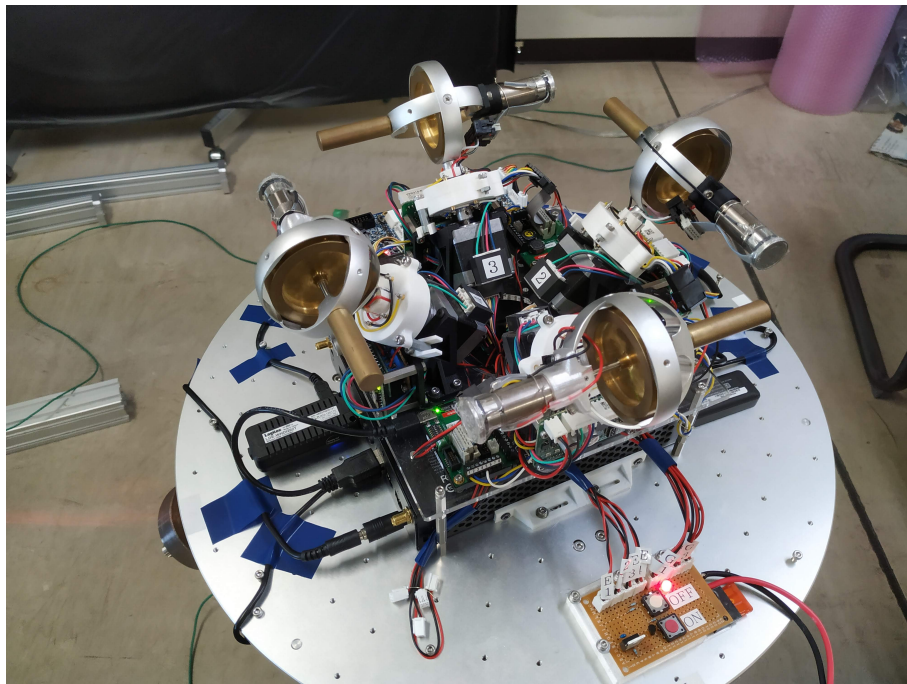


Figure 5.3: Experimental setup located at Osaka University

5.2 Experimental results and C++ conversion

After having implemented the three types of MPC previously explicated, has been decided to utilize the MPC with only the Mode 2 for the experiments. The motivations to implement this MPC with respect to the complete MPC with Mode 1 and 2 come from the difficult definition of η_s . This parameter is the one that in the control algorithm chooses if either the Mode 1 or the Mode 2 must be applied. During the simulations was quite difficult to choose its optimal value, a small variation of its value with respect to the optimal one could compromise the control performance of the system.

Since it was expected a difference, even small, between the behavior of the model used in the simulations and the one of the testbed, it would be necessary to perform different experiments to find out the optimal value of η_s . For this reason the next experiment results have been obtained utilizing the MPC without the integrator state reset.

5.2.1 Conversion of the control algorithm in C++

After having performed the simulations in MATLAB environment, to perform the experiments has been necessary to convert the decided control algorithm in C++. For this purpose MATLAB Coder Toolbox has been used to convert directly the control algorithm in C++.

After, the integrated development environment called Microsoft Visual Studio has been used to create a C++ project in which the converted control algorithm has been tested.

Initially the mathematical model of the testbed has been used to perform a software-in-the-loop simulation as in the Matlab environment, to verify the effectiveness of the converted control algorithm. The results obtained in Visual Studio were perfectly identical to the ones obtained in MATLAB, confirming that the converted algorithm was effective as the original one.

Finally, the final project in Visual Studio has been created to test the control algorithm on the real testbed. It is important to remember that the plant in the control scheme has been substituted by the real testbed, but the mathematical model defined in Chapter 3 was still inside the control algorithm, to predict the future behavior of the testbed.

5.2.2 Experimental results

Different experiments have been performed, but here are reported only the better results I had obtained during the stay at Osaka University. As expected the gravity effect acted as a disturbance for the control system, hence, in order to relax the tracking performance for the testbed, the reference signal has been changed

extending the overall time for the triangular shape of the vertical angular velocity reference. Furthermore, many different values for the weight matrices has been tested, relaxing the tracking performance required for the testbed.

In the following images are reported the actual values of the variables with a blue line; in red are reported the reference signals; and in yellow the virtual reference signals.

In Figure 5.4 is reported the vectorial part of the quaternion. In Figure 5.5 are reported the angular velocity components of the testbed. From output plots arises an elevate oscillatory behaviour, in particular in the horizontal components of the angular velocity, probably due to the gravity effect. Indeed, when misalignment occurs, the gravity acts on the testbed causing the inclination of the floating table. At this point, the controller tries to compensate this effect creating a large control input, but this brings to an inclination on the opposite side. However, the tracking performance of the vectorial part of the quaternion is sensibly better.

In Figure 5.6 are reported the control inputs of the system. In Figure 5.7 is shown

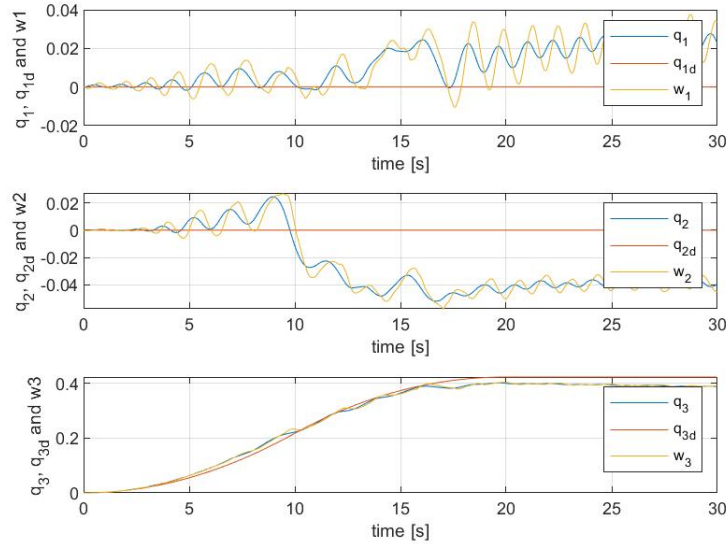


Figure 5.4: Vectorial part of the quaternion obtained from the experiment

the gimbal angles evolution of the testbed. And it is important to notice that in this case the gimbals don't return to a value equal to 0, as happened in the second and third simulation. Finally, in Figure 5.8, are reported the four quaternion components, where is shown that the unit quaternion condition is respected.

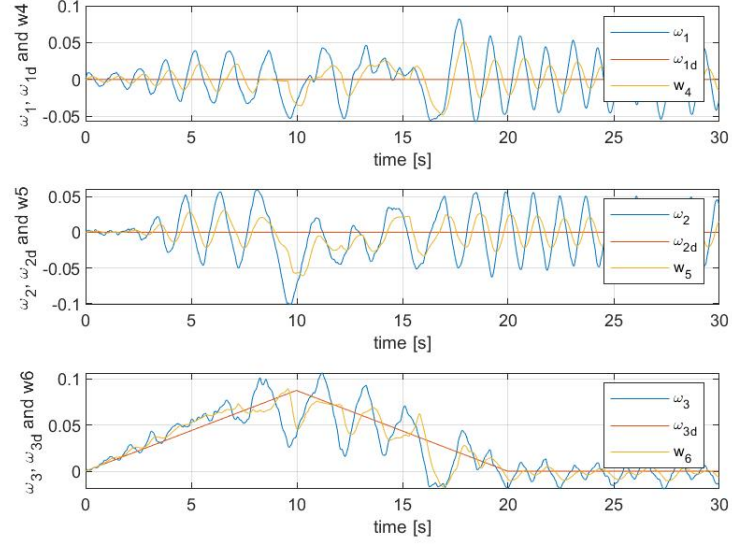


Figure 5.5: Angular velocity components of the testbed obtained from the experiment

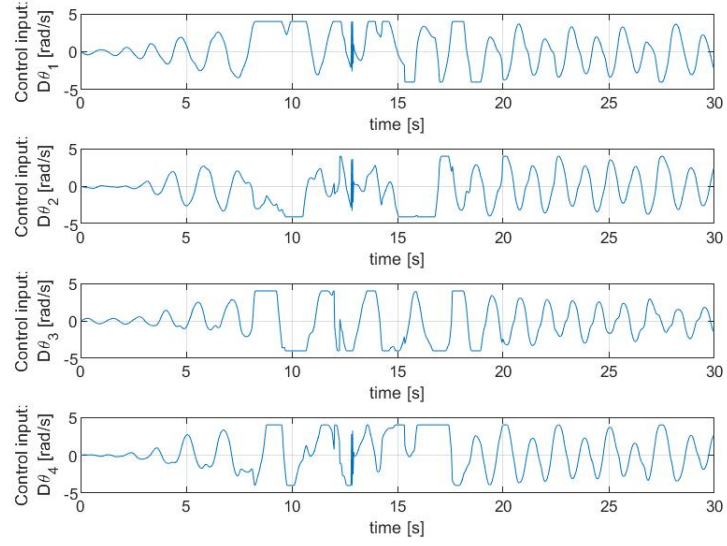


Figure 5.6: Control input signals obtained from the experiment

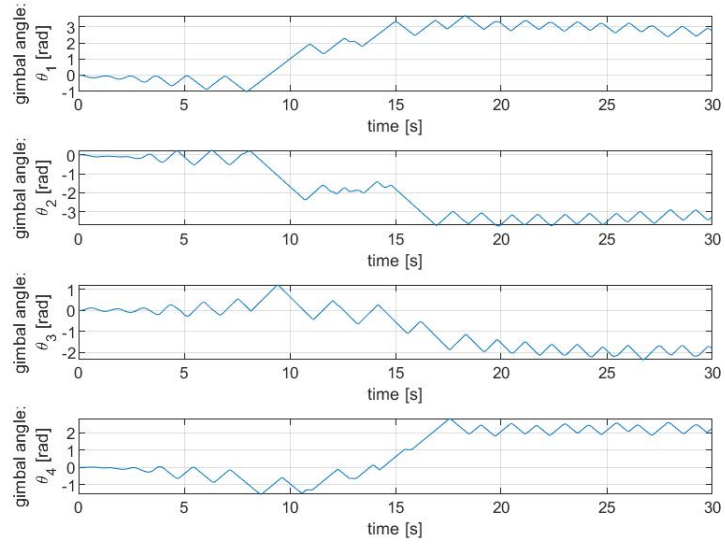


Figure 5.7: Gimbal angles evolution obtained from the experiment

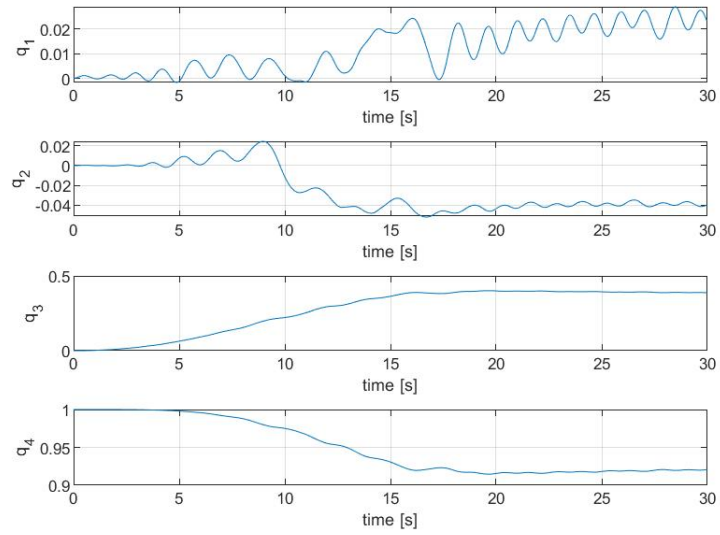


Figure 5.8: Quaternion components obtained from the experiment

Chapter 6

Conclusions and possible future works

In this thesis work the implementation of a linear MPC with virtual reference and integrator state optimization has been analysed for the attitude control of testbed which simulates a spacecraft.

This final essay has presented an introduction where several past related works have been cited, showing the objectives and results achieved.

In Chapter 2, the classic linear MPC approach has been briefly described. After, the linear MPC with virtual reference and integrator state optimization has been described accurately.

In Chapter 3, the dynamic and kinematic equations have been used to construct the mathematical model of the system in an LTI state space formulation.

Three different simulations have been performed in MATLAB, highlighting the benefits brought by the optimization of the virtual reference signal and of the integrator state.

In the first simulation, a classic MPC has been implemented. The results obtained by this simulation, in which just the control input is obtained by the optimization procedure, show a growing oscillatory behaviour in the output signals. This observed trend can compromise the stability of the control system.

The second simulation has been performed implementing an MPC where both the control input and the virtual reference signal are optimized. The results obtained by this simulation have highlighted the benefits brought by the implementation of this further variable. The virtual reference signal at some time during the simulation departs from the real reference signal, creating an initial difference between the actual values of the output variables and real references to follow, but in this way is obtained a decreasing oscillatory behaviour of the output variables.

From the last simulation, also the integrator state has been included in the optimization procedure. With this further characteristic, a better response has been

obtained with respect to the previous simulation, because the oscillations are attenuated faster.

After, the control algorithm has been converted in C++ programming language using MATLAB Coder Toolbox and it has been verified on the real testbed performing experiments. The results obtained by these experiments show an unstable behaviour of the testbed, which presents quite large oscillations along the horizontal components of the angular velocity. This behaviour is probably due to the gravity effect, as explained in the previous chapter. The problem of this experimental setup is that some misalignments between the center of rotation and the center of gravity can happen, creating quite large gravity torques even in presence of small misalignments. To minimize this gravity effect, the experimental setup is equipped with three counter weights, which are attached to the testbed by ball screws. The position of the three counter weight must be adjusted manually, therefore it is very complicated to obtain a perfect alignment of the center of gravity and of the center of rotation.

But another source of instability can be also the adopted mathematical model used inside the MPC control algorithm to predict the future behaviour of the testbed. If this model doesn't represent sufficiently well the testbed, the prediction cannot be accurate and consequently the control inputs too.

A future step of this thesis work is obviously the improvement the experimental results. To this end, a first step could be the redefinition of the model of the system, to obtain a more similar behaviour to the real testbed. Moreover, a dedicated control system to correct the error given by the gravity effect could be designed and implemented on the considered testbed, in order to create conditions more similar to the vacuum space.

The study of the attitude control of the considered testbed can be carry forward investigating the implementation of a more robust Stochastic control strategy. To this end, the Stochastic MPC able to deal with disturbances and uncertainties, treated in [1], should be designed for the testbed. In this way the gravity effect can be treated as disturbance for the control system, improving the tracking performance of the control system. In Section 2.4 , some primary calculations used to implement the considered Stochastic MPC are performed. Next steps are to perform simulations and experiments with this Stochastic control strategy.

Bibliography

- [1] M. Mammarella, M. Lorenzen, E. Capello, H. Park, F. Dabbene, G. Guglieri, M. Romano, and F. Allgöwer, “An offline-sampling smpc framework with application to autonomous space maneuvers,” *IEEE Transactions on Control Systems Technology*, 2018.
- [2] D. Miceli, “Analisi di un sistema avionico basato su control moment gyro,” 2007.
- [3] M. Canale, “Automatic control systems,” 2017.
- [4] A. Bemporad and M. Explicit, “Model predictive control of hybrid systems,” 2; *HYCON Phd On Hybrid Systems, Sienna 18/7/2007*, 2007.
- [5] G. POPESCU, “Pixel geolocation algorithm for satellite scanner data,” *Scientific Papers. Series E. Land Reclamation, Earth Observation & Surveying, Environmental Engineering, Vol. III, Bucharest*, pp. 127–136, 2014.
- [6] B. Wie, “Singularity analysis and visualization for single-gimbal control moment gyro systems,” *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 2, pp. 271–282, 2004.
- [7] K. Srinivasan, D. Gandhi, and M. Venugopal, “Spacecraft attitude control using control moment gyro reconfiguration,” *Studies in Informatics and Control*, vol. 23, no. 3, p. 286, 2014.
- [8] EASAC, *European space exploration: strategic considerations of human versus robotic exploration*. No. August, 2014.
- [9] E. National Academies of Sciences and Medicine, *NASA Space Technology Roadmaps and Priorities Revisited*. Washington, DC: The National Academies Press, 2016.
- [10] I. Courie, Y. Jenie, and R. Poetro, “Simulation of satellite attitude control using single gimbal control moment gyro (sgcmg) system,” in *Journal of Physics: Conference Series*, vol. 1130, p. 012003, IOP Publishing, 2018.
- [11] R. Yadlin, *Stochastic model predictive control and its application for small satellite attitude control*. PhD thesis, Oxford University, UK, 2011.
- [12] K. Ashok, R. Aditya, H. K. Joglekar, M. Siva, and N. Philip, “Cmg configuration and steering approach for spacecraft rapid maneuvers,” *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 118–123, 2016.
- [13] Y. Ikeda, T. Nakajima, and Y. Chida, “Attitude control of spacecraft by nmpe

- with consideration of singularity avoidance of cmg,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 1733–1739, IEEE, 2012.
- [14] M. S. M. Cavalca, R. K. H. Galvão, and T. Yoneyama, “Integrator resetting with guaranteed feasibility for an lmi-based robust model predictive control approach,” in *18th Mediterranean Conference on Control and Automation, MED’10*, pp. 634–639, IEEE, 2010.
- [15] M. Mammarella, E. Capello, F. Dabbene, and G. Guglieri, “Sample-based smpc for tracking control of fixed-wing uav,” *IEEE control systems letters*, vol. 2, no. 4, pp. 611–616, 2018.
- [16] N. Wada, “Model predictive tracking control for constrained linear systems using integrator resets,” *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 3113–3118, 2015.
- [17] M. Ciavola, E. Capello, S. Satoh, and Y. Katsuhico, “Comparison of SDA and MPC controllers of a CMG system for the singularity problem,” in *Proceedings of the SICE Annual Conference 2019, September 10-13, 2019, Hiroshima, Japan*, pp. 541–544, SICE, 2019.
- [18] N. Wada and S. Tsurushima, “Constrained mpc to track time-varying reference signals: Online optimization of virtual reference signals and controller states,” *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 11, pp. S65–S74, 2016.
- [19] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*. Nob Hill Pub., 2009.
- [20] K. Großkatthöfer and Z. Yoon, “Introduction into quaternions for spacecraft attitude representation,” *TU Berlin*, vol. 16, 2012.
- [21] J. Currie, *Practical applications of industrial optimization: from high-speed embedded controllers to large discrete utility systems*. PhD thesis, PhD Thesis, Dept. Elect. Eng., AUT University, 2014.
- [22] D. Higashiyama, Y. Shoji, S. Satoh, and I. Jikuya, “Attitude Control for Spacecraft Using Pyramid-Type Variable-Speed Control Moment Gyros,” 2019.