POLITECNICO DI TORINO

Facoltà di Ingegneria
Corso di Laurea in Ingegneria Aerospaziale

Tesi di Laurea Magistrale

Analysis and comparison of SDA and MPC controllers in a four CMGs system

Relatori:
prof. Elisa Capello
prof. Satoshi Satoh

Candidato:
Martina Ciavola

Anno accademico 2019-2020

Notations

For a matrix $\boldsymbol{F} \in \mathbb{R}^{n\times m}$, F^l denotes the l-th row of F.

For a vector $\boldsymbol{x} \in \mathbb{R}^n$, we define $\|\boldsymbol{x}\|_2 := (\boldsymbol{x}^T\boldsymbol{x})^{1/2}$.

For a vector $\boldsymbol{x} \in \mathbb{R}^n$ and positive definite matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$, we define $\|\boldsymbol{x}\|_P := (\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x})^{1/2}$.

For a symmetric matrix $\boldsymbol{P}$, $\boldsymbol{P} > 0$ ($\boldsymbol{P} \geq 0$) implies that $\boldsymbol{P}$ is a positive definite (positive semidefinite) matrix.

For vectors $\boldsymbol{u}$, $\boldsymbol{v} \in \mathbb{R}^n$, the notation $\boldsymbol{u} < \boldsymbol{v}$ ($\boldsymbol{u} \leq \boldsymbol{v}$) implies $u^{(i)} < v^{(i)}$ ($u^{(i)} \leq v^{(i)}$), $\forall i = 1, \ldots, n$.

For vectors $\boldsymbol{u}$, $\boldsymbol{v} \in \mathbb{R}^n$, the notation $|\boldsymbol{u}| < \boldsymbol{v}$ ($|u| \leq \boldsymbol{v}$) implies $|u(i)| < v(i)$ ($|u(i)| \leq v(i)$), $\forall i = 1, \ldots, n$.

For a positive definite matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$, a vector $\boldsymbol{v} \in \mathbb{R}^n$ and a positive scalar η, we define the set $\mathcal{E}(\boldsymbol{P}, \boldsymbol{v}, \eta) := \{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{x} - \boldsymbol{v})^T\boldsymbol{P}(\boldsymbol{x} - \boldsymbol{v}) \leq \eta\}$

Abstract

In this thesis, the attitude dynamics of a system equipped with a cluster of four Control Moment Gyros (CMGs) in a pyramidal configuration is analyzed. CMGs are angular momentum exchange devices used for precise attitude control, since they are able to generate larger torque compared to Reaction Wheels (RWs). At least three gimbals are necessary for full three-axis control, but a minimum of four CMGs is usually employed to increase system redundancy. Moreover, one important difficulty related to the use of CMGs is their singularity problem in which, under a certain configuration of CMG gimbal angles, no control torque can be generated. For this reason, in order to develop the equations of spacecraft attitude dynamics, the dynamics of a rigid body is studied. Thus, starting from the external control torque and the angular momentum, whenever using momentum devices for attitude control, it can be useful to write the equations in terms of the angular velocity, in order to apply the controller considering the gimbal angles as control inputs. Furthermore, the quaternion kinematic equation is used to describe the spacecraft reorientation in terms of quaternions and the first dynamic test is made in order to check the dynamics implementation. Consequently, two controllers are proposed to avoid the CMGs singularity: Singular Direction Avoidance (SDA) steering law and a variation of Classic Model Predictive Control (CMPC). The first method provides the ability to avoid or escape the singularity with a particular finite gimbal rate. Nevertheless, since the SDA steering law uses the error between the desired and the current attitude, the tracking performance is slightly degradated during the escape from the singularity. This controller is implemented for both simulation and experimental results. The experiments are conducted at the *Yamada Laboratory, Graduate School of Engineering, Osaka University, Osaka, Japan*. The second method is generally used for constrained problems as it can provide closed-loop stability and feasibility at each time step by ensuring that the predicted trajectory satisfies the constraints. MPC is a more complex controller with a higher computational cost, which consists in a minimization of a cost function in order to get the control vector to be applied to the attitude dynamics. The effectiveness of this controller is shown comparing the tracking performance in the simulation results.

Sommario

In questa tesi viene proposta un'analisi della dinamica d'assetto di un sistema equipaggiato con quattro Control Moment Gyro (CMG) in configurazione piramidale. I CMGs sono dispositivi per lo scambio di momento angolare usati per manovre di precisione di controllo d'assetto, essendo in grado di generare un valore di momento molto più alto rispetto alle Reaction Wheels (RWs). Tre CMGs sono sufficienti per il completo controllo d'assetto ma la loro applicazione consiste generalmente nell'utilizzo di almeno quattro CMGs per questioni di ridondanza. Inoltre, un'importante difficoltà relativa all'utilizzo dei CMGs è il problema della singolarità, in base al quale nessun momento di controllo può essere generato quando i *gimbals* dei CMGs si trovano in una particolare configurazione angolare. Per questo motivo, la dinamica del corpo rigido è stata inizialmente studiata per ricavare le equazioni della dinamica d'assetto di uno spacecraft. Quindi, partendo dal momento esterno applicato e dal valore del momento angolare, è utile esprimere le equazioni in termini di velocità angolare per poter applicare il controllore scelto utilizzando gli angoli dei *gimbals* come input di controllo. Successivamente, l'orientazione dello spacecraft viene descritta utilizzando la cinematica dei quaternioni così da poter svolgere il primo test dinamico per appurare la funzionalità della dinamica implementata. A questo punto, per evitare la singolarità dei CMGs, vengono proposti due controllori: il Singular Direction Avoidance (SDA) steering law e una variante del Classic Model Predictive Control (CMPC). Il primo metodo consiste nell'evitare la singolarità dando ai *gimbals* un valore finito di velocità angolare. Nonostante ciò, poichè il metodo SDA utilizza l'errore tra la dinamica attuale e quella desiderata, si osserva un leggero degradamento delle performance in vicinanza della singolarità. Il controllore SDA è implementato per ottenere sia risultati in simulazione che sperimentali. Gli esperimenti sono stati svolti sul testbed presente allo *Yamada Laboratory, Graduate School of Engineering, Osaka University, Osaka, Japan.* Il secondo metodo (MPC) viene generalmente utilizzato per problemi vincolati in quanto riesce a garantire stabilità e fattibilità ad ogni step temporale, assicurando che la traiettoria desiderata soddisfi i suddetti vincoli. Si noti che MPC è un metodo di controllo molto più complesso e che richiede un più elevato costo computazionale, essendo necessaria la minimizzazione di una funzione di costo per ottenere il vettore di controllo che verrà poi applicato alla dinamica d'assetto. L'efficacia del secondo metodo presentato è dimostrata attraverso i risultati di simulazione.

Contents

List of Figures

Chapter 1

Introduction

The benefits of feedback control have been known to mankind for many years, but the formal development as a mathematical tool for the analysis of the behavior of dynamic systems is much more recent. It began 150 years ago with Maxwell works and then continued with all the developments promoted by Laplace, Lyapunov, Bode, Bellman who are a few of the major contributors to the edifice of control theory. Particular interest, both from theoretical and practical point of view, has optimal control theory, which addresses the problem of optimizing a cost index that measures system performance through the choice of system parameters that are designated as control inputs. This work provides a systematic approach to the design of strategies that achieve optimal performance.

During the years, attitude control systems based on Control Moment Gyros CMG have been used for a large scale of satellite as they can generate a large torque compared to Reaction Wheels systems. A CMG is an angular momentum exchange device used to produce control torque for spacecraft attitude control. It consists of a spinning rotor and one or more motorized gimbals that tilt the rotor's angular momentum: as the rotor tilts, the changing angular momentum causes a gyroscopic torque that rotates the spacecraft realizing redundant three-axis 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. When the singularity occurs under certain configurations of CMG gimbal angles, no control torque in a specific direction can be achieved. Many CMG singularity avoidance methods have been developed in literature in order to overcome this problem of implementation, each one with a respective steering law that can lead to some advantages and disadvantages, though torque errors can still be found due to the singularities. Thus, gain scheduling designs are often proposed for Linear Time-Varying (LTV) systems, even if there is a limitation

due to the high number of independent time-varying variables for spacecraft control using CMGs. For this reason, gain scheduling design may be unfeasible. An example of CMG singularity avoidance method is Singular Direction Avoidance (SDA) steering law: when the commanded torque is parallel to the singular direction, zero gimbal angular rates would result from using SDA. This method is discussed in this thesis.

In the last decades, the issues of feasibility of the online optimization and stability of the closed loop system have been extensively studied. The optimal control solution has the particularly 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 [2]. This is applied to both continuous time systems described by sets of differential equations and to discrete time systems formulated in terms of difference equation models. The benefits of optimal control are, however, difficult to achieve in the case of systems with nonlinear models and systems that are subject to constraints on input variables or model states. For both this cases, in general, it is not possible to derive analytic expressions for the optimal control solution but one might hope that optimal solutions could be computed numerically. However, the associated optimization problem is difficult to solve for all but the simpliest cases and approximate solutions have to be considered. This leads to one of the most important reasons behind the phenomenal success of Model Predictive Control (MPC).

MPC is probably the most widely accepted modern control strategy as it offers a sensible compromise between optimality and speed of computation, thanks to its receding horizon implementation. It works predicting future behaviors using a system model, given measurements or estimates of the current state and a hypothetical future input trajectory or feedback control policy. Thus, future inputs are characterized by a finite number of DOF used to optimize a predicted cost and the process is repeated at each time step using updated information on the system state in order to introduce feedback to this strategy. In closed-loop operations, this repetition reduces the gap between the predicted and the actual system response and provides a certain robustness to the uncertainty that can arise from unknown variations in the model parameters as well as disturbances appearing additively in the system dynamics. Considering a system in the absence of any uncertainty, MPC strategy may take account of predicted behavior over a finite horizon only and this lacks guarantees of nominal stability. This problem is overcome by imposing additional conditions as terminal constraints on the predicted model states in order to ensure that the desired steady state is reached at the end of a finite prediction horizon. Therefore, stability and convergence properties are ensured as effects of

these constraints that render a finite horizon equivalent to an infinite one. A stabilizing feedback law, known as terminal control law, could be used to define the predicted control inputs at future times beyond the initial, finite prediction horizon which presents computational challenges in the case of nonlinear systems. This feedback law is often used for the actual system dynamics in the absence of constraints or it can be chosen for the unconstrained, linearized dynamics about the desired steady state. Thus, additional constraints known as terminal constraints are imposed in order to ensure that the control strategy meets the system constraints, ensuring feasibility of the receding horizon strategy. These terminal constraints require that the system state at the end of the initial finite prediction horizon should belong to a subset of state space with the property that once entered the state of the constrained system will never leave it. In the dual-mode approach, the open-loop optimal control is initially applied, and then a terminal control law is utilized after the state variable reaches a positively invariant set. This dual-mode MPC strategy is widely accepted as one of the most systematic approaches to design an MPC that guarantees feasibility and stability. All the ideas above are included in the Classic Model Predictive Control (CMPC) theory, which has the advantages to be stable, to meet constraints and to converge asymptotically to the desired steady state in closed loop operation. However, it may be required that the controller has an acceptable degree of robustness to model uncertainty. We refer to this more challenging control problem as Robust Model Predictive Control (RMPC), in which the uncertainty has known bounds but no further information is assumed. The concern for RMPC would be to guarantee stability, constraint satisfaction and convergence of the state vector to given steady-state condition or set of states, for all possible realizations of uncertainty. However, uncertainty is subject to some statistical regularity and can be modelled as random but with known probability distribution and some or all of the constraints may be probabilistic in nature. We refer to this control problem as Stochastic Model Predictive Control (SMPC). The object of SMPC is to ensure that such constraints, together with any additional hard constraints that may be present, are met in closed loop operation, and to simultaneously stabilize the system.

In order to cope with the disadvantage of significant performance degradation in the presence of uncertainty and disturbances, RMPC has received a great deal of attention for both linear systems and nonlinear systems. It is sometimes possible to formulate a stochastic model to represent the uncertainty and disturbance, as for instance in the case of inflow material in a chemical process or wind speed and turbulence in wind turbine control, which have led to an increasing interest in SMPC. A probabilistic description of the disturbance or uncertainty allows to optimize the average performance or appropriate risk measures. However, the first problem in SMPC is the derivation of computationally tractable methods in order to propagate the uncertainty for evaluating the cost function. The calculation has multivariate

integrals, whose evaluation requires the development of suitable techniques. A second problem in SMPC is related to the difficulty of establishing recursive feasibility, which is essential for stability. In CMPC recursive feasibility is usually guaranteed through showing that the planned input trajectory remains feasible in the next optimization step, while the idea is then extended to RMPC by requiring that the input trajectory remains feasible for all possible disturbances. In SMPC a certain probability of future constraint violation is in general allowed, which leads to significantly less conservative constraint tightening for the predicted input and state. However, in this setup, the probability distribution of the state prediction at some future time depends on both the current state and the time to go so that even under the same control law, the violation probability changes at each sampling time and this might render the optimization problem unfeasible. An exact evaluation of the desired quantities is in general only possible in a linear setup with Gaussian noise or finitely supported uncertainties, as explained in *"Stabilizing model predictive control of stochastic constrained linear systems"* [3]. These approximate solutions include particle approaches and polynomial chaos expansion, as we can see respectively in [4] and [5]. For linear systems with parametric uncertainty, *"Stochastic nonlinear model predictive control with probabilistic constraints"* [6] proposes to decompose the uncertainty tube into a stochastic part computed offline and a robust part computed online. If there is an additive disturbance, the system is usually decomposed into a deterministic, nominal part and an autonomous system involving only the uncertain part in order to reduce the computational cost.

A scenario for MPC applications is Autonomous Rendezvous and Docking (ARVD) maneuvers, which have been widely studied in the recent years, as it is expected that most of the proximity operations will be performed autonomously with high performance and on-board processors. Furthermore, this maneuvers have been gaining relevance due to the complex flight scenario which includes constraints in both the mass and size that can be transported in a launching vehicle (depending on the mission) as well as the growing concern about space debris that has been accumulating since the beginning of space exploration. Different control techniques have been proposed for ARVD maneuvers, including MPC due to its strategy in dealing with constraints, in terms of relative position and velocity, and its degree of robustness to system uncertainties due to its receding horizon implementation. Examples can be found in Linear Quadratic Model Predictive Control (LQMPC) adopted to enforce thrust magnitude limitation, velocity constraints for soft docking, low thrust rendezvous and proximity operations, not taking into account, in all of these approaches, orbital perturbations, disturbances and model errors. Moreover, since the phase of proximity involves highly nonlinear kinematics and dynamics, linear control methodologies have limited performance achievement if the linearization errors are not taken into account. For these reasons, in *"A general sampling-based SMPC*

approach to spacecraft proximity operations" [7], an SMPC controller able to perform the docking between the chaser and the target in an experimental setup [1] is proposed, taking into account both parametric uncertainties and external additive disturbances (due to variations of the spacecraft parameters during the flight) and reducing the computational effort with respect to other SMPC techniques, thanks to the offline evaluation of the feedback gain matrix. Indeed, sampling-based SMPC methods have a high computational cost, which may render these approaches not easily implementable in real applications. The choice of adopting a stochastic approach instead of a robust one is motivated by the fact that the type of uncertainties generally found in ARVD operations have in general a probabilistic description, and the trajectory constraints imposed to the spacecraft may allow a probability of violation.

Recently, MPC has been extended to tracking control problems. Basically, the setpoint tracking control can be achieved by regulating the state to the target state. Also, by inserting the integrator into the feedback loop, it would be possible to achieve zero steady state error in the presence of step disturbances and/or modeling errors. Further, several attempts have been made to extend MPC to tracking control for time-varying reference signals. In many tracking control strategies, the control algorithm is reduced to a convex optimization problem under Linear Matrix Inequality (LMI) constraints. In *"Constrained MPC to Track Time-Varying Reference Signals: Online Optimization of Virtual Reference Signals and Controller States"* [8] it is shown that this method is an effective means to improve the tracking control performance of an MPC system. However, the control method is applicable only to the case where a reference signal is constant so that if a reference signal is time-varying, closed loop stability as well as feasibility of the control algorithm may not be guaranteed. In actual control systems, we frequently encounter a system with control signals constrained to a finite region whose bounds are time-varying. For example, the tire force limitation in a vehicle control system [9] can be modeled as a time-varying saturation nonlinearity imposed on the control signal. Also, the input transformations used to simplify the control system design often cause time-varying input saturation on the new control signal. If the standard MPC is applied to such a system, feasibility and stability may not be guaranteed. In *"Stabilized MPC formulation for robust reconfigurable flight control"* [10], it is shown that the MPC algorithm, with a target recalculation mechanism, is an effective control strategy in dealing with such a control problem through various numerical examples on a realistic nonlinear aircraft model. However, the control algorithm shown in [10] may also not guarantee feasibility and stability of the control system.

[1] developed at the Naval Postgraduate School (NPS) of Monterey, called 'Proximity Operation of Spacecraft: Experimental hardware In-the-loop DYNamic simulator (POSEIDYN)'

The main idea of this thesis is to design and implement two different controllers, SDA and CMPC, for a CMG-based testbed, in which uncertainties and disturbances are also included. As in [11] and [12] the MPC controller is proposed including LTV system, starting from the theory of pole placement. In a similar way, in [13] external disturbances are also included and a Nonlinear Model Predictive Control (NMPC) is used to implement the attitude feedback control of the integrated system of the spacecraft. In this research we use a cost function that includes the norm of the deviation between the reference signal and the virtual reference signal. In the proposed algorithm, the controller state as well as the control sequence and the virtual reference signal are determined at each sampling time so that the cost function is minimized. The main advantages of using MPC controllers with CMG systems are related to two main aspects: the first one is that MPC controller can directly calculate the angular velocity of the gimbals without calculating the external control torque; the second one is that the input, state and hardware constraints can be included in the implementation. Indeed, MPC is chosen alongside SDA because it can take constraints into account, while this cannot be done by SDA because of the singularity problem. As a consequence of the implementing procedure, if the angular velocity of the gimbals is chosen as control input, the stability of the closed loop system and its feasibility are guaranteed, as well as the convergence of the tracking error, while even the tracking performance is improved under input constraints. This feature guarantees the computational efficiency of the proposed controllers and the performance in the experimental tests.

Chapter 2

Spacecraft dynamics

2.1 Rigid-body dynamics

The first step of the analysis is to consider the dynamics of a rigid body as a basis for developing the equations of satellite attitude dynamics. According to French mathematician Michel Chasles's (1793-1880) theorem, the motion of a rigid body can be described by the displacement of any point of the body (the base point) plus a rotation about a unique axis through that point.The magnitude of the rotation does not depend on the base point. For this reason, a rigid body in a general state of motion has an angular velocity vector whose direction is that of the instantaneous axis of rotation. In order to describe the rotational component of the motion of a rigid body in three dimensions, it is necessary taking advantage of the vector nature of angular velocity and knowing how to take the time derivative of moving vectors. Thus, it is possible to analyze the interaction between the motion of a rigid body and the forces acting on it. In the two body problem, if we consider only the traslational component of the motion, we can concentrate all of the mass at the center of mass and then apply the methods of particle mechanics to determine its motion. Analyzing the rotational dynamics requires computing the body's angular momentum, and that in turn requires accounting for how the mass is distributed throughout the body: this mass distribution is described by the six components of the moment of inertia tensor. Thus, from these considerations, it is possible to find the nonlinear Euler's equations of motion that are used to first express our dynamics.

2.2 Reference frames

In the following analysis two reference frames are used to describe the spacecraft dynamics: inertial reference frame and body reference frame. The purpose of this

section is to define these two reference frames used. For the transformation from ECI to Body reference frame please refer to A.

2.2.1 Inertial Frame

The first reference frame considered is the Earth-Centered Inertial (ECI) that is the so called Inertial Geocentric Reference Frame. It is commonly used to study the motion of a body orbiting around Earth, referred to a pseudo geocentric inertial frame, with the axis oriented to the fixed stars. Is has its origin in the center of mass of Earth. Thus, the x axis $\hat{x_I}$ is on the equatorial plane, oriented towards the mean vernal equinox at J2000 epoch, the z axis $\hat{z_I}$ is aligned with Earth spin axis (celestial North Pole) and the y axis $\hat{y_I}$ completes the right-handed frame. Using the same formulation as [14]

$$F_{ECI} = (x_{ECI}, y_{ECI}, z_{ECI}) \tag{2.1}$$

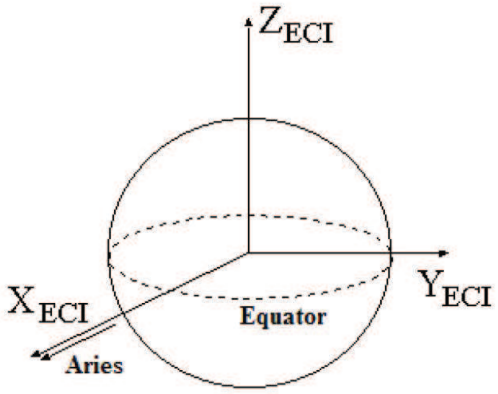


Figure 2.1: Earth Centered Inertial Reference Frame

The actual rotation of Earth θ referred to the inertial reference frame ECI can be computed as

$$\theta = \theta_0 + \Omega_E(t - t_0) \tag{2.2}$$

In this equation, θ_0 is the Earth rotation at time t_0, t is the actual time and, lastly, $\Omega_E = 7.272 \cdot 10^{-5} rad/s$ is the angular velocity of Earth about Z_{ECI}.

2.2.2 Body Frame

The second reference frame used is the Body Frame. It has been adopted to describe the attitude dynamics once introduced the CMGs to the equations of motion. This frame has a body-fixed reference, with axis bound to the spacecraft during its motion. The origin of the Body Frame is in the center of mass of the spacecraft. If we consider principal axis of inertia, the Body Frame has the axis oriented along the spacecraft principal axis of inertia.

$$F_{B(P.I.)} = (x_{B(P.I.)}, y_{B(P.I.)}, z_{B(P.I.)}) \tag{2.3}$$

The $z_{B(P.I.)}$ direction is positive from the lower to the upper side of the spacecraft, the $x_{B(P.I.)}$ is orthogonal to $z_{B(P.I.)}$ and aligned with the solar panels direction and the $y_{B(P.I.)}$ completes the right-handed triad.

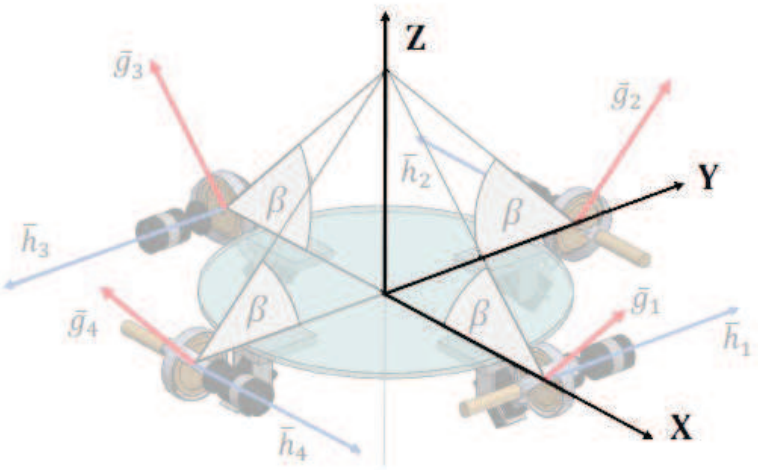


Figure 2.2: Body Reference Frame in CMG cluster

2.3 The inertia tensor

The angular momentum $\partial \boldsymbol{h}$ of a mass element ∂m, moving with velocity $\boldsymbol{v}$ is

$$\partial \boldsymbol{h} = \boldsymbol{r} \times (\partial m \boldsymbol{v}) \tag{2.4}$$

where $\boldsymbol{r}$ is the position vector of the mass, with respect to the body reference frame. For an extended rigid body as shown in Fig. 2.3 (from Ref. [1]), total angular

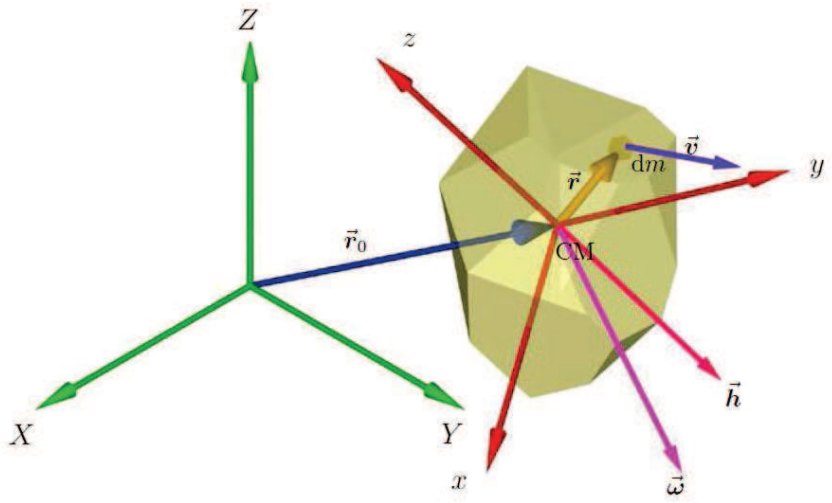


Figure 2.3: Rotating rigid body (Ref. [1])

momentum is given by

$$\boldsymbol{h} = \int_B (\boldsymbol{r} \times \boldsymbol{v}) \partial m \tag{2.5}$$

Proceeding with the analysis as in [1], if the body is rotating around its center of mass, the velocity of every mass element is

$$\boldsymbol{v} = \boldsymbol{\omega} \times \boldsymbol{r} \tag{2.6}$$

so that

$$\boldsymbol{h} = \int_B [\boldsymbol{r} \times (\boldsymbol{\omega} \times \boldsymbol{r})] \partial m \tag{2.7}$$

The integration over the body B of $[\boldsymbol{r} \times (\boldsymbol{\omega} \times \boldsymbol{r}]\partial m$ is strictly function of the mass distribution only, as angular velocity components are independent of body shape and location. Thus, considering

$$\boldsymbol{h} = h_1 \hat{\boldsymbol{x}} + h_2 \hat{\boldsymbol{y}} + h_3 \hat{\boldsymbol{z}} \tag{2.8}$$

it is

$$h_1 = \quad I_x \omega_1 - I_{xy} \omega_2 - I_{xz} \omega_3 \tag{2.9}$$
$$h_2 = -I_{xy} \omega_1 + I_y \omega_2 - I_{yz} \omega_3 \tag{2.10}$$
$$h_3 = -I_{xz} \omega_1 - I_{yz} \omega_2 + I_z \omega_3 \tag{2.11}$$

where the moments of inertia I_x, I_y, I_z, and the products of inertia I_{xy}, I_{xz}, I_{yz} are

$$I_x = \int_B (y^2 + z^2)\partial m; \quad I_y = \int_b (x^2 + z^2)\partial m; \quad I_z = \int_B (x^2 + y^2)\partial m;$$
$$I_{xy} = \int_B (xy)\partial m; \qquad I_{xz} = \int_B (xz)\partial m; \qquad I_{yz} = \int_B (yz)\partial m;$$

so that the equation of the angular momentum in matrix form can be written as

$$\boldsymbol{h}_B = \boldsymbol{I}\boldsymbol{\omega}_B \tag{2.12}$$

The inertia matrix $\boldsymbol{I}$ can be explicated with the moments of inertia and the products of inertia written above and represents the *inertia tensor* in body axis

$$\boldsymbol{I} = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix} \tag{2.13}$$

The matrix $\boldsymbol{I}$ is real and symmetric, so its eigenvalues are real and its eigenvectors are mutually orthogonal. This means that there exists a body reference frame F_P such that the inertia matrix is diagonal

$$\boldsymbol{I} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \tag{2.14}$$

where the *principal moment of inertia* J_x, J_y and J_z are the eigenvalues of $\boldsymbol{I}$ and the related eigenvectors are called *principal axis*. From he following section $\hat{\boldsymbol{I}}, \hat{\boldsymbol{J}}, \hat{\boldsymbol{K}}$ is the notation used for the inertial reference frame and $\hat{\boldsymbol{i}}, \hat{\boldsymbol{j}}, \hat{\boldsymbol{k}}$ is the notation used for the body reference frame.

2.4 Euler's equation of motion

When using a momentum device for attitude control it can be useful to write the attitude equation of motion in terms of angular momentum. The common approach used to analyze the dynamics of a rigid body refers to Euler's equation of motion which states that the time derivative of the angular momentum is equal to the total external torque applied to the body. In vector form, in an inertial reference frame indicated with $^\circ$

$$\frac{d^\circ \boldsymbol{h}}{dt} = {}^\circ\boldsymbol{M} \tag{2.15}$$

The angular momentum has the analytical expression

$${}^\circ\boldsymbol{h} = h_x\hat{\boldsymbol{I}} + h_y\hat{\boldsymbol{J}} + h_z\hat{\boldsymbol{K}} = {}^\circ\boldsymbol{I}{}^\circ\boldsymbol{\omega} \tag{2.16}$$

It is possible to express each parameter of the moment equation in a co-moving body reference frame with angular velocity Ω, using an appropriate rotation matrix ${}^{\circ}\boldsymbol{R}_B$

$$ {}^{\circ}\boldsymbol{I} = {}^{\circ}\boldsymbol{R}_B \boldsymbol{I}_B {}^{\circ}\boldsymbol{R}_B^T \tag{2.17} $$

Hence, in this frame we assume for simplicity that

- The moving xyz axis are the principal axis of inertia
- The moments of inertia relative to xyz are constant in time

so that the formulation for $\boldsymbol{\omega}_B$ becomes

$$ \boldsymbol{\omega}_B = {}^{\circ}\boldsymbol{R}_B {}^{\circ}\boldsymbol{\omega} \tag{2.18} $$

The angular momentum in the co-moving body reference frame is expressed as

$$ \boldsymbol{h}_B = h_x\hat{\boldsymbol{i}} + h_y\hat{\boldsymbol{j}} + h_z\hat{\boldsymbol{k}} \tag{2.19} $$

The time derivative of $\boldsymbol{h}_B$ is $\dot{\boldsymbol{h}}_B = \dot{\boldsymbol{h}}_{rel} + \boldsymbol{\Omega} \times \boldsymbol{h}_B$ so that 2.15 can be rewritten as

$$ \boldsymbol{M}_B = \dot{\boldsymbol{h}}_{rel} + \boldsymbol{\Omega} \times \boldsymbol{h}_B \tag{2.20} $$

If the angular velocity of the moving xyz coordinate system Ω and the angular velocity of the rigid body itself ω are both absolute kinematic quantities, equation 2.17 contains their components as projected onto the axis of the non-inertial xyz frame, as

$$ \boldsymbol{\Omega} = \Omega_x\hat{\boldsymbol{i}} + \Omega_y\hat{\boldsymbol{j}} + \Omega_z\hat{\boldsymbol{k}} \tag{2.21} $$

$$ \boldsymbol{\omega}_B = \omega_x\hat{\boldsymbol{i}} + \omega_y\hat{\boldsymbol{j}} + \omega_z\hat{\boldsymbol{k}} \tag{2.22} $$

The absolute angular acceleration $\boldsymbol{\alpha}$ is obtained as

$$ \boldsymbol{\alpha}_B = \dot{\boldsymbol{\omega}}_B = \boldsymbol{\alpha}_{rel} + \boldsymbol{\Omega} \times \boldsymbol{\omega}_B = \frac{d\omega_x}{dt}\hat{\boldsymbol{i}} + \frac{d\omega_y}{dt}\hat{\boldsymbol{j}} + \frac{d\omega_z}{dt}\hat{\boldsymbol{k}} + \boldsymbol{\Omega} \times \boldsymbol{\omega}_B \tag{2.23} $$

It is generally true that

$$ \alpha_x \neq \dot{\omega}_x \qquad \alpha_y \neq \dot{\omega}_y \qquad \alpha_z \neq \dot{\omega}_z \tag{2.24} $$

From the equation

$$ \boldsymbol{h}_B = J_x\omega_x\hat{\boldsymbol{i}} + J_y\omega_y\hat{\boldsymbol{j}} + J_z\omega_z\hat{\boldsymbol{k}} \tag{2.25} $$

where J_x, J_y and J_z are the moments of inertia, constant in time, about the principal axis of inertia, it is possible to write

$$ \dot{\boldsymbol{h}}_{rel} = J_x\dot{\omega}_x\hat{\boldsymbol{i}} + J_y\dot{\omega}_y\hat{\boldsymbol{j}} + J_z\dot{\omega}_z\hat{\boldsymbol{k}} \tag{2.26} $$

so that, substituting the equations 2.22 and 2.23 into equation 2.17 yields

$$\boldsymbol{M}_B = J_x\dot{\omega}_x\hat{\boldsymbol{i}} + J_y\dot{\omega}_y\hat{\boldsymbol{j}} + J_z\dot{\omega}_z\hat{\boldsymbol{k}} + \begin{vmatrix} \hat{\boldsymbol{i}} & \hat{\boldsymbol{j}} & \hat{\boldsymbol{k}} \\ \Omega_x & \Omega_y & \Omega_z \\ J_x\omega_x & J_y\omega_y & J_z\omega_z \end{vmatrix} \tag{2.27}$$

Expanding the cross product and collecting terms leads to

$$M_x = J_X\dot{\omega}_x + J_z\Omega_y\omega_z - J_y\Omega_z\omega_y \tag{2.28}$$
$$M_y = J_y\dot{\omega}_y + J_x\Omega_z\omega_x - J_z\Omega_x\omega_z \tag{2.29}$$
$$M_z = J_z\dot{\omega}_z + J_y\Omega_x\omega_y - J_x\Omega_y\omega_x \tag{2.30}$$

If the co-moving frame is rigidly attached to the body, then its angular velocity is the same as that of the body $\boldsymbol{\Omega} = \boldsymbol{\omega}_B$. In that case, it follows that

$$\alpha_x = \dot{\omega}_x \qquad \alpha_y = \dot{\omega}_y \qquad \alpha_z = \dot{\omega}_z \tag{2.31}$$

and the Euler's equation of motion can be rewritten as

$$\boldsymbol{M}_B = \dot{\boldsymbol{h}}_{rel} + \boldsymbol{\omega}_B \times \boldsymbol{h}_B \tag{2.32}$$

where

$$M_x = J_x\dot{\omega}_x + (J_z - J_y)\omega_y\omega_z \tag{2.33}$$
$$M_y = J_y\dot{\omega}_y + (J_x - J_z)\omega_z\omega_x \tag{2.34}$$
$$M_z = J_z\dot{\omega}_z + (J_y - J_x)\omega_x\omega_y \tag{2.35}$$

2.5 Equation of motion for a spacecraft controlled by a cluster of CMGs

A Control Moment Gyro is made of a spinning wheel mounted on a pivoting gimbal, with the axis perpendicular to the wheel spin axis. Instead of accelerating and decelerating the wheel in order to obtain the proper reaction from the spacecraft platform, the momentum exchange between the wheel and the bus is achieved rotating the wheel spin axis about the gimbal. The consequence is a gyroscopic torque obtained in a direction perpendicular to both the spin and the gimbal axis. At least three gimbals are necessary for full three-axis control, but a minimum of four CMGs is usually employed though a higher number of gimbals can be considered for increasing system redundancy and overall control power. In Fig. 2.4, referring to [1], it is shown an example of a cluster of four CMGs in a pyramid mounting.

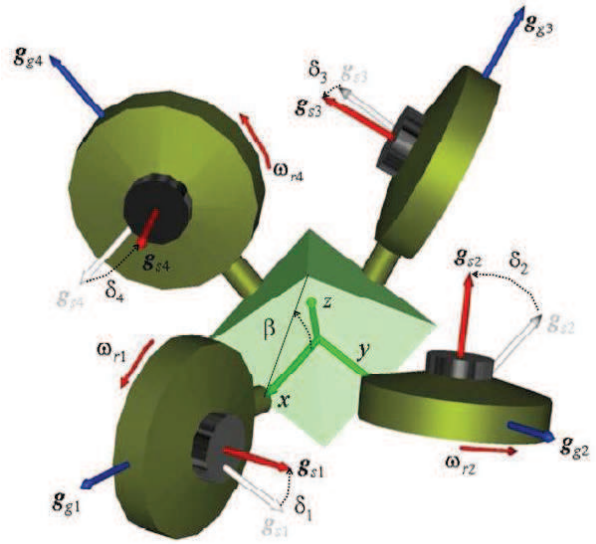


Figure 2.4: Example of a cluster of four CMGs in a pyramid mounting (Ref. [1])

Starting from the equation of motion 2.32 already written in the body fixed reference frame, it is now possible to introduce

$$\begin{aligned} \boldsymbol{M}_B &= \dot{\boldsymbol{h}}_B + \boldsymbol{\omega}_B \times \boldsymbol{h}_B \\ \boldsymbol{h}_B &= \boldsymbol{J}_B \boldsymbol{\omega}_B + \boldsymbol{h}_B^C \end{aligned} \tag{2.36}$$

where

- $\dot{\boldsymbol{h}}_B$ is the time derivative of the total angular momentum, spacecraft and CMGs (rotating parts)
- $\boldsymbol{J}_B \boldsymbol{\omega}_B$ refers only to the spacecraft
- $\boldsymbol{h}_B^C$ refers only to the CMG system

The equation of the CMGs angular momentum can be explicated as

$$\boldsymbol{h}_B^C = h_W \sum_{i=1}^{4} \bar{\boldsymbol{h}}_i(\theta_i) \tag{2.37}$$

where

- θ_i are the gimbal angles $i = 1 \div 4$

- $h_W = J_W \omega_W$ is the angular momentum of each wheel (with rotation at constant speed). h_W is constant in settings, the specific numerical value refers to the testbed used.

Note that the inertia tensor $\boldsymbol{J}_B$ is assumed to take into consideration the contribution of the spin-wheels to the mass distribution as if they were still. For this reason only the relative contribution to angular momentum is added. By replacing all the terms introduced, the equation of motion becomes

$$\boldsymbol{M}_B = \boldsymbol{\omega}_B \times (\boldsymbol{J}_B \boldsymbol{\omega}_B + \boldsymbol{h}_B^C) + \boldsymbol{J}_B \dot{\boldsymbol{\omega}}_B + \dot{\boldsymbol{h}}_B^C \tag{2.38}$$

Assuming for simplicity a torque-free environment and a zero initial angular momentum, it is possible to rewrite the equation 2.38 as

$$\boldsymbol{J}_B \dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times (\boldsymbol{J}_B \boldsymbol{\omega}_B) = \boldsymbol{u} \tag{2.39}$$

where

$$\boldsymbol{u} = -\dot{\boldsymbol{h}}_B^C - \boldsymbol{\omega}_B \times \boldsymbol{h}_B^C \tag{2.40}$$

is the control torque generated on the spacecraft bus by the action on the spinning gimbals. In this equation, $\dot{\boldsymbol{h}}_B^C$ is the internal control torque while $\boldsymbol{\omega}_B \times \boldsymbol{h}_B^C$ is the reaction torque of the spacecraft to all CMGs. The required control torque can be determined on the basis of the quaternion control that is analyzed in the following section. Once the required control torque $\boldsymbol{u}$ is known, the CMG steering logic can be determined as

$$\dot{\boldsymbol{h}}_B^C = -\boldsymbol{u} - \boldsymbol{\omega}_B \times \boldsymbol{h}_B^C \tag{2.41}$$

For a cluster of four CMGs, the internal angular momentum vector $\dot{\boldsymbol{h}}_B^C$ is a function of the rotation angles $\theta_1, \theta_2, \theta_3, \theta_4$ of each wheel about its gimbal axis, so that

$$\boldsymbol{h}_B^C = \boldsymbol{h}(\theta) \tag{2.42}$$

Taking the time derivative of this equation it is possible to get

$$\dot{\boldsymbol{h}}_B^C = h_W \boldsymbol{A}(\theta) \dot{\boldsymbol{\theta}} \tag{2.43}$$

Thus, we can convert the desired torque to a command for CMGs in order to have an attitude control by tracking $\dot{\boldsymbol{\theta}}$

$$\dot{\boldsymbol{\theta}} = \frac{1}{h_W} \boldsymbol{A}(\theta)^T (\boldsymbol{A}(\theta) \boldsymbol{A}(\theta)^T)^{-1} (-\boldsymbol{u} - \boldsymbol{\omega}_B \times \boldsymbol{h}_B^C) \tag{2.44}$$

Anyway, due to the fact that $\boldsymbol{A}(\theta)$ is not a square matrix, there could be the presence of a singularity that may not lead the problem to a convex solution. The case of a cluster of four CMGs mounted with the gimbal axis perpendicular to the faces of a

pyramid, the sides of which are inclined of a skew angle β, is the most utilized. In order to perform the reorientation, it is necessary to drive the gimbal according to Equation 2.42, where, considering $s\beta = \sin\beta$ and $c\beta = \cos\beta$, it is

$$\boldsymbol{A}(\theta) = \begin{bmatrix} -c\beta\cos\theta_1 & \sin\theta_2 & c\beta\cos\theta_3 & -\sin\theta_4 \\ -\sin\theta_1 & -c\beta\cos\theta_2 & \sin\theta_3 & c\beta\cos\theta_4 \\ s\beta\cos\theta_1 & s\beta\cos\theta_2 & s\beta\cos\theta_3 & s\beta\cos\theta_4 \end{bmatrix} \tag{2.45}$$

Thus, considering a torque-free environment as stated, the equation of motion can be finally rewritten as

$$\dot{\boldsymbol{\omega}}_B = -h_W \boldsymbol{J}_B^{-1} \boldsymbol{A}(\theta)\dot{\boldsymbol{\theta}} \tag{2.46}$$

2.6 The quaternion dynamics

In order to define the spacecraft orientation in terms of quaternion $q = (q_1, q_2, q_3, q_4)^T$, it is now possible to introduce the quaternion kinematic equation that can be written in two similar forms

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Sigma}(\omega_B)\boldsymbol{q} \qquad \text{or} \qquad \dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\eta}(q)\boldsymbol{\omega}_B \tag{2.47}$$

where $\Sigma(\omega_B)$ and $\eta(q)$ are respectively given by

$$\boldsymbol{\Sigma}(\omega_B) = \begin{bmatrix} 0 & \omega_{z_B} & -\omega_{y_B} & \omega_{x_B} \\ -\omega_{z_B} & 0 & \omega_{x_B} & \omega_{y_B} \\ \omega_{y_B} & -\omega_{x_B} & 0 & \omega_{z_B} \\ -\omega_{x_B} & -\omega_{y_B} & \omega_{z_B} & 0 \end{bmatrix}, \qquad \boldsymbol{\eta}(q) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{2.48}$$

In the algorithm described in this thesis, the formulation $\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Sigma}(\omega_B)\boldsymbol{q}$ is used. The spacecraft attitude obtained with quaternion kinematics can also be expressed in terms of Euler angles as follows

$$\phi = \arctan\left(\frac{2(q_4q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)}\right) \tag{2.49}$$

$$\theta = \arcsin\left(2(q_4q_2 - q_1q_3)\right) \tag{2.50}$$

$$\psi = \arctan\left(\frac{2(q_1q_2 + q_3q_4)}{1 - 2(q_2^2 + q_3^2)}\right) \tag{2.51}$$

Furthermore, we can define the state vector of the complete attitude dynamics as

$$\boldsymbol{x}_p = \begin{Bmatrix} \boldsymbol{q} \\ \\ \boldsymbol{\omega} \end{Bmatrix} \quad \in \mathbb{R}^{n_p} \tag{2.52}$$

In our analysis, $n_p = 7$.
It is then possible, considering equations 2.46 and 2.47, to write the system dynamics as

$$\dot{\boldsymbol{x}}_p = \begin{Bmatrix} \dot{\boldsymbol{q}} \\ \\ \dot{\boldsymbol{\omega}} \end{Bmatrix} = \begin{Bmatrix} \dot{\boldsymbol{\omega}}_B = -h_W \boldsymbol{J}_B^{-1} \boldsymbol{A}(\theta)\dot{\boldsymbol{\theta}} \\ \\ \dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Sigma}(\omega_B)\boldsymbol{q} \end{Bmatrix} \tag{2.53}$$

In the state-space formulation, considering $\boldsymbol{x}_p$ as the state vector, the system 2.53 can be rewritten as

$$\dot{\boldsymbol{x}}_p = \begin{bmatrix} 0_{4\times4} & \frac{1}{2}\boldsymbol{\Sigma}(\omega) \\ \\ 0_{3\times4} & 0_{3\times3} \end{bmatrix} \boldsymbol{x}_p + \begin{bmatrix} 0_{4\times4} \\ \\ -h_W \boldsymbol{J}^{-1}\boldsymbol{A}(\theta) \end{bmatrix} \dot{\boldsymbol{\theta}} \tag{2.54}$$

where $\dot{\boldsymbol{\theta}} \in \mathbb{R}^4$ is supposed to be the control input.

The system 2.54 is linearized, considering the desired attitude in terms of angular velocity $\boldsymbol{\omega}_d = [\omega_{x_d}, \omega_{y_d}, \omega_{z_d}]^T$ and $\boldsymbol{q}_d = [q_{1_d}, q_{2_d}, q_{3_d}, q_{4_d}]^T$ as equilibrium point and introducing $\boldsymbol{A}(\theta_0)$ matrix calculated in $\boldsymbol{\theta} = [0,0,0,0]^T = \boldsymbol{\theta}_0$ in the first step.

$$\dot{\boldsymbol{x}}_p = \begin{bmatrix} 0_{4\times4} & \frac{1}{2}\boldsymbol{\Sigma}(\omega_d) \\ \\ 0_{3\times4} & 0_{3\times3} \end{bmatrix} \boldsymbol{x}_p + \begin{bmatrix} 0_{4\times4} \\ \\ -h_W \boldsymbol{J}^{-1}\boldsymbol{A}(\theta_0) \end{bmatrix} \dot{\boldsymbol{\theta}} \tag{2.55}$$

$$\dot{\boldsymbol{x}}_p = \boldsymbol{J}_x \boldsymbol{x}_p + \boldsymbol{J}_u \dot{\boldsymbol{\theta}} \tag{2.56}$$

2.7 Dynamics implementation and simulation test

Using *Matlab* and *Simulink* it has been possible to implement the dynamics of the spacecraft analyzed in the previous sections. The dynamics of nonlinear equations has been fulfilled on *Simulink* (see B) while the inputs have been given on a *Matlab* script. In Fig. 2.5, 2.6, 2.7) the process is described step by step.

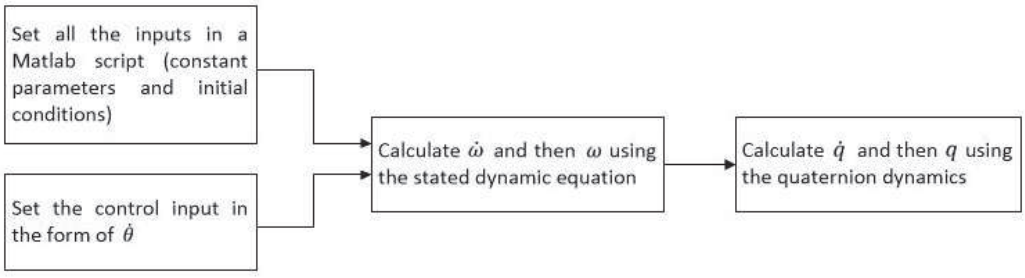


Figure 2.5: Nonlinear dynamics implementation flow

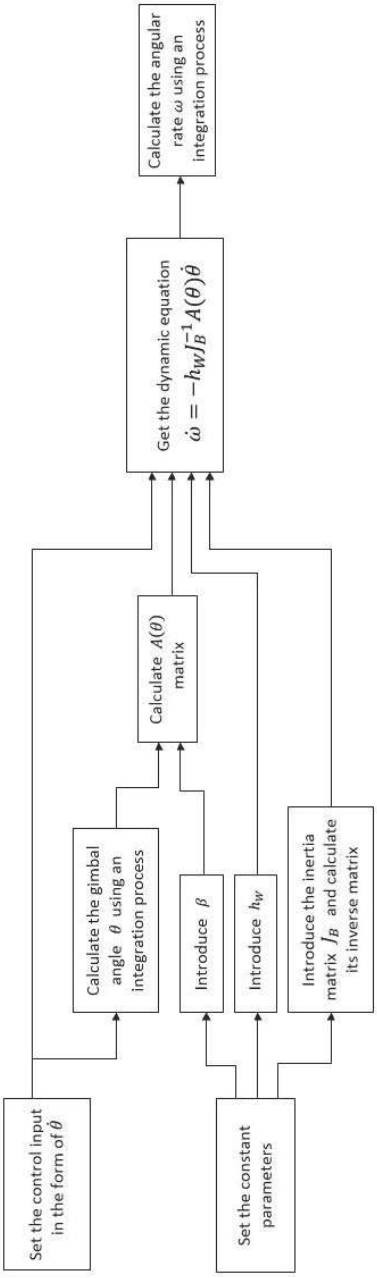


Figure 2.6: Particular of ω and θ dynamics implementation flow

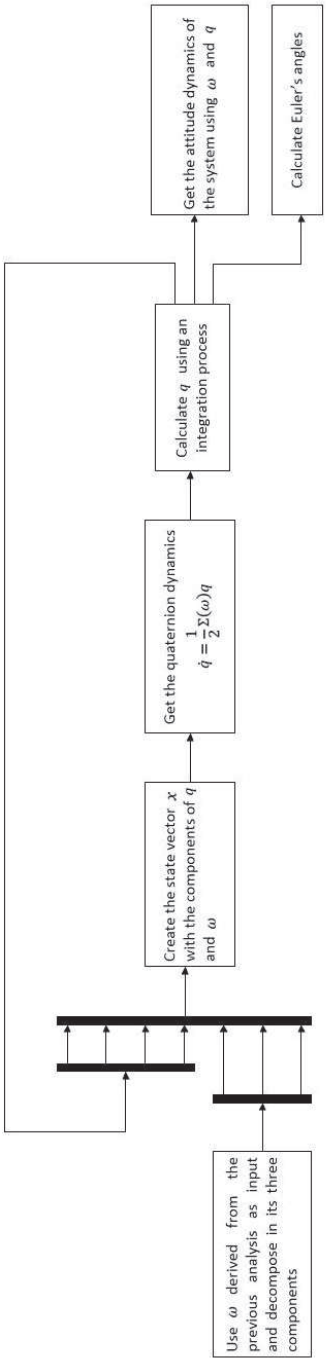


Figure 2.7: Particular of q dynamics implementation flow

Before introducing the controller to the dynamics, a test has been made in order to check the correct process between the *Simulink* blocks. The constant parameters

and the initial conditions given to the system are shown below.

$$J = \begin{bmatrix} 0.9684 & -0.0062 & -0.0087 \\ -0.0062 & 0.9768 & -0.0074 \\ -0.0087 & -0.0074 & 1.3000 \end{bmatrix}$$

$$\begin{aligned} \beta &= 45^\circ \\ h_W &= 0.0361 \\ \dot{\theta}_{max} &= 4 \ [rad/s] \\ q_0 &= [0, 0, 0, 1]^T \\ \omega_0 &= [0, 0, 0]^T \ [rad/s] \\ \dot{\omega}_0 &= [0, -0.3, -0.5]^T \ [rad/s^2] \\ \theta_0 &= [0, 0, 0, 0]^T \ [rad] \\ \dot{\theta}_0 &= [0, 0, 0, 0]^T \ [rad/s] \end{aligned}$$

Considering the results achieved with a given dynamics and considering the same control input, it is possible to find the following graphs comparing the two dynamics.

As we can see from the results, the dynamics implemented, under a given control input, is able to track the desired attitude shown in Fig.2.8 and Fig.2.10. In the following Chapter the first controller, Singular Direction Avoidance (SDA) Steering Law, is introduced to the already implemented dynamics and the results of both the simulation and the experiments on the testbed are presented.

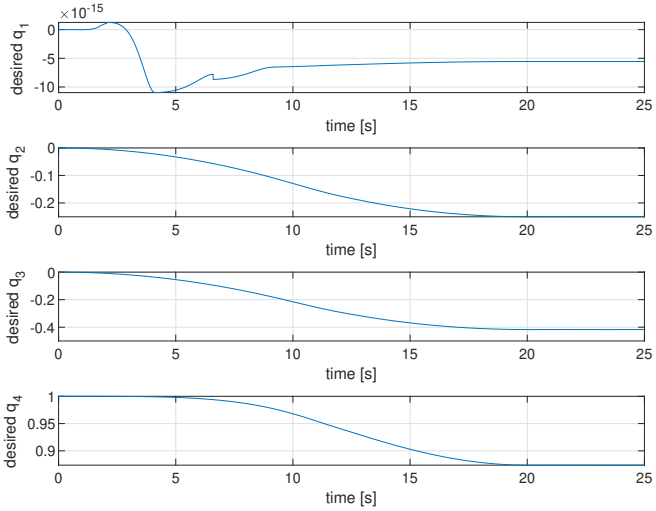


Figure 2.8: Desired quaternion attitude

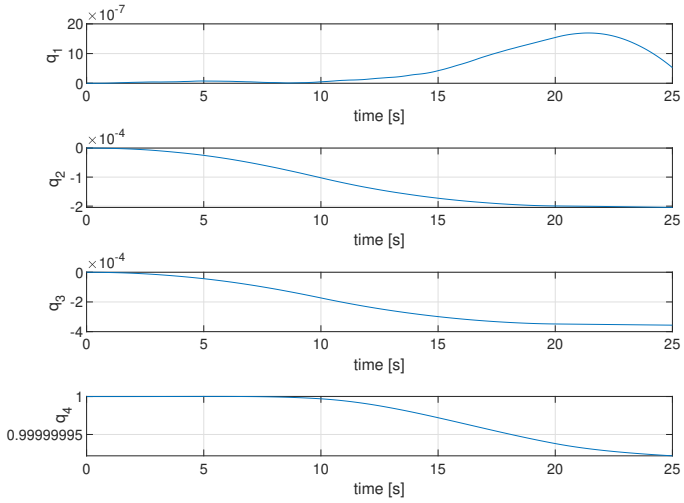


Figure 2.9: Dynamic test results: quaternion

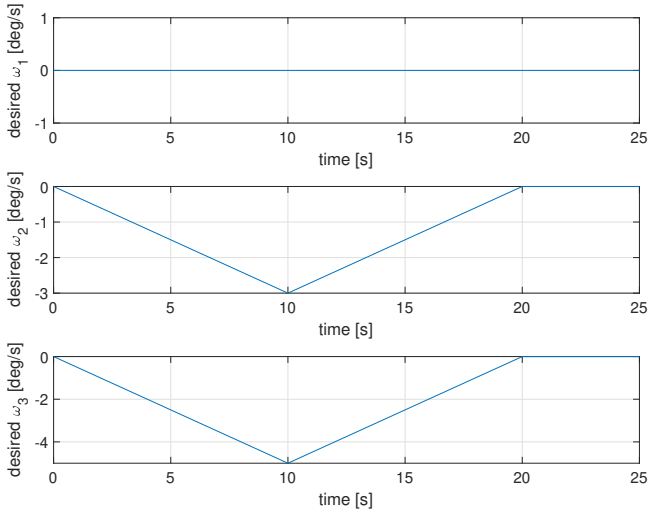


Figure 2.10: Desired ω

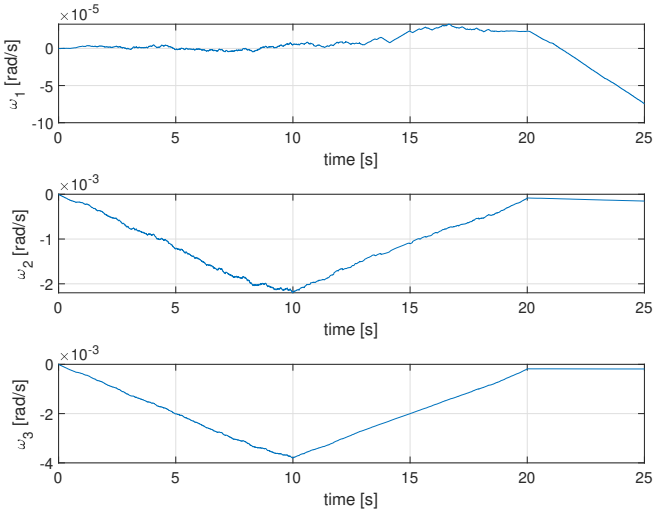


Figure 2.11: Dynamic test results: ω

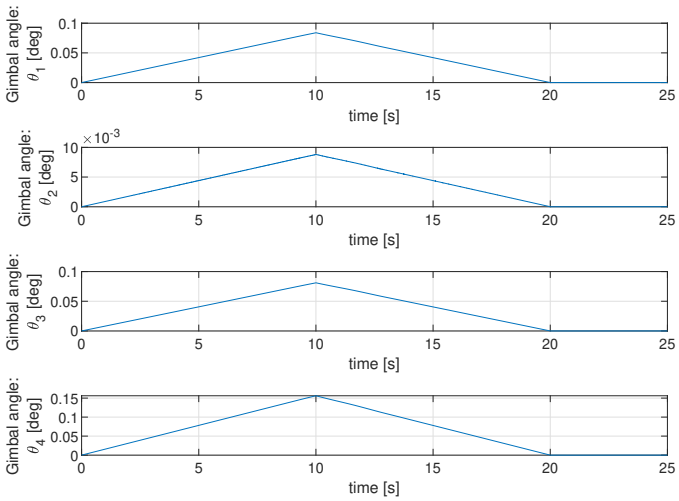


Figure 2.12: θ

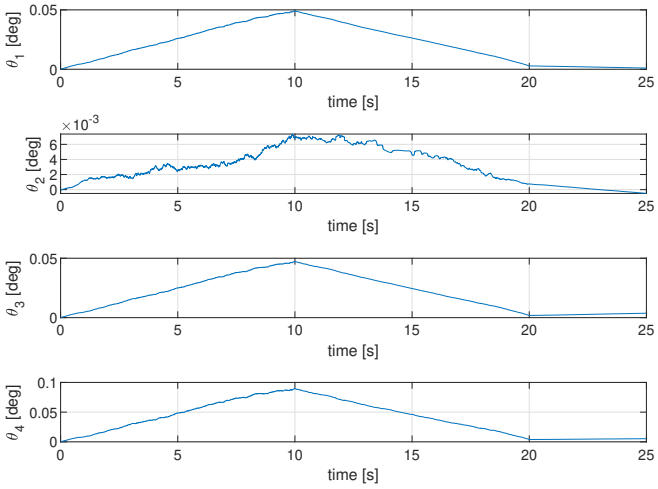


Figure 2.13: Dynamic test results: θ

Chapter 3

Singular Direction Avoidance (SDA) Steering Law

3.1 Basics on Control Moment Gyros singularities

Control Moment Gyros (CMGs) are spacecraft attitude control actuators that can be use for attitude hold and large spacecraft reorientation or for slew maneuvering. They provide the necessary torques by changing the direction of the angular momentum vector with respect to the spacecraft reference frame. One of the most important benefit is their torque amplification property as the flywheel of a CMG spins at a constant speed and the gimbal torque results in a gyroscopic torque (orthogonal to both the spin and gimbal axes) which is larger than the gimbal axis command torque. However, though torque amplification and other benefits like no gases expulsion and the use of electricity instead of fuel as power source, the employment of CMGs in practice has been hindered by the possibility of geometric singularities for certain combination of gimbal angles. Moreover, the complexity associated with the non-linearity of the attitude control problem, along with the terms introduced with the spinning flywheels, poses the largest obstacle to their use in spacecraft designs.

For definition, a singular state for a CMG is a gimbal angle combination at which no torque is possible along a certain direction. We can identify two types of singular states: *external* or *internal singularities*. *External* or *saturation singularities* occur when the sum of all CMG angular momenta reaches its maximum due to the fact that a CMG changes only the direction of the angular momentum vector and this leads to a maximal momentum surface. For this reason, For this reason, external singularities happens for those combinations in which the total CMG cluster momentum has reached this surface so that the it cannot generate torque directed

outward this surface. Furthermore, *internal singularities* exist for singular states at which the total angular momentum is smaller that the maximum. Description, analysis and classification of CMGsingularities can be found in [15], [16], [17] while [18], [19], [20], offer the visualization of the angular momentum surface.

While maneuvering, in order to be able to generate any commanded torque, the gimbal angles should be steered away from the singular states. There exists a variety of methods to avoid or escape these singularities and one of the most common one is based on the minimum two-norm, pseudo-inverse solution to the gimbal steering equation [17], [21], [22]. Though this method does not explicitly avoid the singularities, many steering logics are all based on it as it can be efficiently implemented online without the need of any offline calculation. Moreover, one of the features of this method is to steer the gimbal angles towards the singularities and rapidly transit through them with a finite torque error, which should be kept at a minimum value. The basic form of this method is known as Singularity Robust (SR) method and the variation implemented below is known as Singular Direction Avoidance (SDA) steering law, which provides the ability to avoid or escape any singularity with a particular finite gimbal rate. As a matter of fact, different choices of the singularity avoidance parameters produce different torque errors in the vicinity of the singularities.

There exist alternative methods for handling the CMGs singularities. They can be broadly categorized as local gradient methods and global avoidance methods. Even though these methods produce a torque which is exactly the same as the required torque, there still exist singularities which cannot be avoided using them. Indeed, the offline calculation needed for these methods makes them of limited use for online steering proceedings.

3.2 Moore-Penrose (MP) solution and Singular Direction Avoidance (SDA) Steering Law

The CMG steering equation we refer to during the analysis, following the same indications as in [21] and [23], is

$$B\ddot{\theta} + D\dot{\theta} = L_r \tag{3.1}$$

where both $B, D \in \mathbb{R}^{3\times N}$ matrices. However, B is a constant matrix while D depends on ω and θ. The complete formulation of matrices B, D and L_r can be found in C. Since the norm of the matrix B is quite smaller with respect to the norm of D, in order to take full advantage of the torque amplification property of CMGs,

the gimbal commands need to be given at the gimbal rate level. Note that from this section, it will not be used the bold notation to represent vectors or matrices anymore but the dimensions will always be fully determined.

Choosing the gimbal rates and the accelerations that satisfy 3.1 ensures the stability of the system, that is $\omega \rightarrow \omega_f$ and $q \rightarrow q_f$, where the constant terms ω_f and q_f are the desired final values for the reorientation maneuver. Thus, it is possible to neglect the term $B\ddot{\theta}$ and rewrite the steering equation as

$$D\dot{\theta} = L_r \tag{3.2}$$

with the objective to solve for $\dot{\theta}$, given any value of L_r. We can explicit the equation as

$$\dot{\theta} = D^{-1}L_r \tag{3.3}$$

As previously stated, matrix D in 3.2 has dimension $3 \times N$. Its maximal rank is 3 and whenever it has maximal rank the equation 3.2 can be solved as

$$\dot{\theta} = D^T(DD^T)^{-1}L_r \tag{3.4}$$

When there are more than three CMGs in the cluster, then the solution for $\dot{\theta}$ is underdetermined and a minimum two-norm solution can be calculated from this equation. This basic form for steering laws is called Moore-Penrose (MP) solution. Since the equation in 3.2 is underdetermined ($N \geq 3$), there are more than one solution. However, the MP solution is the one that minimizes

$$\min_{\dot{\theta}} \|\dot{\theta}\|^2 \quad \text{subject to} \quad D\dot{\theta} = L_r \tag{3.5}$$

In particular, the CMG singular states are defined as those for which the rank of the matrix D is less than 3 in 3.2. At these singular states, the matrix DD^T is not invertible and the steering law in 3.4 fails to produce the required torque. Indeed, the linear system has a solution if and only if the vector L_r is in the range space of the matrix D, which is always the case of rank $D = 3$. However, if rank $D < 3$ there exist torque vector directions that cannot be met and no set of gimbal commands $\dot{\theta}$ can produce torque along this direction. Moreover, the magnitude of the gimbal rate $\dot{\theta}$ becomes excessive in the vicinity of the singular states, violating the gimbal rate constraints.

The fact that MP steering law fails whenever the rank of D matrix is less than 3 can be shown in the solution 3.4, which drives the gimbal angles towards singular configurations. For this reason, this law is of limited use on a real system unless some other steering logics avoid or escape the singularities. Therefore, if the required torque is orthogonal to the singular direction, there is no reason to avoid

the singularity during the reorientation profile. Moreover, using the Singular Value Decomposition (SVD) it is possible to steer through some singularities, resulting in a much smoother reorientation.

The Singularity Robust (SR) steering law is a modified version of Moore-Penrose solution, originally developed in [24]. In the SR theory, in order to keep D matrix well-conditioned and invertible, an extra singularity avoidance parameter is added to the pseudo-inverse of D when the system is close to the singularity. This addition ensures finite gimbal rates in the vicinity of the singularity. The SR steering law is given by

$$\dot{\theta} = D^T(DD^T + \alpha I_3)^{-1} L_r \tag{3.6}$$

This equation solves the following minimization problem

$$\min_{\dot{\theta}} \{\frac{1}{2}\alpha\|\dot{\theta}\|^2 + \frac{1}{2}\|D\dot{\theta} - L_r\|^2\} \tag{3.7}$$

In the equation in 3.6, the parameter α is chosen to be small or zero away from the singular states and it takes a nonzero value at the singular states. Indeed, it is negligible when DD^T is nonsingular but increases as a singularity is approached. It is commonly used for α value

$$\alpha = \alpha_0 e^{-\det(DD^T)}$$

where α_0 is a small constant. Near or at a singularity, we have to accept some deviation, as it may be possible to produce the desired torque with finite gimbal rates. Moreover, this steering logic shows gimbal lock in cases when the requested torque direction is parallel to the singular direction, so that $D^T L_r = 0$. Thus, the gimbal is locked at the same position and the commanded gimbal rate is zero.

Starting from the observation that the singularity of the matrix is determined by its smallest singular value, a different approach has been proposed to deal with the singularity problem. In this second analysis, only the value close to the singularity is modified, instead of all the three values in equation 3.6. This method is called Singular Direction Avoidance (SDA) steering law and has the benefit of ensuring a smaller torque error than the SR method.

It is generally possible to decompose any $n \times m$ matrix into the product of three matrices using the Singular Value Decomposition as

$$D = USV^T \tag{3.8}$$

where U is a unitary matrix of dimension $\mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ unitary matrix and S is a diagonal matrix of dimension $\mathbb{R}^{m\times n}$. In particular, U and V vectors and S matrix

can be explicated as

$$U = [\bar{u}_1, \bar{u}_2, \bar{u}_3]$$
$$V = [\bar{v}_1, \bar{v}_2, \bar{v}_3, \bar{v}_4]$$
$$S = \begin{bmatrix} \sum_{r\times r} & 0_{r\times(4-r)} \\ 0_{(3-r)\times r} & 0_{(3-r)\times(4-r)} \end{bmatrix}$$

where $\sum_{r\times r} = \text{diag}[\sigma_1, \ldots, \sigma_r]$. As we can see, if S has more columns than rows, so that $m < n$, the last $n - m$ columns are all zeros.

It is then possible to compute D^{-1}, in the case where D is a square matrix, as

$$D^{-1} = VS^{-1}U^T \tag{3.9}$$

When $m < n$, the matrix $D^T(DD^T)^{-1}$ can be calculated form the elements of the SVD by discarding the last $m-n$ zero columns of S and V, inverting the new $m \times m$ square matrix S and forming the following

$$D^{\dagger} = V_t S_t^{-1} U^T \tag{3.10}$$

where V_t and S_t are the truncated matrices. This equation gives the minimum norm solution to the undetermined case and can be easily used for the CMG problem when rank $(D) = 3$. However, whenever the gimbals become singular, S_t^{-1} does not exist because a singular value goes to zero.

Therefore, the SDA steering logic is given by

$$\dot{\theta} = VS_{SDA}^{\ddagger} U^T L_r \tag{3.11}$$

where

$$S_{SDA}^{\ddagger} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 \\ 0 & \frac{1}{\sigma_2} & 0 \\ 0 & 0 & \frac{\sigma_3}{\sigma_3^2+\alpha} \\ 0 & 0 & 0 \end{bmatrix} \tag{3.12}$$

In this case, the singularity avoidance parameter α is chosen such that

$$\alpha = \alpha_0 e^{-k_\sigma \bar{\sigma}_3^2}$$

where $\bar{\sigma}_3 = \sqrt{(N/3)}(\sigma_3/h_W)$ is a non-dimensional variable normalized with respect to the magnitude of the flywheel momentum h_W, so that the response of the system is independent from the system size. Moreover, α and k_σ are arbitrary constants.

Equation 3.11 can be explicated in the same form as 3.6, so that

$$\dot{\theta} = -[\bar{v}_1\ \bar{v}_2\ \bar{v}_3] \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 \\ 0 & \frac{1}{\sigma_2} & 0 \\ 0 & 0 & \frac{\sigma_3}{\sigma_3^2+\alpha} \end{bmatrix} \begin{bmatrix} \bar{u}_1^T \\ \bar{u}_2^T \\ \bar{u}_3^T \end{bmatrix} L_r$$

$$\dot{\theta} = -A^T(AA^T + \lambda \bar{u}_3 \bar{u}_3^T)^{-1}\tau_c$$

where $A = A(\theta)$ derives from the dynamics in 2.45, λ is the SDA steering law parameter and $\bar{u}_3$ is the unitary matrix. The two forms are equivalent and τ_c is defined using the errors between the desired attitude and the one calculated by the system, with the appropriate control gains as follows

$$\tau_c = -k_d(q - q_d) - k_p(\omega - \omega_d) \tag{3.13}$$

3.3 Implementation and simulation results

The Proportional Derivative (PD) controller with SDA Steering Law has been fully implemented on *Matlab*, using the dynamics already fulfilled in the dynamic simulation test.

The dynamic constant parameters used in the simulations are

$$\beta = 45^\circ$$
$$h_W = 0.0361$$
$$\dot{\theta}_{max} = 4\ \ [rad/s]$$
$$J = \begin{bmatrix} 0.9684 & -0.0062 & -0.0087 \\ -0.0062 & 0.9768 & -0.0074 \\ -0.0087 & -0.0074 & 1.3000 \end{bmatrix}$$

The PD gains and SDA parameter are, respectively

$$k_d = 12.5$$
$$k_p = 5$$
$$\lambda = 0.01$$

The initial conditions for the desired attitude are as follows

$$q_{d0} = [0, 0, 0, 1]^T$$
$$\omega_{d0} = [0, 0, 0]^T\ \ [rad/s]$$
$$\dot{\omega}_{d0} = [0, -0.3, -0.5]^T\ \ [rad/s^2]$$

while the initial conditions for the current attitude, gimbal angles and gimbal rate are

$$
\begin{aligned}
q_0 &= [0, 0, 0, 1]^T \\
\omega_0 &= [0, 0, 0]^T \ \ [rad/s] \\
\theta_0 &= [0, 0, 0, 0]^T \ \ [rad] \\
\dot{\theta}_0 &= [0, 0, 0, 0]^T \ \ [rad/s]
\end{aligned}
$$

Lastly, the simulation is made considering

$$
\begin{aligned}
t_0 &= 0 \ \ [s] \\
t_f &= 25 \ \ [s] \\
dt &= 10^{-3} \ \ [s]
\end{aligned}
$$

The results are shown below.

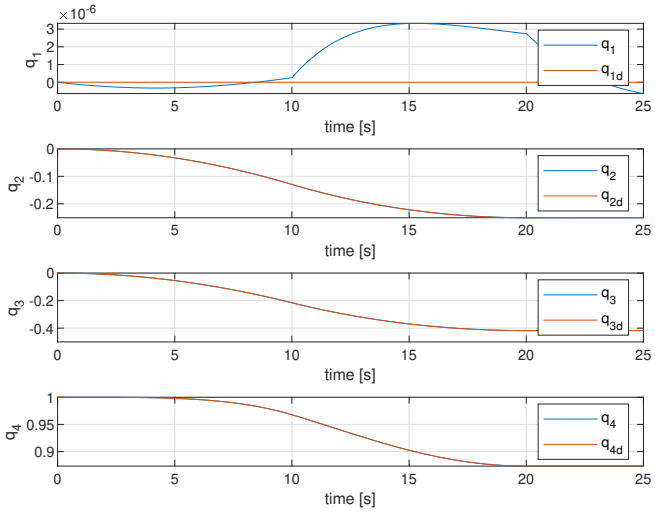


Figure 3.1: Quaternion attitude controlled by PD-SDA steering law

The desired attitude is represented by the red line in 3.1 and 3.2. It is in terms of quaternions and angular velocity and it changes over the time. Though the desired attitude is time-varying, the graphs show that the PD controller is able to track it.

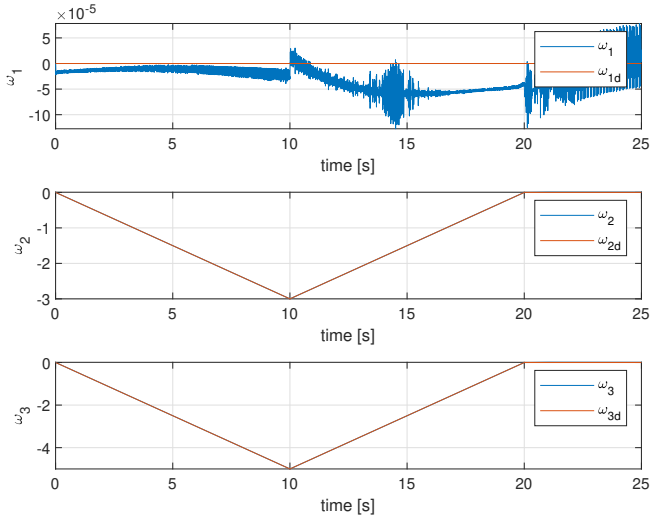


Figure 3.2: Angular velocity ω controlled by PD-SDA steering law

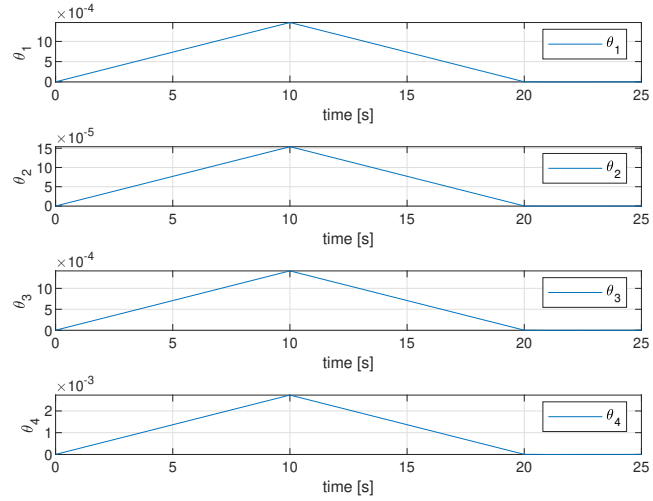


Figure 3.3: Gimbal angles θ controlled by PD-SDA steering law

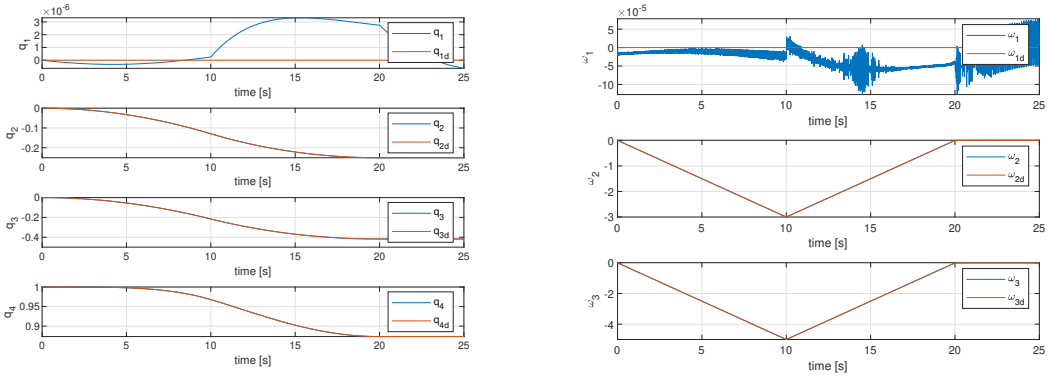
3.4 Simulation results with parameter variation

In this section, an analysis of the simulation results with parameter variation is considered. Each parameter is chosen to vary keeping all the others at the nominal value. The variations of the parameters are considered in the following order:

- Variation of the inertia matrix J_B
- Variation of the CMG skew angle β
- Variation of the PD-SDA gains k_d, k_p
- Variation of the SDA parameter λ

3.4.1 Variation of the inertia matrix J_B

The analysis focuses on the variation of the inertia matrix J_B, considering three different matrices which represent three system dynamics. The results achieved for the three J_B matrices will then be compared to the nominal results.



(a) *Quaternion dynamics* (b) *Angular velocity* ω

Figure 3.4: Simulation results in terms of q and ω considering the nominal values of J_B, β, k_d, k_p and λ

Considering J_{B_n} the nominal J_B matrix written in the previous section, the three

different matrices considered are:

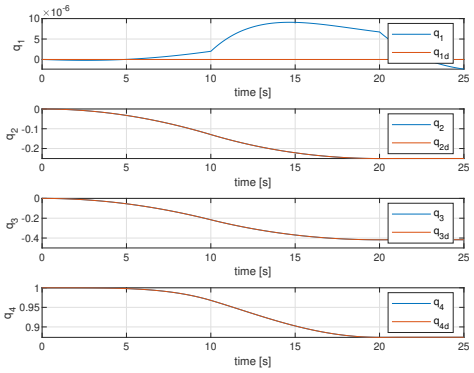
$$J_{B_1} = \begin{bmatrix} 0.5684 & -0.0124 & -0.0042 \\ -0.0124 & 0.5768 & -0.0140 \\ -0.0042 & -0.0140 & 1.5000 \end{bmatrix}$$

$$J_{B_2} = \begin{bmatrix} 0.8684 & -0.0062 & -0.0160 \\ -0.0062 & 0.1768 & -0.0004 \\ -0.0160 & -0.0004 & 1.8000 \end{bmatrix}$$

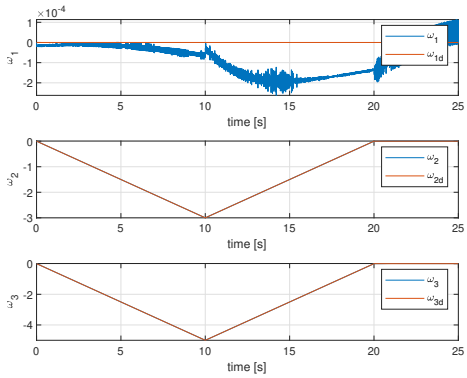
$$J_{B_3} = \begin{bmatrix} 1.0000 & -0.0050 & -0.0100 \\ -0.0050 & 1.0000 & -0.0150 \\ -0.0100 & -0.0150 & 1.0000 \end{bmatrix}$$

The results, in term of quaternions and angular velocity, are shown below.

As we can see in 3.4, 3.5 and 3.6, the tracking performance of the three different dynamics are as worse as the value of J_B gets away from the nominal one. The desired attitude is still tracked but a variation of other dynamic parameters is needed to keep the controller effective and with a small error.

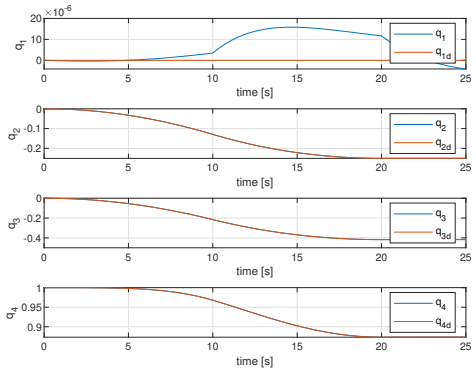


(a) *Quaternion dynamics with* J_{B_1}

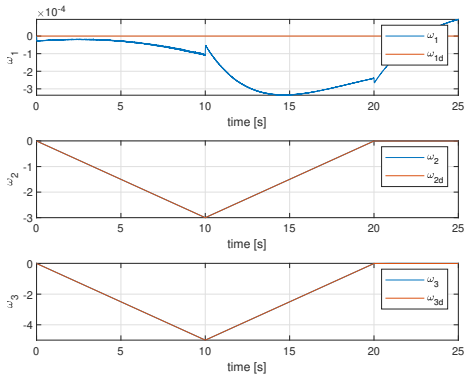


(b) *Angular velocity* ω *with* J_{B_1}

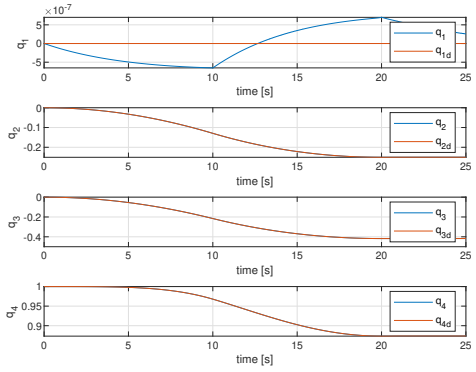
Figure 3.5: Simulation results in terms of q and ω considering J_{B_1} matrix



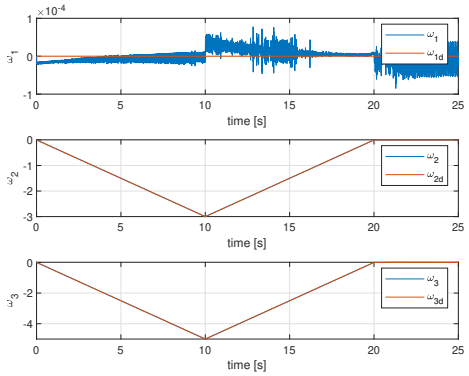
(a) *Quaternion dynamics with* J_{B_2}



(b) *Angular velocity* ω *with* J_{B_2}



(c) *Quaternion dynamics with* J_{B_3}



(d) *Angular velocity* ω *with* J_{B_3}

Figure 3.6: Simulation results in terms of q and ω considering J_{B_2} and J_{B_3} matrices

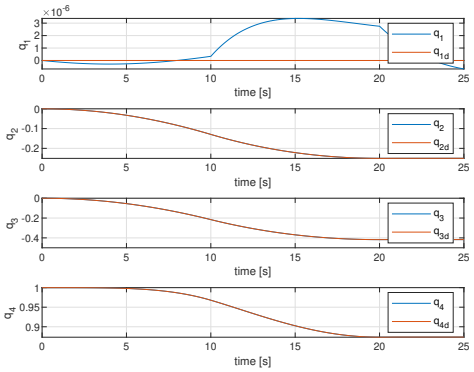
3.4.2 Variation of the CMG skew angle β

The second analysis focuses on the variation of the Control Moment Gyro skew angle β, which is a variation also related to the dynamics of the system, as for the matrix J_B.

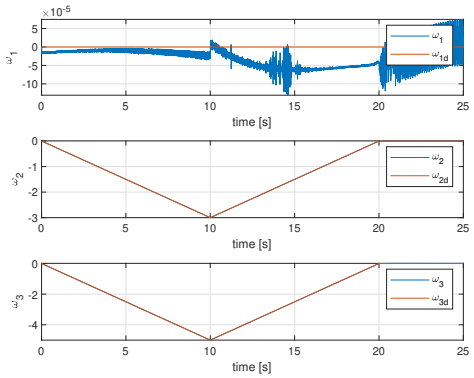
Considering $\beta_n = 45°$ the nominal value of β written in the previous section, the three different skew angles considered are:

$$\beta_1 = 40°$$
$$\beta_2 = 44°$$
$$\beta_3 = 50°$$

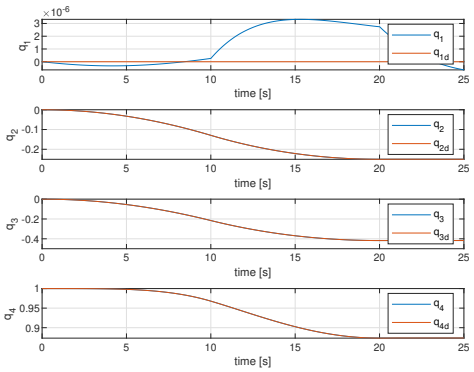
The results are shown in the next page.



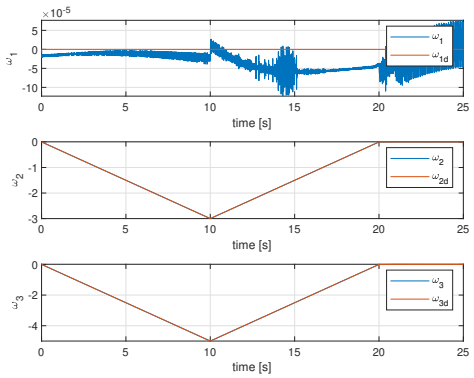
(a) *Quaternion dynamics with* β_1



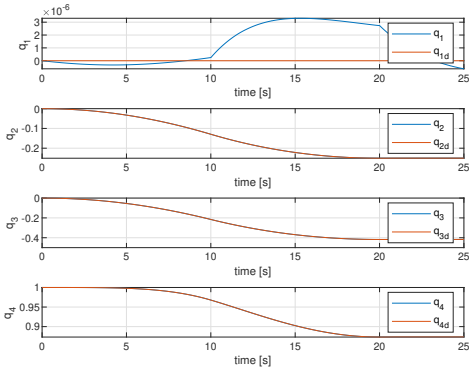
(b) *Angular velocity* ω *with* β_1



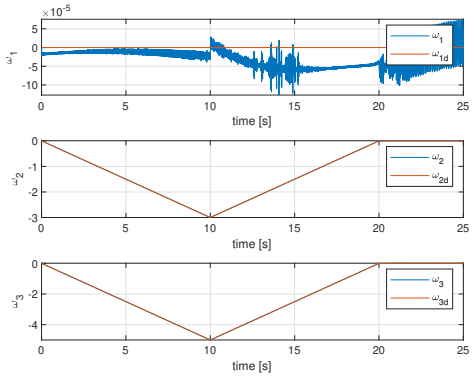
(c) *Quaternion dynamics with* β_2



(d) *Angular velocity* ω *with* β_2



(e) *Quaternion dynamics with* β_3



(f) *Angular velocity* ω *with* β_3

Figure 3.7: Simulation results in terms of q and ω considering three different values of the skew angle β

From Fig. 3.4 and 3.7 it is possible to notice that the results achieved with $\beta_2 = 44°$ are similar to the nominal ones. Moreover, the results related to both β_1 and β_3 leads to a slightly different trend of the tracking performance. Nevertheless, it is still possible to ensure a good accuracy to the attitude of the system.

3.4.3 Variation of the PD-SDA gains k_d, k_p

The third analysis focuses on the variation of the PD-SDA gains k_d and k_p, used in the PD formulation as multiplicative factors of the errors between the desired attitude and the current one, as in 3.13.

Considering k_{d_n} and k_{p_n} the nominal values of k_d and k_p, respectively, previously set as

$$k_{d_n} = 12.5$$
$$k_{p_n} = 5$$

the three different couples of gains chosen for the analysis are:

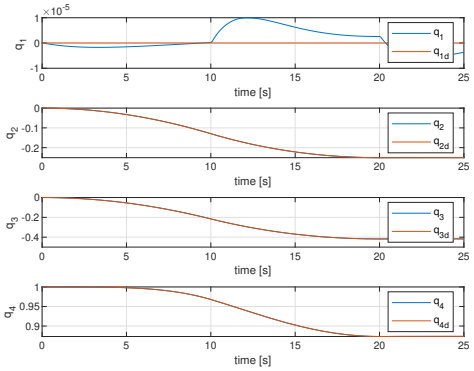
$$k_{d_1} = 1.5$$
$$k_{p_1} = 1.5$$

$$k_{d_2} = 10$$
$$k_{p_2} = 3$$

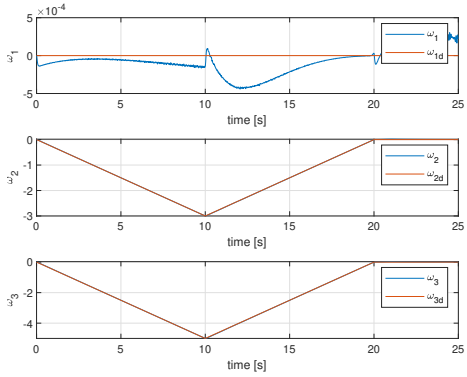
$$k_{d_3} = 25$$
$$k_{p_3} = 15$$

The results are shown in the next page.

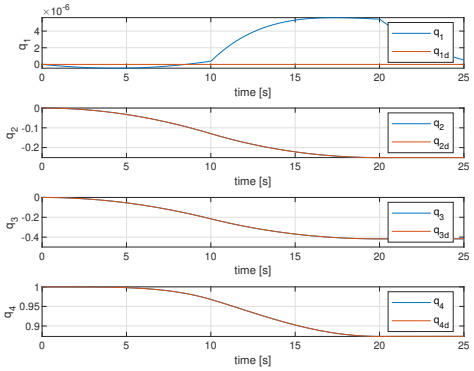
Fig. 3.4 and 3.8 show the importance of choosing the appropriate PD gains in order to achieve a good accuracy of the tracking performance for the PD controller. The first and third couples of gains, totally different from the nominal one, leads to a wider fluctuation of the solution, while the second couple of gains, which is similar to the nominal one, has a more accurate trend. Nevertheless, in all the three cases the desired attitude is tracked.



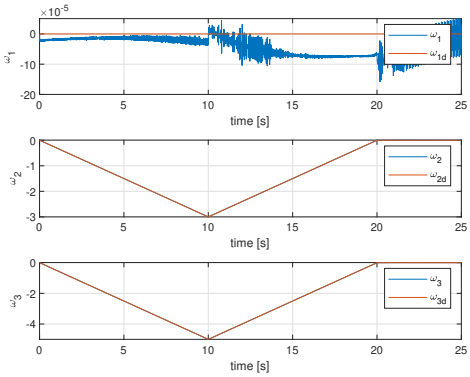
(a) *Quaternion dynamics with* k_{d_1}, k_{p_1}



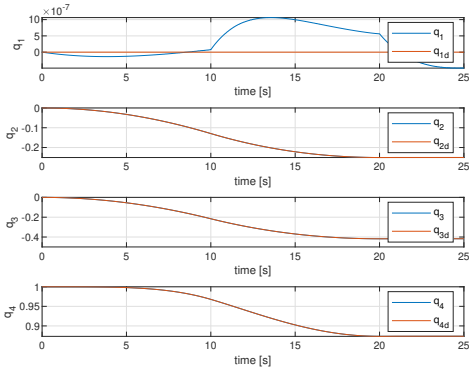
(b) *Angular velocity* ω *with* k_{d_1}, k_{p_1}



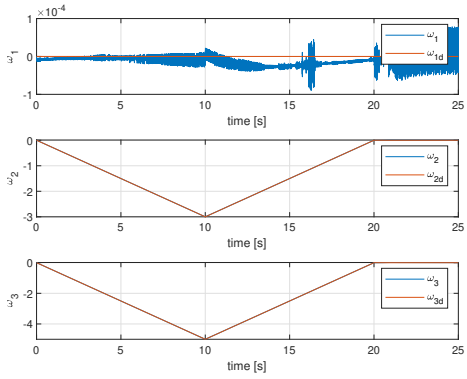
(c) *Quaternion dynamics with* k_{d_2}, k_{p_2}



(d) *Angular velocity* ω *with* k_{d_2}, k_{p_2}



(e) *Quaternion dynamics with* k_{d_3}, k_{p_3}



(f) *Angular velocity* ω *with* k_{d_3}, k_{p_3}

Figure 3.8: Simulation results in terms of q and ω considering three different couples of SDA gains

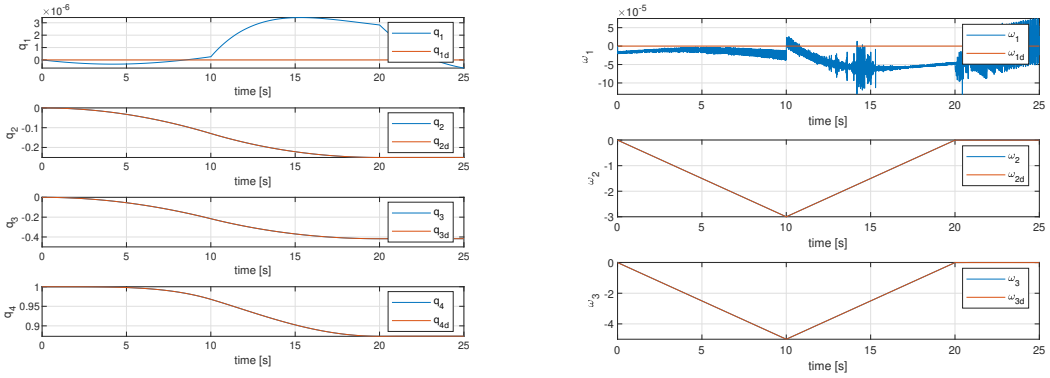
3.4.4 Variation of the SDA parameter λ

The fourth and last analysis focuses on the variation of the SDA parameter λ, which is used to properly calibrate the steering law in order to avoid the CMG singularity.

Considering $\lambda_n = 0.01$ the nominal value of λ written in the previous section, the three different SDA parameters considered are:

$$\lambda_1 = 0.0001$$
$$\lambda_2 = 10$$
$$\lambda_3 = 100$$

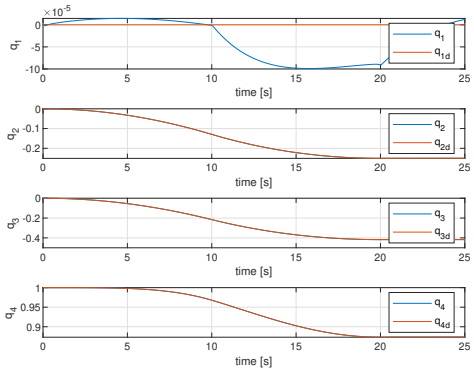
The results are shown below.



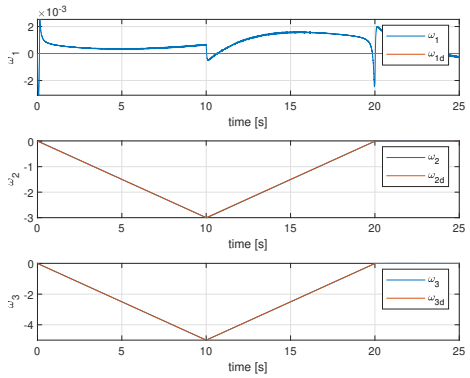
(a) *Quaternion dynamics with* λ_1 (b) *Angular velocity* ω *with* λ_1

Figure 3.9: Simulation results in terms of q and ω considering λ_1

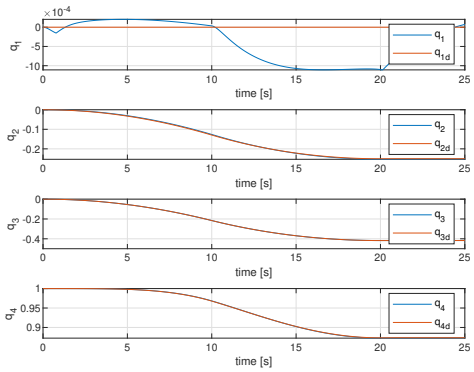
As we can see from Fig. 3.4, 3.9 and 3.10, the variation of λ leads to a different trend of the first component of both q and ω vectors. Moreover, the more SDA steering law parameter λ increases its value, the more the tracking performance has a degradation of accuracy. This is clearly shown in 3.10 (d), where also the second component of the angular velocity ω has a shift compared to the desired value so that the results are not completely effective.



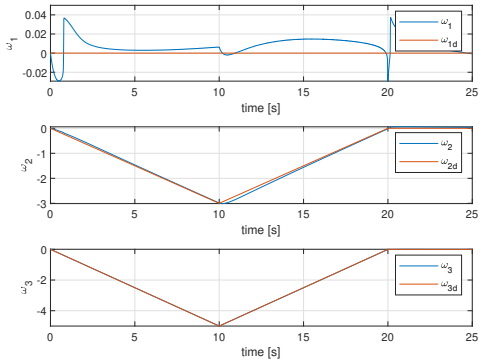
(a) *Quaternion dynamics with* λ_2



(b) *Angular velocity* ω *with* λ_2



(c) *Quaternion dynamics with* λ_3



(d) *Angular velocity* ω *with lambda*$_3$

Figure 3.10: Simulation results in terms of q and ω considering λ_2 and λ_3

3.5 Testbed description and experimental results

The last section of the PD analysis is about the experimental results. The experiments are made on a testbed located at the *Yamada Laboratory, Graduate School of Engineering, Osaka University, Osaka, Japan.*

3.5.1 Experimental setup description

As shown in 3.11, the experimental setup simulates a spacecraft using an air floating table. The equipment consists in a compressor, a table and a support column. The table floats slightly due to the compressed air and it can rotate freely around three axes with almost no friction. However, the gravity torque arises when the center of rotation and the center of gravity of the experimental setup are misaligned. Thus, in order to minimize this gravity torque, the position of the three counter weights, which are attached to the experimental platform by ball screws, is manually adjusted and the period of the 3D pendulum is empirically maximized, so that the center of gravity and the center of rotation coincide as much as possible.

Figure 3.12 shows four CMGs in pyramid configuration arranged on the device table. In addition, a 9-axes (9 DOF) sensor, two batteries, a PC and a microcomputer are installed, in order to acquire the correct attitude. The PC controls the entire system, reads sensor information, sends instructions to the microcomputer and makes calculations according to the steering law. The microcomputer rotates at a constant speed while reading the CMG wheels information. Thus, angular speed information is sent to the main PC as needed. Moreover, the device is controlled and set up with wireless HDMI, wireless keyboard and mouse, and the experimental device is completely stand-alone.

Therefore, 3.13 shows a close up of one CMG which has a DC motor for the wheel rotation and a wheel angular speed control. It consists in a photo sensor for the system control and a servo motor for the gimbal rotation. In addition, 3.14 shows a simple diagram of the experimental operations.

In tables 3.1, 3.2 and 3.3, it is possible to find information on the CMG performance, 9 DOF sensor performance and computer performance, respectively. For specific product names of experimental tools, refer to D.

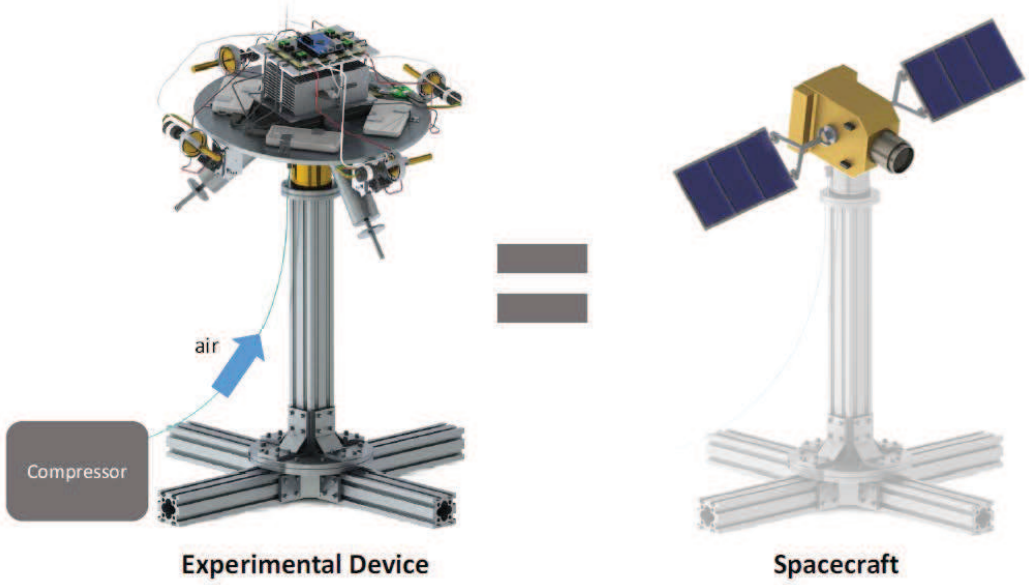


Figure 3.11: Experimental setup

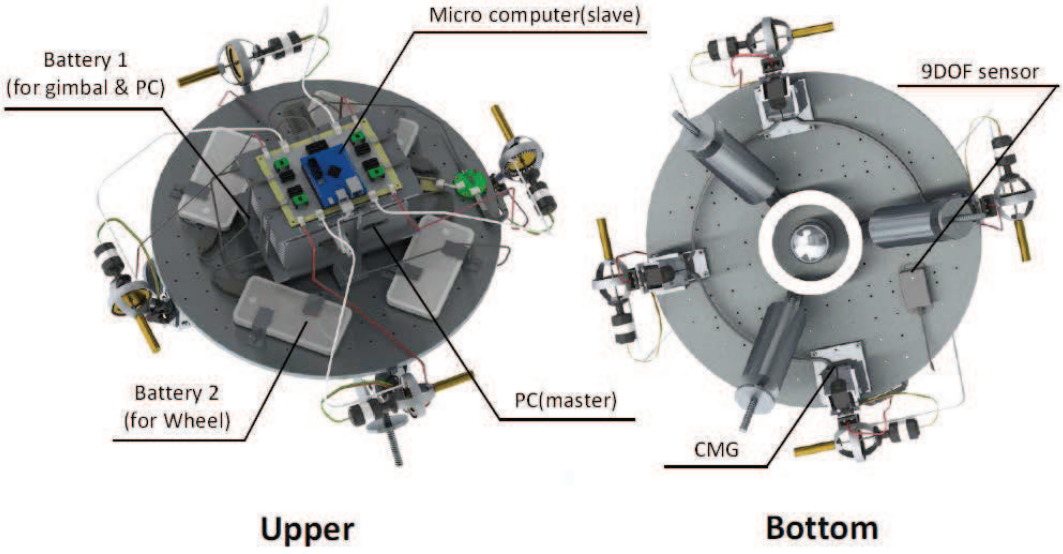


Figure 3.12: Tools arranged on the device table

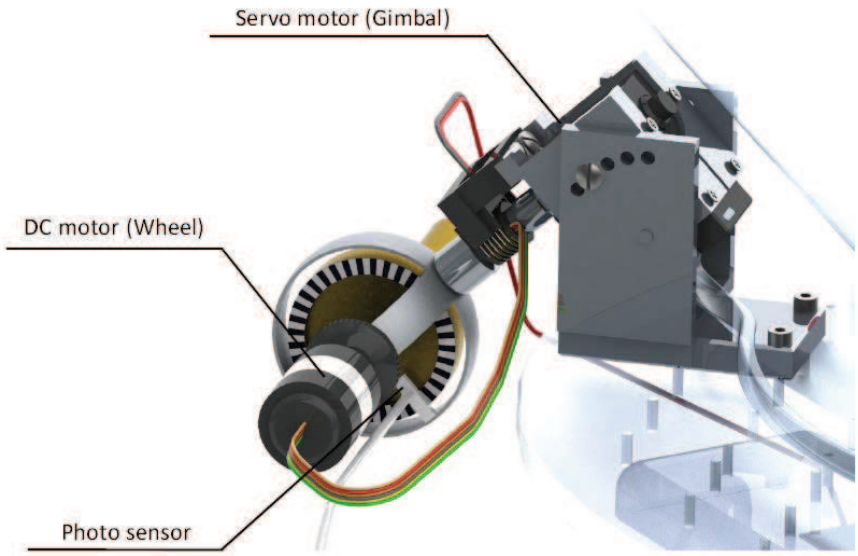


Figure 3.13: Control Moment Gyro details

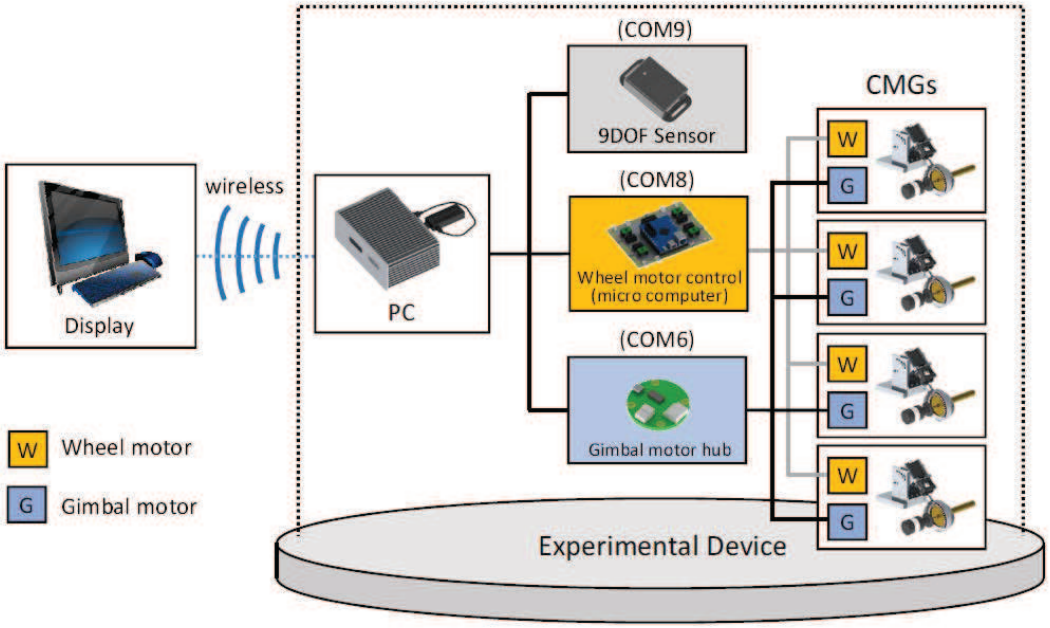


Figure 3.14: Operation diagram

Table 3.1: CMG performance

Wheels		
Nominal angular speed	[rpm]	10000
Angular speed error	[rpm]	±107
Maximum angular speed	[rpm]	14500
Moment of inertia	[kgm^2]	3.44×10^{-5}
Angular momentum	[Nm]	0.0361
Gimbals		
Maximum torque	[Nm]	4.1
Maximum angular velocity	[rad/s]	4.76
Angular error	[deg]	0.088
Operating angle range	[deg]	±320

Table 3.2: 9 DOF sensor performance

Acceleration sensor		
Resolution	[bit]	14
Noise density	$[\mu g/\sqrt{Hz}]$	650
Sensitivity	[g/digit]	0.003-0.012
Gyro sensor		
Resolution	[bit]	16
Noise density	$[\mathrm{deg/s}/\sqrt{Hz}]$	0.009
Sensitivity	[deg/s/digit]	0.00833
Magnetic sensor		
Resolution	[bit]	12
Sensitivity	[mGa/digit]	0.73

Table 3.3: Computer performance

PC		
CPU	-	Core i5-4250U
Clock frequency	[GHz]	1.3
Memory	[GB]	4
Microcomputer		
CPU	-	RX CPU(RX62N)
Clock frequency	[MHz]	100
Memory	[KB]	512

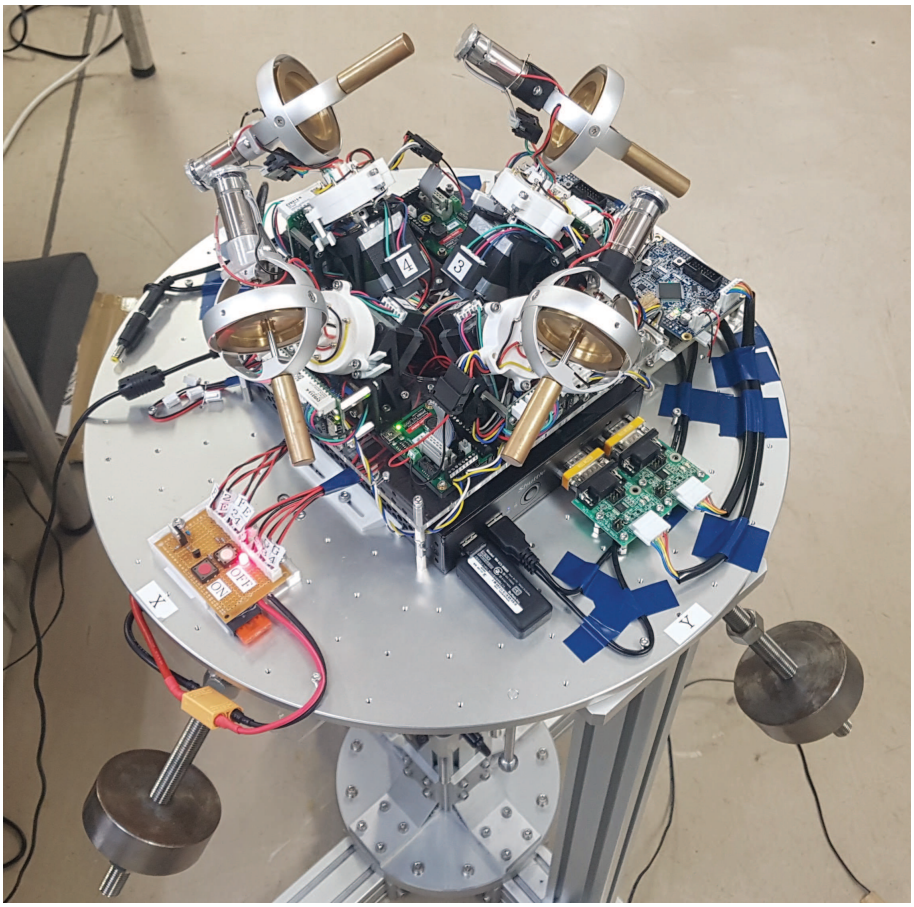


Figure 3.15: Experimental testbed in the Yamada Laboratory, Osaka University

3.5.2 Procedure for using the experimental setup

The experimental setup must be used correctly according to the following procedure.

Before the experiment
Press the two power buttons in 3.16 on the center of the table, one for the PC and one for the gimbal motor batteries.
Turn on the PC using wireless HDMI and wireless keyboard. As shown in Figure 3.17, turn on all the power supplies for the gimbal motors, the wheel motors and the wheel control microcomputer.
Check that the wheel motor power brush is not touching so that the wheel motor can rotate properly.
Turn on the compressor and wait until there is no vibration noise. When the vibration noise stops, remove the column lid and place the table on the column with great care.

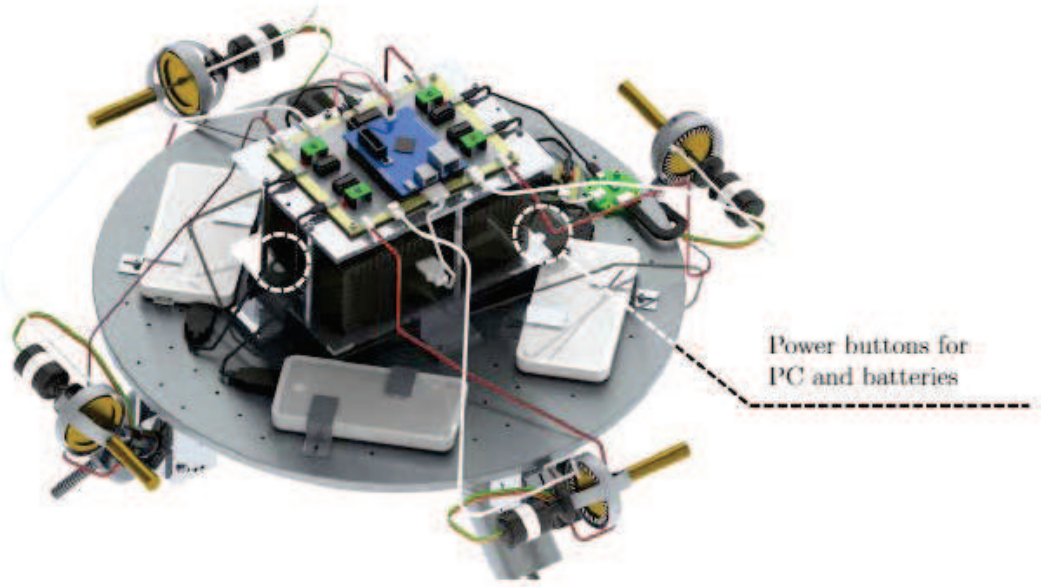


Figure 3.16: First step to switch on the table tools

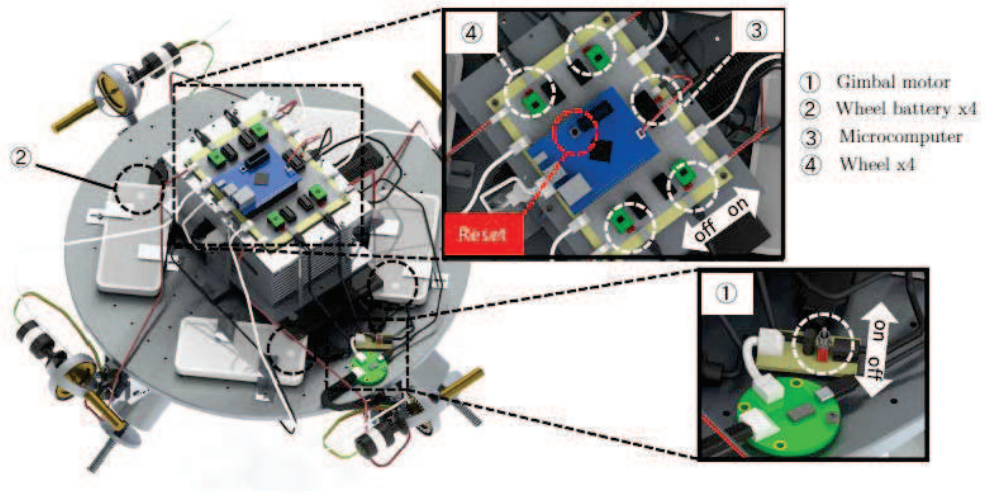


Figure 3.17: Power supplies switch buttons

From the beginning to the end of the experiment
In order to communicate with the wheel control microcomputer, start Tera Term program as shown in 3.18.
In order to communicate with the gimbal motors, launch the B3M Series Manager in 3.19, set the top of the screen to COM6 and the adjacent field to 150,000. If you press *Read All* and the motor information appears, the communication is complete. After confirmation, close B3M Series Manager.
In order to conduct the experiment, the *Matlab* code used in the simulation is converted in C_{++} language. Start Visual C_{++} and select the file. By pressing F5 on your keyboard the experiment starts.
The first part of the experiment consists in moving the gimbals to an arbitrary initial angle, then rotating the wheels until they reach the nominal angular speed and starting the experiment. After that, the 9 DOF sensor is automatically calibrated and the steering law is activated.
To end the experiment, press any key while the setup is working in order to stop the gimbal drive and wheels rotation.
To prepare the setup for the next experiment, press the reset button of the wheel control microcomputer in 3.17 for about 1 s. When the wheels end up rotating, close Visual C_{++} program, lower the table from the column and put on the column lid. Lastly, loosen the compressor plug and let the air out.

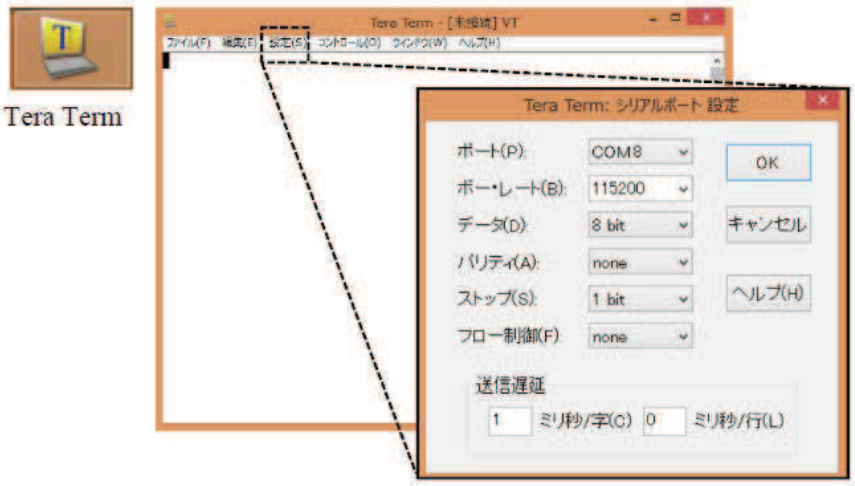


Figure 3.18: Tera Term

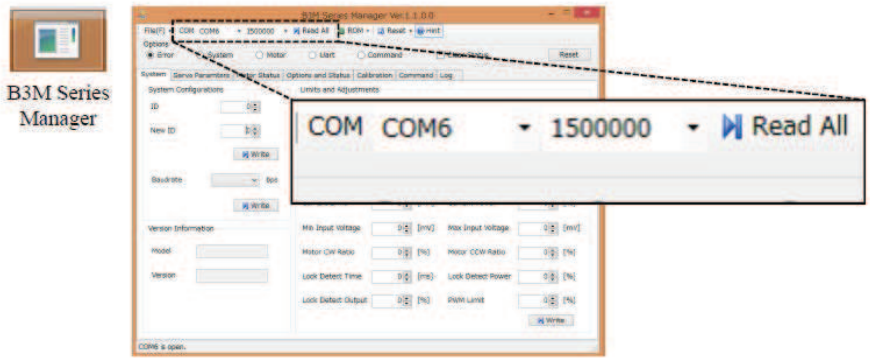


Figure 3.19: B3M Series Manager

3.5.3 Experimental results

The parameters set on the simulations are now chosen to conduct the experiments with the testbed. In particular

$$\beta = 45^\circ$$
$$\omega_N = 10^4 \ \ rpm$$

where β is the skew angle and ω_N is the nominal angular velocity. From ω_N and the wheel moment of inertia $J_W = 3.44 \cdot 10^{-5} \ \ kgm^2$, the wheel angular momentum h_W can be defined as

$$h_W = J_W \omega_N = 0.0361 \ \ Nm \tag{3.14}$$

In all the experiments, a constraint of $\dot{\theta}_{max} = 4 \ \ rad/s$ is imposed on the angular velocity so that there will not be any failure of the gimbal motor. Moreover, the SDA Steering Law parameter is set as $\lambda = 0.01$.

The inertia matrix of the experimental setup is the one used in the simulations, that is

$$J = \begin{bmatrix} 0.9684 & -0.0062 & -0.0087 \\ -0.0062 & 0.9768 & -0.0074 \\ -0.0087 & -0.0074 & 1.300 \end{bmatrix} kgm^2$$

Thus, the PD gains are set as

$$k_d = 12.5$$
$$k_p = 5$$

The data obtained from the 9 DOF sensor are subject to a low-pass filter with a frequency of $3 \ Hz$. The time set for the experiment is $25 \ s$ as shown in the reference trajectory and the time step is $1 \ ms$.

As for the simulation, the initial conditions for the desired trajectory are as follows

$$q_{d0} = [0, 0, 0, 1]^T$$
$$\omega_{d0} = [0, 0, 0]^T \ \ [rad/s]$$
$$\dot{\omega}_{d0} = [0, -0.3, -0.5]^T \ \ [rad/s^2]$$

while the initial conditions for the current attitude, gimbal angles and gimbal rate

are

$$
\begin{aligned}
q_0 &= [0, 0, 0, 1]^T \\
\omega_0 &= [0, 0, 0]^T \ \ [rad/s] \\
\theta_0 &= [0, 0, 0, 0]^T \ \ [rad] \\
\dot{\theta}_0 &= [0, 0, 0, 0]^T \ \ [rad/s]
\end{aligned}
$$

The results are shown in the pictures below.

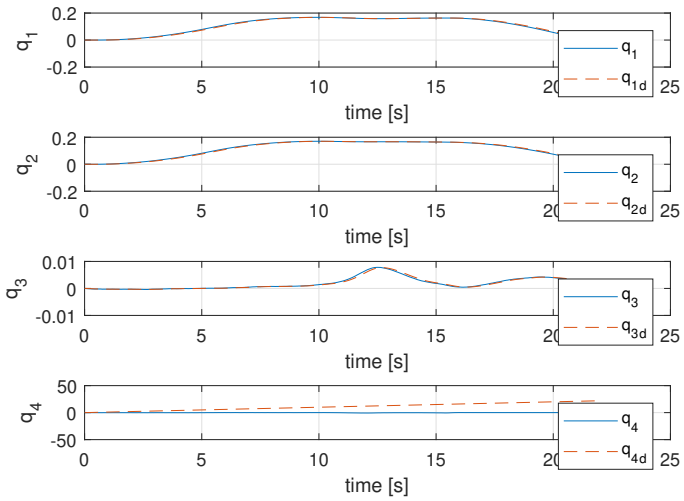


Figure 3.20: Experimental setup: PD-SDA controlled quaternion dynamics

Figures 3.20, 3.21 and 3.22 show the desired attitude, represented by the red and the black dotted lines, in terms of quaternions and angular velocity as in the simulation results, and the present attitude, represented by the blue line. Though the desired attitude is time-varying, the graphs show that the PD controller is able to track it. The effectiveness of this controller is confirmed by the simulation and experimental results achieved up to 15 s. Thus, even if SDA steering law is considered, torque errors can be observed due to the vicinity of the singularity, while after 15 s of experimental test gimbal lock occurs and a performance degradation is highlighted in both 3.21 and 3.22.

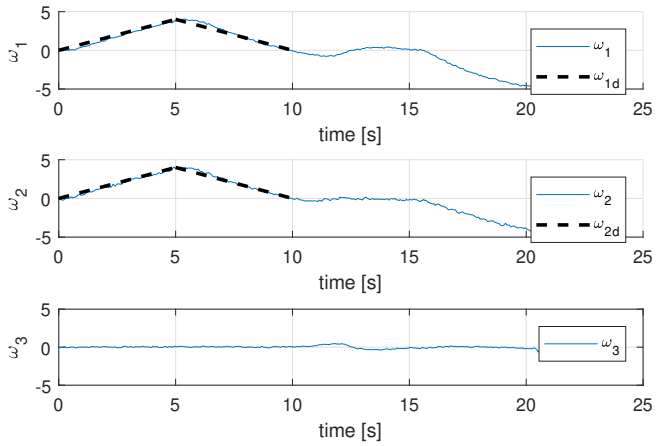


Figure 3.21: Experimental setup: PD-SDA controlled angular velocity ω

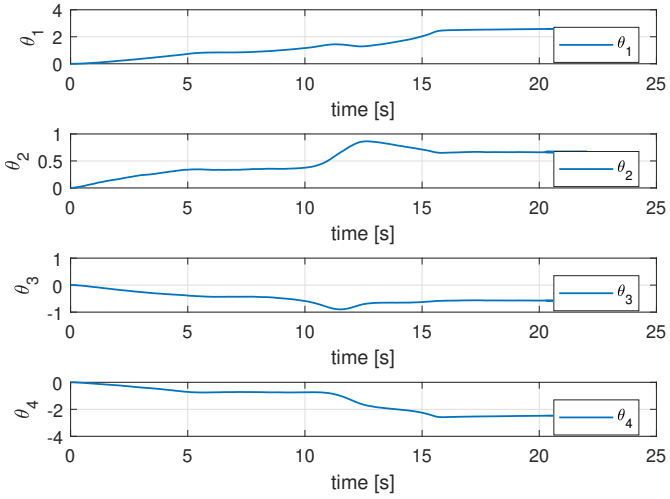


Figure 3.22: Experimental setup: PD-SDA controlled gimbal angles θ

Chapter 4

Model Predictive Control

4.1 Classic, Robust and Stochastic MPC features

As previously introduced in the first chapter, optimal control theory has been developed starting from the calculus of variations, choosing the system parameters that will be used as control inputs in order to optimize a cost index that measures the system performance. Considering linear systems, the optimal control solution has a linear state feedback and a quadratic cost function, and the feedback gains can be computed by solving the steady-state Riccati equation. Moreover, this method can be applied to both continuous time systems, described by sets of differential equations, and to discrete time systems, formulated in terms of difference equation models. On the other side, considering nonlinear systems or systems under constraints on input variables or model states, the benefits of optimal control are more difficult to achieve, as it is not possible to derive an analytic expression for optimal control solution. For this reason, the solution is in general computed numerically and we have to consider approximate solutions, as the analysis is still difficult for almost all cases.

Model Predictive Control (MPC) is one of the most used control strategy due to its capability to guarantee both optimality and computational speed. It is possible to define three main MPC strategies: Classic, Robust and Stochastic. The first and simplest MPC strategy uses the measurements of the current state of the system and a hypothetical future input trajectory to predict future behavior. Therefore, the Degree Of Freedom of the future inputs are used to optimize the cost function. Since feedback control policy is sometimes used instead of a hypothetical future input trajectory, in order to introduce feedback to the optimization strategy, which consists in implementing only the first control input of the optimal control sequence, the whole process is repeated at each time step using updated available information

on the current system state. This repetition is needed to reduce the gap between the predicted and the current system state. Since it is possible to have an imperfect knowledge or unknown variations in the model parameters due to the disturbances affecting the system or uncertainties on both state and control input, MPC can also provide a certain degree of robustness to the control process. In order to ensure nominal stability and convergence properties at the end of a finite prediction horizon, additional conditions were first added as terminal constraints on the predicted model states, even though there were many computational difficulties in the case of nonlinear systems. It was subsequently introduced a stabilizing feedback control law as terminal control law that could be used to define the predicted control inputs at each future step. This law is often used as optimal control law for the actual system dynamics without any constraints or for the unconstrained and linearized dynamics about the desired steady state. Furthermore, terminal constraints are also imposed to guarantee the feasibility of the receding horizon strategy and the satisfaction of the constraints imposed. This kind of constraints generally require that the system state should belong to a subset of state space at the end of the initial finite prediction horizon and, once entered the state of the constrained system, never leaves the subset. All the features explained are the basis of Classic Model Predictive Control (CMPC).

Considering the state model of a system be

$$x_+ = Ax + Bu + Dw \tag{4.1}$$

where x and x_+ indicate respectively the current state and the next time step state, u denotes the control input vector and w is an unknown vector of external disturbances. It is important to highlight that uncertainty is an important feature of control applications, due to the presence of additive disturbances in the system model but it can also be multiplicative in nature, as a result of an imprecise knowledge of the model parameters. Despite the presence of uncertainties, it is essential in both cases (additive or multiplicative) that the properties of closed loop stability and performance are preserved. For these reasons, open loop strategies are computationally convenient but they may be conservative since they ignore information about future uncertainty that will be available to the controller. In the state model considered, whether the numerical values of the matrices A, B and D are not known, it is the case of multiplicative uncertainty.

In this condition, Model Predictive Control first used a linear state feedback law with a parameterization of predicted control inputs, in which the feedback gain is computed online at each time step as an optimization variable. This method includes quadratic constraints as Linear Matrix Inequalities in order to ensure that

the predicted state in contained in ellipsoids in which each constraint is satisfied. Additional optimization variables were then introduced in the form of a perturbation sequence applied to the predicted linear feedback law, even though they considerably increase the online computation which may be overly demanding for systems with many uncertain parameters or high order systems. Defining the predicted state and input trajectories to be optimized through an offline optimization, and maximizing the volume of an ellipsoidal region of attraction, can produce a significant reduction in online computation. However, in this case the uncertainty is bounded by a known set of values, whether the model parameters may lie in particular intervals or in the case they are constant in time. Thus, these uncertainties may be described by a number of linear inequalities and the problem would probably be subject to constraints that have a linear representation such as $Fx + Bu \leq 1$ which expresses the constraints apply to the state, to the control input or both. Whether it is required the controller should have an appropriate robustness to the uncertainties with none but known bounds, MPC theory shifts to Robust Model Predictive Control (RMPC). The main purposes of RMPC are to guarantee stability, constraints satisfaction and convergence of the state vector to a given steady state, together with robustness. Moreover, it is possible, in theory, to compute the optimal solution offline at the regions of the state space defined by different active constraints. However, the number of these regions generally grows exponentially with the size of the system so that this approach may not be practicable for other than low order systems and short prediction horizons. In order to avoid this problem, it is used an approach based on an online interpolation between the current state and a state at which the optimal control law is known. The strategy is so that the active constraint set is updated during the interpolation and a constrained optimization problem is solved at each set change. This programming approach leads to a computational reduction in online steps, though offline steps still require the computation of controllability sets. Thus, it may be preferable to perform the optimization over a restricted class of control policies, in order to have a good approximation of the optimal solution at a fraction of the computational cost.

Furthermore, MPC may incur in significant performance degradation in the presence of disturbances or uncertainties and, in order to cope this disadvantage, a stochastic model has been frequently formulated to represent these uncertainties which let the interest in Stochastic Model Predictive Control increase. Thus, whether the model uncertainties have only a known probabilistic distribution or the constraints are probabilistic in nature, the theory evolves to Stochastic MPC (SMPC). Many examples can be found in process control, financial engineering, electricity generation and distribution, climate control and telecommunications network traffic control ([25],[26],[27],[28],[29],[30],[31],[32]). This situation cannot be described by a hard constraint such as $Fx + Gu \leq 1$, but it can more appropriately

be modelled by probabilistic constraints in the form $Pr\{Fx + Gu \leq 1\} \leq p$ where p represents a given probability. Hence, SMPC ensures that such constraints, together with any additional hard constraint that may be present, are met in closed-loop operation and simultaneously stabilize the system by causing the state to converge to a given steady-state set.

The determination of probability distributions for predicted variables is required for constraint satisfaction, though the calculation is relatively easy only for additive uncertainty, given the linear dependence of predictions on disturbances. Multiplicative uncertainties, on the other hand, represent a more challenging occurrence as the predicted future model states are random variables and they are multiplied by stochastic model parameters in order to generate the successor states. Though a probabilistic description of the disturbance or uncertainty can optimize the performance or the appropriate risk measures, allowing a small probability of constraint violation by introducing chance constraints is more appropriate in some applications. Moreover, hard constraints due to the physical limitations can be considered in the same setup.

One of the problems in Stochastic Model Predictive Control is the method to propagate the uncertainty for evaluating the cost function and the chance constraints, which require suitable techniques to approximate or bound the desired quantities. Though this approach allows uncertainties in the system, the computational cost of the online optimization increases and recursive feasibility may not be guaranteed. Thus, considering linear systems with additive stochastic disturbance, the system is usually decomposed into a deterministic part and an uncertain part. Subsequently, the method consists in computing a confidence region for the uncertain part (using the results for constraint tightening) and then analyzing the given evolution of the uncertainties, directly tightening the constraints. Even if the distribution of model uncertainty is a useful information that should be taken into account in the design of MPC algorithms, in SMPC a certain probability of future constraint violation is generally allowed though the probability distribution of the state prediction at some future time depends on both the current state and the time to go, which leads to a significantly less conservative constraint tightening. Whether some or all of the system constraints are probabilistic in nature, their violations are permitted providing the frequency or generically the number of violations in a given time interval, which should be below a predefined limit. Thus, the violation probability changes from time k to time $k + 1$ even under the same control law and the optimization problem may become infeasible. Even if the probability distribution of the model uncertainty is available, it may be necessary to enforce the conditions that require constraints to be satisfied with certainty. However, this strategy is conservative where constraint violation is allowed and this leads to a weaker performance

and smaller regions of attraction.

Another problem for SMPC is the difficulty of guarantee the recursive feasibility. Since it is essential for stability, in Classic Model Predictive Control described above the recursive feasibility is usually guaranteed through showing that the planned input trajectory remains feasible in each optimization step. Thus, in Stochastic MPC strict recursive feasibility can only be guaranteed for the case that the uncertainty in predicted states and inputs has a finitely supported distribution. As a matter of fact, even if a feasible predicted trajectory exists in the current state, the presence of uncertainty with unbounded support may result in a successor state for which it is impossible to guarantee the existence of a feasible trajectory. However, though this difficulty has characterized earlier SMPC formulations, which used to consider uncertainty with Gaussian distribution due to its mathematical convenience, this assumption is now often not consistent with practice.

4.2 The algorithm used for a variation of Classic MPC

Considering the model dynamics already described in *Chapter 2*, a linear state space formulation is used and it can be described by the following simple equation

$$x(k+1) = Ax(k) + Bu(k)$$

The system is subject to constraints, both on the state vector and on the control input, and they are written in the form

$$u_{min} \leq u \leq u_{max}$$

The desired attitude is described is terms of q and ω, while neither parametric uncertainties nor additive disturbances are considered in this formulation. For a clear explanation of the mathematical notations used below please refer to *Notations* paragraph.

The system is described by the following equations

$$x_p(k+i+1|k) = A_p x_p(k+i|k) + B_p u(k+i|k) \tag{4.2}$$

$$y(k+i|k) = C_p x_p(k+i|k) \tag{4.3}$$

where $x_p \in \mathbb{R}^{n_p}$ is the state vector, $u \in \mathbb{R}^m$ is the control input and $y \in \mathbb{R}^{n_y}$ is the controlled output. Notice that, from this chapter, the dimension of each variable will be explained instead of indicating vectors with bold letters. The matrices A_p,

B_p and C_p are created starting from J_x and J_u in 2.56 after a discretization process. The form $x_p(k+i|k)$ shows the state vector at time $k+i$ calculated at time k.

The constraints imposed to the control input are:

$$|u(k+i|k)| \leq \bar{u}, \forall i \geq 0 \tag{4.4}$$

where $\bar{u} \in \mathbb{R}^m$. It is possible to rewrite this constraint in the form

$$\Psi_u u(k+i|k) \leq \theta_u, \forall i \geq 0 \tag{4.5}$$

where $\theta_u \in \mathbb{R}^{2m}$ and $\Psi_u \in \mathbb{R}^{2m \times m}$. Furthermore, both θ_u and Ψ_u are constant matrices. Considering the case $m = 4$, which represents the dimension of the control input in our simulation, the two matrices are given as:

$$\theta_u = \begin{bmatrix} \bar{u}^{(1)} & \bar{u}^{(1)} & \bar{u}^{(2)} & \bar{u}^{(2)} & \bar{u}^{(3)} & \bar{u}^{(3)} & \bar{u}^{(4)} & \bar{u}^{(4)} \end{bmatrix}^T \tag{4.6}$$

$$\tag{4.7}$$

$$\Psi_u = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}^T \tag{4.8}$$

The constraints imposed on the state vector are given in a similar formulation and can be expressed as follows:

$$\Psi_{x_p} x_p(k+i+1|k) \leq \theta_x, \forall i \geq 0 \tag{4.9}$$

where $\Psi_{x_p} \in \mathbb{R}^{n_{\Psi_x} \times n_p}$. Ψ_{x_p} is a constant matrix as well as θ_x, in which all the elements are positive.

For a given reference signal $r(k|k) \in \mathbb{R}^{n_y}$, we have to design the control law

$$u(k+1|k) = \mathcal{K}(x_p(k+i|k), r(k|k)) \tag{4.10}$$

that minimizes the deviation between $r(k|k)$ and $y(k|k)$ at each time step and achieves

$$\lim_{x \to \infty} y(k|k) = \bar{r} \tag{4.11}$$

if $r(k|k) = \bar{r}$, $\forall k \geq T_f$, where $\bar{r}$ is a constant vector and T_f is a positive constant.

In order to eliminate the steady state error, the following integrator is introduced to the feedback system

$$x_c(k+i+1|k) = x_c(k+i|k) + e(k+i|k) \tag{4.12}$$

$$e(k+i|k) = w(k|k) - y(k+i|k) \tag{4.13}$$

where $x_c \in \mathbb{R}^{n_c}$ and $w(k|k) \in \mathbb{R}^{n_y}$ is a virtual reference signal.

Starting from the system in 4.2 and 4.3, and then in 4.12 and 4.13, where both x_p and x_c change at each time step, it is possible to introduce the following matrices

$$A := \begin{bmatrix} A_p & 0 \\ -C_p & I \end{bmatrix}, \quad B := \begin{bmatrix} B_p \\ 0 \end{bmatrix}, \quad E := \begin{bmatrix} 0 \\ I \end{bmatrix}$$
$$C := \begin{bmatrix} -C_p & 0 \end{bmatrix}, \quad D_W := \begin{bmatrix} I \end{bmatrix} \tag{4.14}$$

in order to rewrite the system in the form

$$x(k+i+1|k) = Ax(k+i|k) + Bu(k+i|k) + Ew(k|k) \tag{4.15}$$
$$e(k+i|k) = Cx(k+i|k) + D_w w(k|k) \tag{4.16}$$

where $x := [x_p^T; x_c^T]^T$.

In order to define the cost function, 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

$$\Pi = A\Pi + B\Gamma + E \qquad 0 = C\Pi + D_w$$

This condition ensures that the setpoint tracking control problem is solvable in case of linear systems.

After introducing the vector x as above, it is then possible to rewrite the state constraints expressed in 4.9 in the form

$$\Psi_x x(k+i+1|k) \leq \theta_x, \forall i \geq 0 \tag{4.17}$$

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

The steady state value of the reference signal satisfies $\Psi_u \Gamma \bar{r} \leq \theta_u$ and $\Psi_x \Pi \bar{r} \leq \theta_x$. Thus, the control signal u and the state vector x satisfy the constraints in 4.5 and 4.17 in the steady state.

Thus, introducing ξ and v expressed as

$$\xi(k+i|k) := x(k+i|k) - \Pi w(k|k) \tag{4.18}$$
$$v(k+i|k) := u(k+i|k) - \Gamma w(k|k) \tag{4.19}$$

the error system can now be derived as

$$\xi(k+i+1|k) = A\xi(k+i|k) + Bv(k+i|k) \tag{4.20}$$
$$e(k+i|k) = C\xi(k+i|k) \tag{4.21}$$

The following cost function is adopted

$$J(k) := \sum_{i=0}^{H_s-1} \{\|\xi(k+i+1|k)\|_Q^2 + \|v(k+i|k)\|_R^2\} + \\ + \|\xi(k+H_s|k)\|_P^2 + \|\Pi(w(k|k) - r(k|k))\|_M^2 \tag{4.22}$$

where Q, R and P are positive definite matrices and H_s is the finite horizon.

The last step of this process is to minimize the cost function assuming $u(k+i+1|k) = u(0|k)$ at each sampling time. The implementation process will be fully analyzed in the next paragraph.

4.3 Implementation process

The implementation of the variation of the Classic Model Predictive Control algorithm has been made entirely on *Matlab*, without using *Simulink* block diagrams, which have been very helpful during the dynamic test. The reason of this choice is the same as for PD algorithm test, the necessity of converting the *Matlab* code in C_{++} in order to do the experiments on the testbed.

The first step is defining the constant parameters and then introducing the desired attitude that needs to be tracked with the application of this controller. The attitude used for MPC is the same as for PD controller and dynamic test and it can be seen in 2.8 and 2.10. The initial conditions imposed also for this desired time varying signal are

$$\begin{aligned} q_{d_0} &= [0,0,0,1]^T \\ \omega_{d_0} &= [0,0,0]^T \ \ [rad/s] \\ \dot{\omega}_{d_0} &= [0,-0.3,-0.5]^T \ \ [rad/s^2] \\ \theta_{d_0} &= [0,0,0,0]^T \ \ [rad] \\ \dot{\theta}_{d_0} &= [0,0,0,0]^T \ \ [rad/s] \end{aligned}$$

Thus, using J_x and J_u matrices as expressed in 2.56, a discretization is applied in order to create A_p, B_p and C_p matrices as follows

$$\begin{aligned} A_p &= dt \cdot J_x + I_{n_{x_p} \times n_{x_p}} \\ B_p &= dt \cdot J_u \\ C_p &= I_{n_{x_p} \times n_{x_p}} \end{aligned}$$

where $dt = 0.001s$. J_x and J_u matrices are calculated in continuous time though a discrete time calculation is needed for this analysis. Hence, as clearly explained in the previous paragraph, A, B, C, D_w and E matrices are created as

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

$$C := \begin{bmatrix} -C_p & 0 \end{bmatrix}, \quad D_W := \begin{bmatrix} I \end{bmatrix} \tag{4.23}$$

Furthermore, $A(\theta)$ matrix in 2.45 is considered time-varying as the parameter $\theta \in \mathbb{R}^m$ which figures in the matrix values is updated at each time step. For this reason, J_u matrix defined as

$$J_u = \begin{bmatrix} 0_{4\times 4} \\ \\ -h_W J^{-1} A(\theta) \end{bmatrix} \tag{4.24}$$

is considered parameter dependent and consequently B_p and B follow the same indication.

It is then possible to introduce Π and Γ matrices starting from the matrices in 4.23. Since Π and Γ are required for the assumption that ensures the solvability of the setpoint tracking control problems in case of linear systems, their values needs to be related to the ones of the dynamic matrices stated above. For this reason, considering the dimension $n = n_p + n_c$, we first define

$$AA := \begin{bmatrix} I_{n\times n} - A & -B \\ C & 0_{n_p \times m} \end{bmatrix}, \quad BB := \begin{bmatrix} E \\ D_w \end{bmatrix}$$

where $AA \in \mathbb{R}^{(n+n_p)\times(n+m)}$ and $BB \in \mathbb{R}^{(n+n_p)\times n_p}$. These matrices, with the help of the *Matlab* command *linsolve*, are used to create $X \in \mathbb{R}^{(n+m)\times n_p}$ matrix, solution of the QR factorization. Indeed, *linsolve* command solves the linear system $AX = B$ using Lower-Upper (LU) factorization with partial pivoting when A is a square matrix, or QR factorization with column pivoting otherwise. With this step, Π and Γ are already calculated, as

$$\Pi := X(1 : n, :) \tag{4.25}$$

$$\Gamma := X(n + 1 : n + m, :) \tag{4.26}$$

and they can satisfy the conditions

$$\Pi = A\Pi + B\Gamma + E$$
$$0 = C\Pi + D_w$$

The input constraint in 4.5 during $0 \leq i \leq H_s - 1$ can be rewritten in the following form

$$\tilde{\Psi}_u(\tilde{v} + \tilde{\Gamma} w(k|k)) \leq \tilde{\theta}_u, \forall i = 1, \ldots, mH_s \tag{4.27}$$

where $\tilde{\Psi}_u :=$block-diag$[\Psi_u, \ldots, \Psi_u]$, $\tilde{\theta}_u := [\theta_u^T, \ldots, \theta_u^T]^T$ and $\tilde{\Gamma} := [\Gamma^T, \ldots, \Gamma^T]^T$.

From 4.20 it is possible to obtain the vector of future predictions of the error state, up to a horizon $H_s = 10$ as $\tilde{\xi} = \mathbb{A}\xi(k|k) + \mathbb{A}_B\tilde{v}$, where $\tilde{\xi} := [\xi(k+1|k)^T, \ldots, \xi(k + H_s|k)^T]^T$ and

$$\mathbb{A} := \begin{bmatrix} A \\ \vdots \\ A^{H_s} \end{bmatrix}, \quad \mathbb{A}_B := \begin{bmatrix} B & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{H_s-1}B & \cdots & B \end{bmatrix} \tag{4.28}$$

Moreover, the state constraints in 4.17 during $0 \leq i \leq H_s - 1$ can be rewritten in the following form

$$\tilde{\Psi}_x(\mathbb{A}\xi(k|k) + \mathbb{A}_B\tilde{v} + \tilde{\Pi} w(k|k)) \leq \tilde{\theta}_x \tag{4.29}$$

where $\tilde{\Psi}_x :=$block-diag$[\Psi_x, \ldots, \Psi_x]$, $\tilde{\theta}_x := [\theta_x^T, \ldots, \theta_x^T]^T$ and $\tilde{\Pi} := [\Pi^T, \ldots, \Pi^T]^T$.

In order to calculate P matrix which is needed to minimize the cost function, a Linear Matrix Inequality (LMI) system is solved. The LMI features are shown in E. Considering the dimensions $n_q = m = 4$ and $n_\omega = 3$, two arbitrary positive definite matrices, Q and R, are defined as follows

$$Q = \begin{bmatrix} 10 \cdot I_{n_q \times n_q} & 0_{n_q \times n_\omega} & 0_{n_q \times n_p} \\ 0_{n_\omega \times n_q} & 10^{-1} \cdot I_{n_\omega \times n_\omega} & 0_{n_\omega \times n_p} \\ 0_{n_p \times n_q} & 0_{n_p \times n_\omega} & 10 \cdot I_{n_p \times n_p} \end{bmatrix} \tag{4.30}$$

$$R = \begin{bmatrix} 10 \cdot I_{m \times m} \end{bmatrix} \tag{4.31}$$

where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$.

Thus, we introduce a bounded convex set $\mathcal{W}_b$, which is a subset of $\{w \in \mathbb{R}^{n_y} : \Psi_x \Pi w \leq \theta_x, |\Gamma w| \leq \bar{u} - \rho\}$, where $\rho > 0$. The parameter ρ, from its definition, needs to satisfy $0 < \rho \leq \bar{u}$. Moreover, since ρ restricts the maximum value of the control signal of the error system, in order to achieve higher tracking performance, a larger value should be chosen. Therefore, ρ also restricts the maximum amplitude of the virtual reference signal as $\bar{w} \in \mathcal{W}_b$. For this reason, a smaller value needs to be chosen in order to track a larger reference signal. This discussion leads to the fact that the parameter ρ is determined by considering the trade off between the maximum amplitude of the virtual reference signal and the tracking control performance.

Considering the error system in 4.20 and 4.21, we assume that $w(k|k) = \bar{w}$, $\forall k \geq 0$, where $\bar{w}$ is an arbitrary constant vector such that $\bar{w} \in \mathcal{W}_b$ with $\rho > 0$. The values chosen for $\bar{w}$ are:

$$\bar{w} = [0, 0, 0, 1, 0, 0, 0]^T \tag{4.32}$$

Thus, considering given positive definite matrices Q and R as above, and positive scalars $\bar{\eta}_2$ and τ, there exist a matrix Y and a positive definite matrix S that satisfy

$$\begin{bmatrix} S & * & * & * \\ AS + BY & S & * & * \\ Y & 0 & R^{-1} & * \\ AS + BY & 0 & 0 & Q^{-1} \end{bmatrix} > 0 \tag{4.33}$$

$$\begin{bmatrix} \frac{1}{\bar{\eta}_2} S & * \\ Y^{(l)} & \rho^{(l)2} \end{bmatrix} \geq 0, \forall l = 1, \ldots, m \tag{4.34}$$

$$\begin{bmatrix} \tau S & * & * \\ 0 & \theta_x^{(j)2} - \tau\bar{\eta}_2 & * \\ \Psi_x^{(j)}(AS + BY) & \Psi_x^{(j)}\Pi v_i & 1 \end{bmatrix} \geq 0, \forall i = 1, \ldots, n_s, \forall j = 1, \ldots, n_x \tag{4.35}$$

where $*$ stands for symmetric block in matrix inequalities.

From this LMI system we assume that $\xi(k + H_s|k) \in \mathcal{E}(P, 0, \bar{\eta}_2)$, where

$$P = S^{-1} \tag{4.36}$$

and $P \in \mathbb{R}^{n \times n}$. Then, by applying the control law $v(k + i|k) = F\xi(k + i|k)$ to the error system in 4.20 and 4.21, where $F = YS^{-1}$, the set $\mathcal{E}(P, 0, \bar{\eta}_2)$ becomes positively invariant and it is possible to hold the following relations

$$J_2(k) := \sum_{i=H_s}^{\infty} \{\|\xi(k + i + 1|k)\|_Q^2 + \|v(k + i|k)\|_R^2\} < \|\xi(k + H_s|k)\|_P^2$$

$$|u(k + i|k)| \leq \bar{u}, \quad \forall i \geq H_s$$
$$\Psi_x x(k + i + 1|k) \leq \theta_x, \quad \forall i \geq H_s$$
$$\|\xi(k + H_s|k)\|_P^2 \leq \bar{\eta}_2$$

The proof of this theorem can be found in F.

Based on the discussion above, we design a feedback gain $F = YS^{-1}$ which makes the state variables achieve fast convergence and expands the size of the set $\mathcal{E}(P, 0, \bar{\eta}_2)$ by suitably choosing the parameters Q, R, $\bar{\eta}_2$ and τ. This formulation is an LMI optimization problem. Also, as we can see from 4.35, the positive scalars τ and $\bar{\eta}_2$ need to be chosen so that the inequalities

$$\theta_x^{(j)2} - \tau\bar{\eta}_2 \geq 0, \quad \forall j = 1, \ldots, n_x$$

are satisfied. In this implementation process, considering the maximum allowable value of the rotational speed $\omega_{max} = 5\ \ [rad/s]$, we define

$$\bar{\eta}_2 = 10^5 \tag{4.37}$$

$$\tau = \omega_{max}/\bar{\eta}_2 = 5 \cdot 10^{-5} \tag{4.38}$$

Hence, P matrix is used as a final constraint for the new $\tilde{Q}$ and $\tilde{R}$ matrices involved in the second LMI system used to minimize the cost function. $\tilde{Q}$ and $\tilde{R}$ are defined as

$$\tilde{Q} = \begin{bmatrix} Q & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & Q & 0 \\ 0 & \cdots & 0 & P \end{bmatrix} \tag{4.39}$$

$$\tilde{R} = \begin{bmatrix} R & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R \end{bmatrix} \tag{4.40}$$

where $\tilde{Q} \in \mathbb{R}^{n_{\tilde{Q}} \times n_{\tilde{Q}}}$ and $\tilde{R} \in \mathbb{R}^{n_{\tilde{R}} \times n_{\tilde{R}}}$, with $n_{\tilde{Q}} = n \cdot H_s$ and $n_{\tilde{R}} = m \cdot H_s$.

By using the representation of $\mathbb{A}$ and $\mathbb{A}_b$ in 4.28 and applying the Schur complement in [33], it can be shown that the inequality

$$J_1(k) := \sum_{i=0}^{H_s-1} \{\|\xi(k+i+1|k)\|_Q^2 + \|v(k+i|k)\|_R^2\} < \tilde{\eta}_1$$

can be rewritten as

$$\begin{bmatrix} \tilde{\eta}_1 & * & * \\ \tilde{v} & \tilde{R}^{-1} & * \\ \mathbb{A}\xi(k|k) + \mathbb{A}_B\tilde{v} & 0 & \tilde{Q}^{-1} \end{bmatrix} > 0$$

where $\tilde{\eta}_1$ is a positive scalar. Moreover, by applying the Schur complement to the inequality

$$\|\xi(k+H_s|k)\|_P^2 \leq \bar{\eta}_2$$

it is possible to write

$$\begin{bmatrix} \bar{\eta}_2 & * \\ \xi(k+H_s|k) & S \end{bmatrix} \geq 0$$

Since $\xi(k+H_s|k)$ can be represented as $\xi(k+H_s|k) = A^{H_s}\xi(k|k) + \mathbb{A}_H\tilde{v}$, where

$$\mathbb{A}_H := \begin{bmatrix} A^{H_s-1}B \\ \vdots \\ B \end{bmatrix} \tag{4.41}$$

we obtain an LMI with respect to the variable $\tilde{v}$. Thus, in order to solve

$$\min_{\tilde{v},\tilde{w},\tilde{\eta_1},\tilde{\eta_2},\tilde{\eta_3},\tilde{x}_c} \quad \tilde{\eta}_1 + \tilde{\eta}_2 + \tilde{\eta}_3 \tag{4.42}$$

the LMI system is described by the following inequalities

$$\begin{bmatrix} \tilde{\eta}_1 & * & * \\ \tilde{v} & \tilde{R}^{-1} & * \\ \mathbb{A}\tilde{\xi}(\tilde{x}_c,\tilde{w}) + \mathbb{A}_B\tilde{v} & 0 & \tilde{Q}^{-1} \end{bmatrix} > 0 \tag{4.43}$$

$$\begin{bmatrix} \tilde{\eta}_2 & * \\ A^{H_s}\tilde{\xi}(\tilde{x}_c,\tilde{w}) + \mathbb{A}_H\tilde{v} & S \end{bmatrix} \geq 0 \tag{4.44}$$

$$\begin{bmatrix} \tilde{\eta}_3 & * \\ \Pi(\tilde{w} - r(k|k)) & M^{-1} \end{bmatrix} \geq 0 \tag{4.45}$$

$$\tilde{\Psi}_u(\tilde{v} + \tilde{\Gamma}\tilde{w}) \leq \tilde{\theta}_u \tag{4.46}$$

$$\tilde{\Psi}_x(\mathbb{A}\tilde{\xi}(\tilde{x}_c,\tilde{w}) + \mathbb{A}_B\tilde{v} + \tilde{\Pi}\tilde{w}) \leq \tilde{\theta}_x \tag{4.47}$$

$$|\Gamma\tilde{w}| \leq \theta_u - \rho \tag{4.48}$$

where the matrix $M = P + k \cdot I_{n\times n}$, the vector $\tilde{\xi}(\tilde{x}_c,\tilde{w}) := [x_p(k|k)^T, \tilde{x}_c^T]^T - \Pi\tilde{w}$ and $\rho := [\rho_1, \cdots, \rho_m]^T$. Moreover, the condition 4.45 derives from the inequality $\|\Pi(\tilde{w} - r(k|k))\|_M^2 \leq \tilde{\eta}_3$ and the conditions 4.46, 4.47 derive from 4.27 and 4.29, respectively.

Once calculated $\bar{v}$, it is possible to apply the condition

$$\tilde{v}(k+i+1|k) = \tilde{v}(0|k) \tag{4.49}$$

so that only the first value of the current $\tilde{v}$ is chosen for the next time step. Thus, from the relation 4.19 we can simply derive $\tilde{u}$ as follows

$$\tilde{u}(k+i|k) = \tilde{v}(k+i|k) + \Gamma\tilde{w}(k|k) \tag{4.50}$$

where $\bar{u}$ is exactly the control input which consists in the angular velocity of the gimbal angles $\dot{\theta}$. Furthermore, a comparison between $\dot{\theta}$ and $\dot{\theta}_{max}$ is needed in order to correctly choose the updated value which will be used in the following step. In particular, if $\dot{\theta} \geq \dot{\theta}_{max}$, $\dot{\theta}_{max}$ is chosen; if $\dot{\theta} \leq -\dot{\theta}_{max}$, $-\dot{\theta}_{max}$ is chosen.

Finally, with the updated value of $\dot{\theta}$ it is possible to calculate the $A(\theta)$ matrix used in the dynamic equation of $\dot{\omega}$ in 2.46 and consequently solve the quaternion equation of $\dot{q}$ in 2.47 in order to achieve the desired attitude.

4.4 Implementation and simulation results

The MPC controller has been fully implemented on *Matlab* using the dynamics already fulfilled in the dynamic simulation test, as for the PD controller with SDA steering law.

The dynamic constant parameters used in the simulation are

$$\begin{aligned} \beta &= 45^\circ \\ h_W &= 0.0361 \\ \dot{\theta}_{max} &= 4 \ [rad/s] \\ \omega_{max} &= 1.3 \ [rad/s] \end{aligned}$$

$$J = \begin{bmatrix} 0.9684 & -0.0062 & -0.0087 \\ -0.0062 & 0.9768 & -0.0074 \\ -0.0087 & -0.0074 & 1.3000 \end{bmatrix}$$

The MPC parameters used for the minimization of the cost function are

$$\begin{aligned} H_s &= 10 \\ \eta_2 &= 10^4 \\ \tau &= \frac{\omega_{max}}{\eta_2} \end{aligned}$$

$$Q = \begin{bmatrix} 10 \cdot I_{4\times4} & 0_{4\times3} & 0_{4\times7} \\ 0_{3\times4} & 10^{-1} \cdot I_{3\times3} & 0_{3\times7} \\ 0_{7\times4} & 0_{7\times3} & 10 \cdot I_{7\times7} \end{bmatrix}$$

$$R = \begin{bmatrix} 10 \cdot I_{4\times4} \end{bmatrix}$$

The initial conditions for the desired attitude are as follows

$$\begin{aligned} q_{d_0} &= [0, 0, 0, 1]^T \\ \omega_{d_0} &= [0, 0, 0]^T \ [rad/s] \\ \dot{\omega}_{d_0} &= [0, -0.3, -0.5]^T \ [rad/s^2] \end{aligned}$$

while the initial conditions for the current attitude, gimbal angles and gimbal rate

are

$$
\begin{aligned}
q_0 &= [0, 0, 0, 1]^T \\
\omega_0 &= [0, 0, 0]^T \ [rad/s] \\
\theta_0 &= [0, 0, 0, 0]^T \ [rad] \\
\dot{\theta}_0 &= [0, 0, 0, 0]^T \ [rad/s] \\
x_{c_0} &= [0, 0, 0, 0, 0, 0, 0]^T
\end{aligned}
$$

Lastly, the simulation is made considering

$$
\begin{aligned}
t_0 &= 0 \ [s] \\
t_f &= 20 \ [s] \\
dt &= 10^{-2} \ [s]
\end{aligned}
$$

The results are shown in the following page.

As in the previous simulations, 4.1 and 4.2 show the desired attitude, represented by the red line, and the current attitude, represented by the blue line. The MPC controller is able to track the time-varying desired attitude, even with some fluctuations around the nominal value due to the vicinity of the singularity. Nevertheless, the fluctuations have a small width so that a good tracking performance is achieved. Thus, the effectiveness of both controllers, PD-SDA and MPC, is confirmed by simulations.

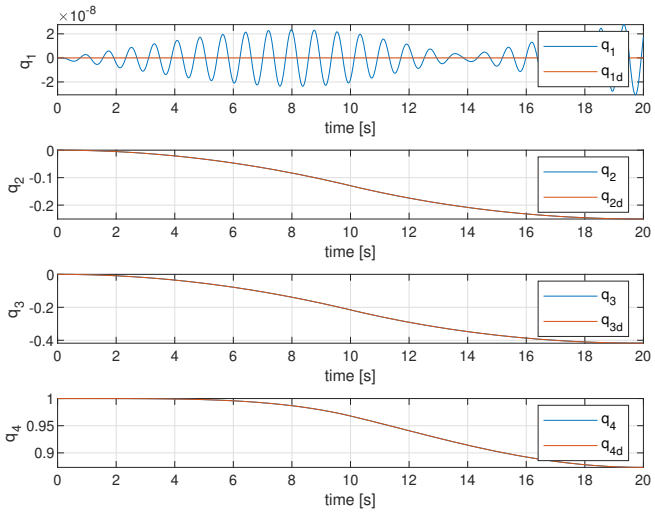


Figure 4.1: Quaternion attitude controlled by MPC

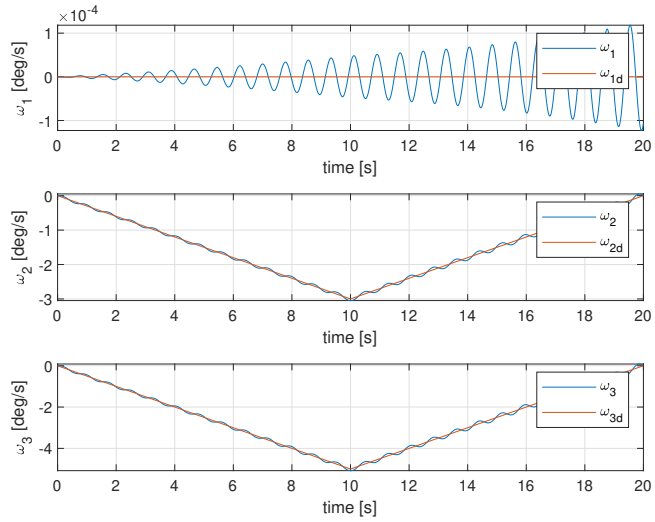


Figure 4.2: Angular velocity ω controlled by MPC

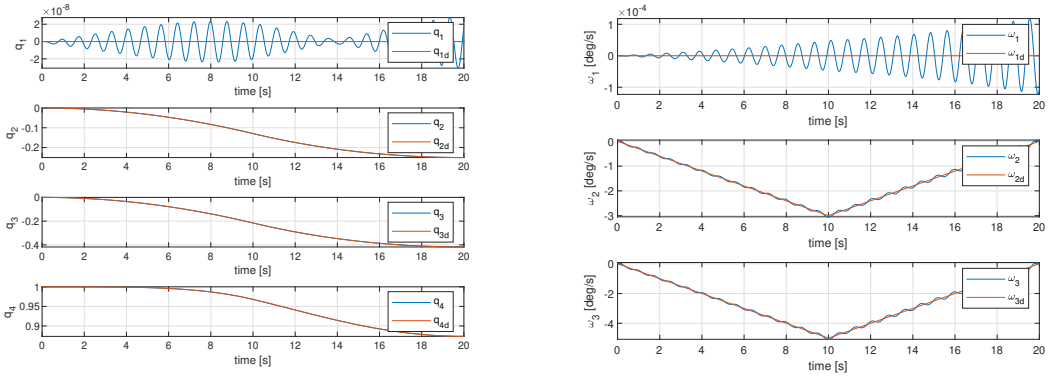
4.5 Simulation results with parameter variation

In this section, an analysis of the simulation results with parameter variation is considered. As for the SDA analysis, each parameter is chosen to vary keeping all the others at the nominal value. The variations of the parameters are considered in the following order:

- Variation of the inertia matrix J_B
- Variation of the CMG skew angle β
- Variation of x_{c_0} vector
- Variation of Q, R matrices

4.5.1 Variation of the inertia matrix J_B

The analysis focuses on the variation of the inertia matrix J_B, considering three different matrices which represent three different system dynamics. The results achieved for the three different J_B matrices will then be compared to the nominal results.



(a) *Quaternion dynamics* (b) *Angular velocity* ω

Figure 4.3: Simulation results in terms of q and ω considering the nominal values of J_B, β, x_{c_0}, Q and R

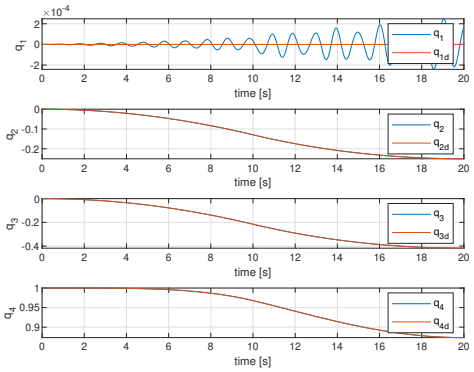
Considering J_{B_n} the nominal J_B matrix written in the previous section, the three different matrices considered are:

$$J_{B_1} = \begin{bmatrix} 0.5684 & -0.0124 & -0.0042 \\ -0.0124 & 0.5768 & -0.0140 \\ -0.0042 & -0.0140 & 1.5000 \end{bmatrix}$$

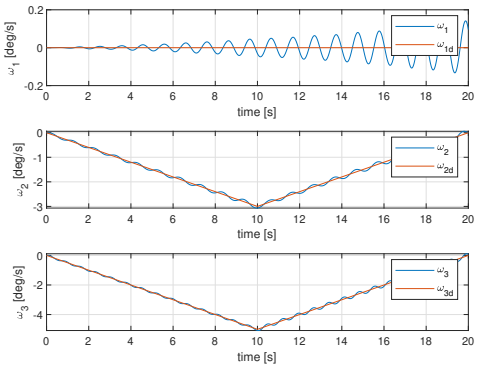
$$J_{B_2} = \begin{bmatrix} 0.8684 & -0.0062 & -0.0160 \\ -0.0062 & 0.1768 & -0.0004 \\ -0.0160 & -0.0004 & 1.8000 \end{bmatrix}$$

$$J_{B_3} = \begin{bmatrix} 1.0000 & -0.0050 & -0.0100 \\ -0.0050 & 1.0000 & -0.0150 \\ -0.0100 & -0.0150 & 1.0000 \end{bmatrix}$$

The results, in terms of quaternions and angular velocity, are shown in the pictures below.

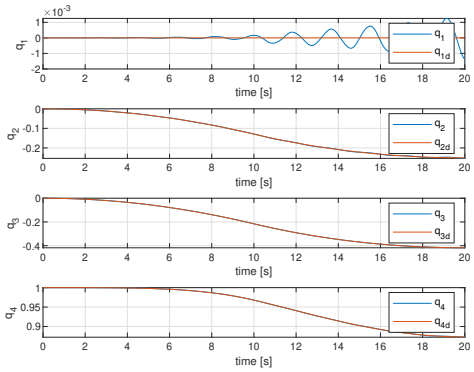


(a) *Quaternion dynamics with* J_{B_1}

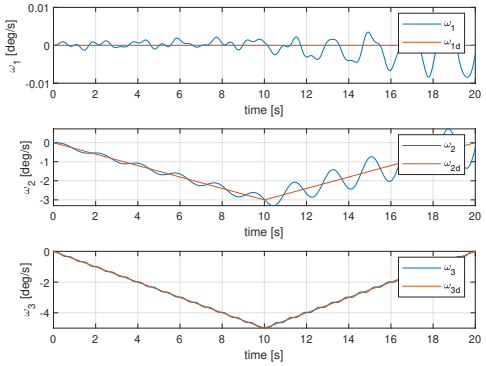


(b) *Angular velocity* ω *with* J_{B_1}

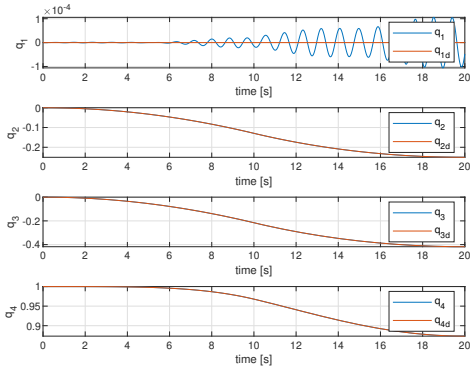
Figure 4.4: Simulation results in terms of q and ω considering J_{B_1} matricx



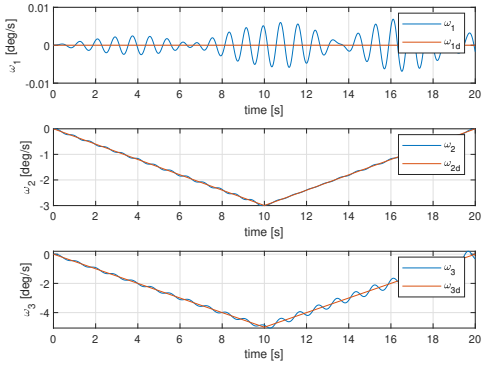
(a) *Quaternion dynamics with* J_{B_2}



(b) *Angular velocity* ω *with* J_{B_2}



(c) *Quaternion dynamics with* J_{B_3}



(d) *Angular velocity* ω *with* J_{B_3}

Figure 4.5: Simulation results in terms of q and ω considering J_{B_2} and J_{B_3} matrices

As we can see in 4.3, 4.4 and 4.5, the tracking performance of the three different dynamics are as worse as the value of J_B gets away from the nominal one. The desired attitude is still tracked but the fluctuations have a more important width so that the error makes the controller ineffective without changing MPC parameters other than the inertia matrix J_B.

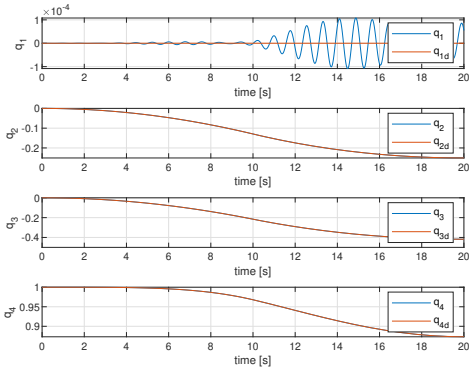
4.5.2 Variation of the CMG skew angle β

The second analysis focuses on the variation of the Control Moment Gyro skew angle β, which is a variation also related to the dynamics of the system.

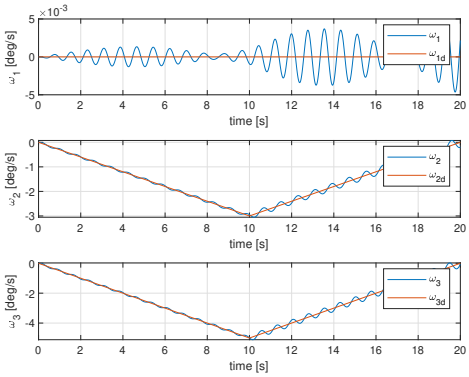
Considering $\beta_n = 45°$ the nominal value of β written in the previous section, the three different angles considered are:

$$\beta_1 = 40°$$
$$\beta_2 = 44°$$
$$\beta_3 = 50°$$

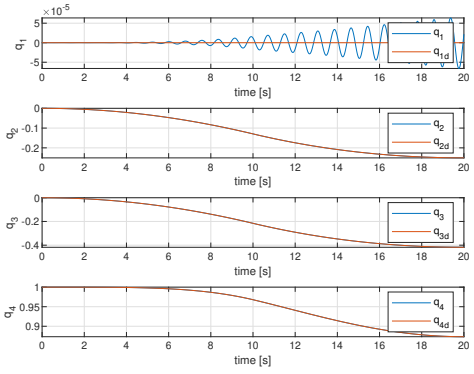
The results are shown in the pictures below.



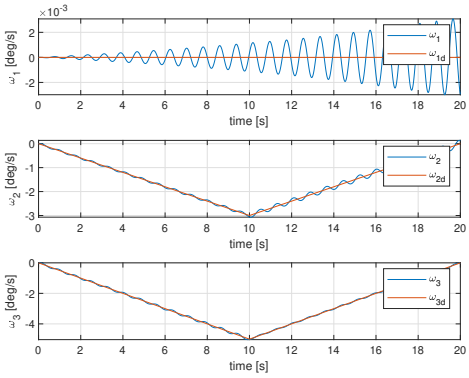
(a) *Quaternion dynamics with* β_1



(b) *Angular velocity* ω *with* β_1

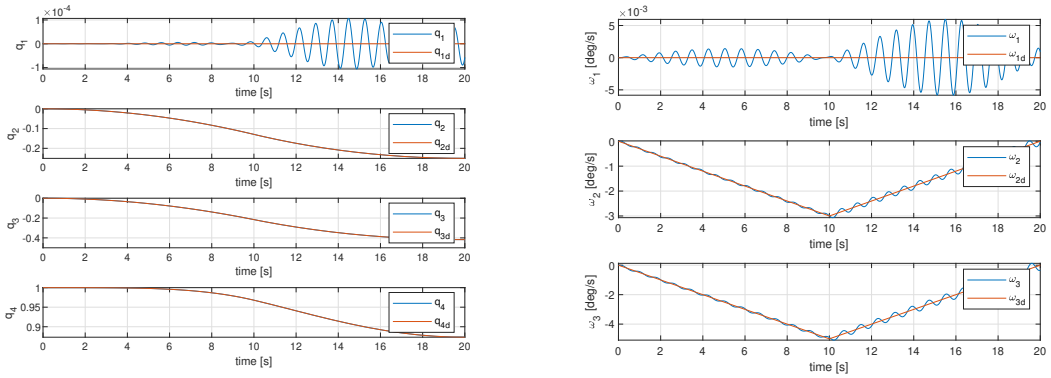


(c) *Quaternion dynamics with* β_2



(d) *Angular velocity* ω *with* β_2

Figure 4.6: Simulation results in terms of q and ω considering β_1 and β_2



(a) *Quaternion dynamics with* β_3 (b) *Angular velocity* ω *with* β_3

Figure 4.7: Simulation results in terms of q and ω considering β_3

From Fig. 4.6 and 4.7 it is possible to notice that the results achieved with $\beta_2 = 44°$ are similar to the nominal ones. Moreover, both β_1 and β_3 have 5° less and more than the nominal value, respectively, and this leads to a slightly different trend of the tracking performance. Nevertheless, it is still possible to ensure a good accuracy to the attitude of the system.

4.5.3 Variation of x_{c_0} vector

The third analysis focuses on the variation of the vector x_{c_0} related to the error dynamics of the system.

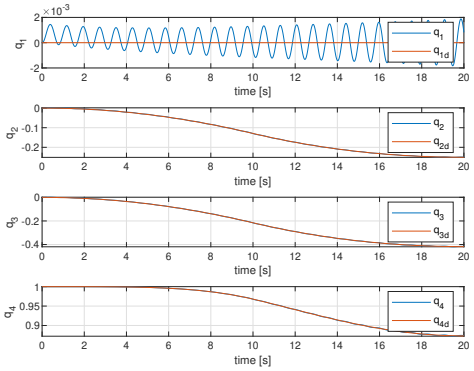
Considering $x_{c_{0,n}}$ the nominal value of x_{c_0} set as

$$x_{c_{0,n}} = [0, 0, 0, 0, 0, 0, 0]^T$$

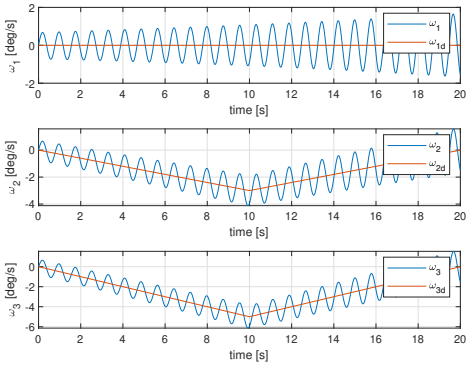
where $x_0 = [x_{p_0}; x_{c_0}]^T$, the three different values assigned to the initial condition of the error dynamic vector are:

$$\begin{aligned} x_{c_{0,1}} &= 0.1 \cdot [1, 1, 1, 1, 1, 1, 1]^T \\ x_{c_{0,2}} &= 0.001 \cdot [1, 1, 1, 1, 1, 1, 1]^T \\ x_{c_{0,3}} &= 0.0001 \cdot [1, 1, 1, 1, 1, 1, 1]^T \end{aligned}$$

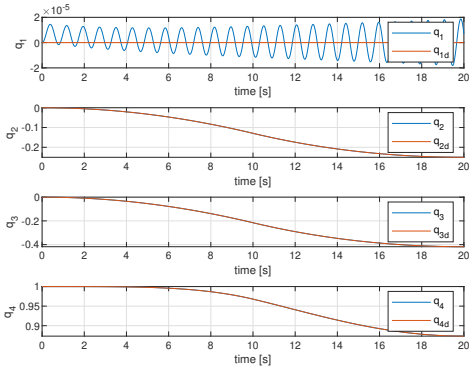
The results are shown in the next page.



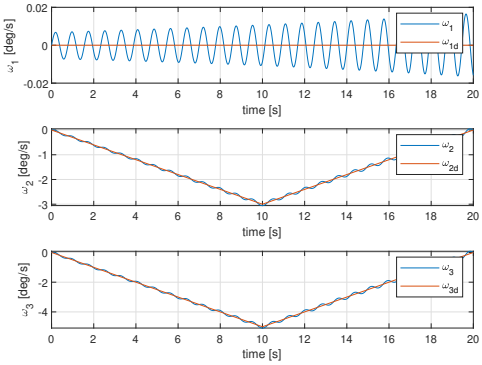
(a) *Quaternion dynamics with* $x_{c_{0,1}}$



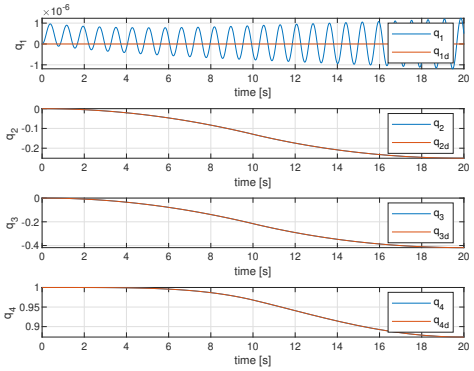
(b) *Angular velocity* ω *with* $x_{c_{0,1}}$



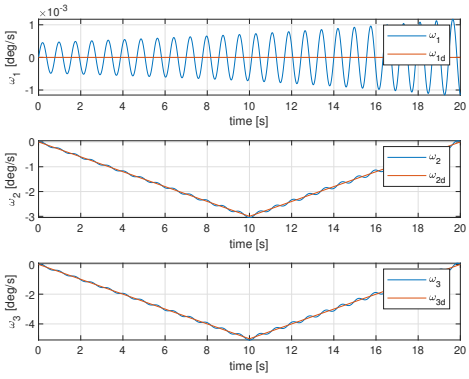
(c) *Quaternion dynamics with* $x_{c_{0,2}}$



(d) *Angular velocity* ω *with* $x_{c_{0,2}}$



(e) *Quaternion dynamics with* $x_{c_{0,3}}$



(f) *Angular velocity* ω *with* $x_{c_{0,3}}$

Figure 4.8: Simulation results in terms of q and ω considering three different x_{c_0} vectors

Fig. 4.8 show how the tracking performance improves as the values related to the error dynamics get smaller. This is an intuitive result and perfectly on point with the analysis made so far.

4.5.4 Variation of Q, R matrices

The fourth and last analysis focuses on the variation of Q matrix and R matrix, needed to minimize the MPC cost function and optimize the control problem.

Considering Q_n and R_n the nominal values of Q and R, respectively, previously set as

$$Q_n = \begin{bmatrix} 10 \cdot I_{4\times4} & 0_{4\times3} & 0_{4\times7} \\ 0_{3\times4} & 10^{-1} \cdot I_{3\times3} & 0_{3\times7} \\ 0_{7\times4} & 0_{7\times3} & 10 \cdot I_{7\times7} \end{bmatrix}$$
$$R_n = \begin{bmatrix} 10 \cdot I_{4\times4} \end{bmatrix}$$

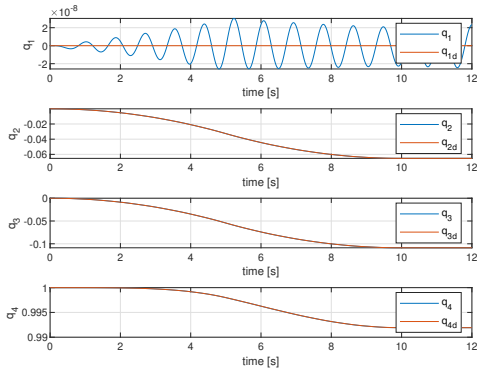
the three different couples of matrices chosen for the analysis are:

$$Q_1 = \begin{bmatrix} 10 \cdot I_{4\times4} & 0_{4\times3} & 0_{4\times7} \\ 0_{3\times4} & 0.5 \cdot I_{3\times3} & 0_{3\times7} \\ 0_{7\times4} & 0_{7\times3} & 10 \cdot I_{7\times7} \end{bmatrix}$$
$$R_1 = \begin{bmatrix} 10 \cdot I_{4\times4} \end{bmatrix}$$

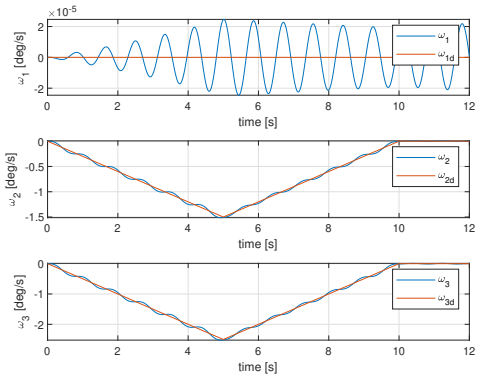
$$Q_2 = \begin{bmatrix} 10 \cdot I_{4\times4} & 0_{4\times3} & 0_{4\times7} \\ 0_{3\times4} & 30 \cdot I_{3\times3} & 0_{3\times7} \\ 0_{7\times4} & 0_{7\times3} & 10 \cdot I_{7\times7} \end{bmatrix}$$
$$R_2 = \begin{bmatrix} 10 \cdot I_{4\times4} \end{bmatrix}$$

$$Q_3 = \begin{bmatrix} 10^4 \cdot I_{4\times4} & 0_{4\times3} & 0_{4\times7} \\ 0_{3\times4} & 10^4 \cdot I_{3\times3} & 0_{3\times7} \\ 0_{7\times4} & 0_{7\times3} & 10 \cdot I_{7\times7} \end{bmatrix}$$
$$R_3 = \begin{bmatrix} 10^{-3} \cdot I_{4\times4} \end{bmatrix}$$

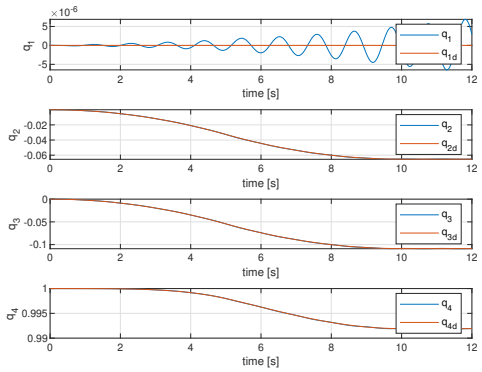
The results are shown in the next page.



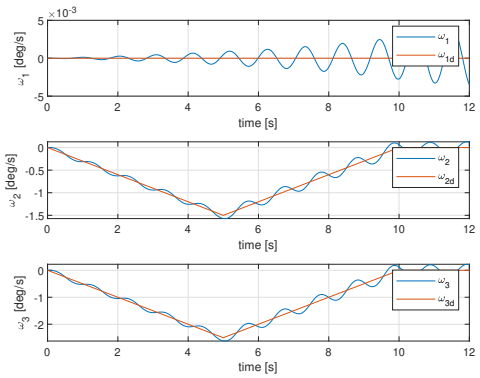
(a) *Quaternion dynamics with* Q_1, R_1



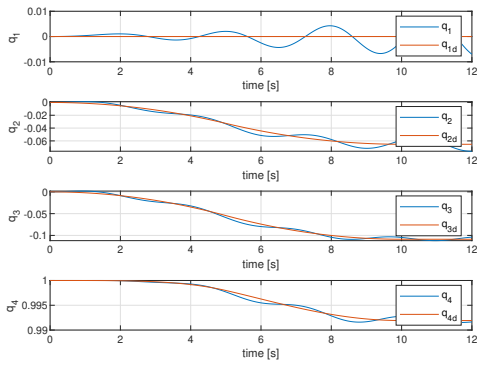
(b) *Angular velocity* ω *with* Q_1, R_1



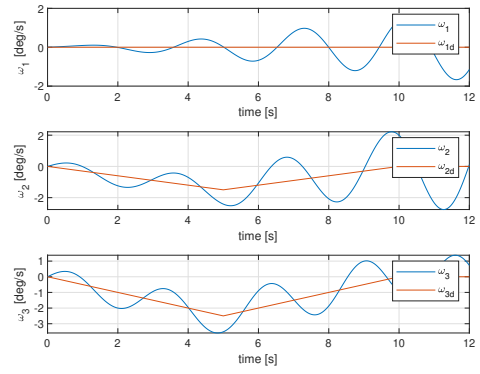
(c) *Quaternion dynamics with* Q_2, R_2



(d) *Angular velocity* ω *with* Q_2, R_2



(e) *Quaternion dynamics with* Q_3, R_3



(f) *Angular velocity* ω *with* Q_3, R_3

Figure 4.9: Simulation results in terms of q and ω considering three different Q and R matrices

Q and R, from the implementation process description, are two arbitrary positive definite matrices needed to calculate P matrix used to minimize the cost function through solving a Linear Matrix Inequality. In Fig.4.9, it is possible to notice a degradation of tracking performance as we move away from the nominal values. In particular, the terms of Q matrix that have been changed in the first two steps are the ones related to the angular velocity ω, while all the terms are different from the nominal ones in the last try and this leads to the said degradation.

Chapter 5

Conclusions

This work proposes an analysis of two controllers applied on a system with a cluster of four Control Moment Gyros in a pyramidal configuration. The first step of the analysis consists in a study of the dynamics of a rigid body in order to develop the equations of spacecraft attitude dynamics. A rigid body in a general state of motion has an angular velocity vector whose direction is that of the instantaneous axis of rotation. Describing the rotational component of a rigid body motion in its three components is necessary to analyze the interaction between the motion and the forces acting on the rigid body. Thus, analyzing the rotational dynamics requires computeing the body's angular momentum and how the mass is distributed throughout the body: this mass distribution is described by the six components of the moment of inertia tensor.

In order to mantain the attitude, an actuator which can generate a large torque, in this case a cluster of CMGs, is required. Whenever using a momentum device for attitude control, it can be useful to write the attitude equations of motion in terms of angular momentum, which states that the time derivative of the angular momentum is equal to the external control torque applied to the body. If using a cluster og CMGs, the momentum exchange between the wheel and the bus leads to a gyroscopic torque in perpendicular direction to both the spin and the gimbal axis. At least three gimbals are necessary for full three-axis control, but a minimum of four CMGs is usually employed for increasing system redundancy. For these reasons, it is useful to write the equation of motion in terms of angular velocity instead of the external control torque, in order to apply the two controllers considering the gimbal angles as control inputs. The case of a cluster of four CMG mounted with the gimbal axis perpendicular to the faces of a pyramid, the sides of which are inclined of a certain skew angle, is the most utilized. Moreover, in order to define the spacecraft reorientation in terms of quaternions, the quaternion kinematic equation is introduced.

Nevertheless, a CMG system leads to the presence of a singularity in a combination of gimbal angles which cannot generate torque in that certain direction. While, maneuvering, the gimbal angles should be able to avoid the singular states in order to generate any commanded torque. The method used in this analysis is the Singularity Robust (SR) method and the variation implemented is the Singular Direction Avoidance (SDA) steering law, which provides the ability to avoid or escape any singularity with a particular finite gimbal rate. Since the controller and the gimbal steering law, which generate a commanded torque and the control torque to escape from or avoid the singularity, respectively, are not designed simultaneously, an error derives from the application of this method and the performance is slightly degradated during the escape from singularity.

Several simulation tests are made to study the trend of the tracking performance of SDA steering law. The results are all described in terms of quaternions and angular velocity and though the desired attitude is time-varying, the SDA controller is able to track it, even changing some dynamic parameters or PD control gains. The results are confirmed with both simulation and experimental tests.

The second controller analyzed is a variation of Classic Model Predictive Control (CMPC). One the reasons of MPC success is due to the ability to provide optimal solution for constrained problems. Moreover, closed-loop stability and feasibility of MPC are required at each time step and this is guaranteed by ensuring that the predicted trajectory satisfies the constraints. Thus, one way to achieve computational efficiency is to reduce the Degree Of Freedom over which the optimization of MPC is performed. On the other hand, improvements are possible using longer horizons but leads to an increment of computational cost.

In this work, a simple algorithm for a variation of CMPC for constrained linear system is proposed. In this method, the integrator state is reset at each sampling time so that the cost function is minimized. Moreover, the optimization process is reduced to an optimization problem with LMI constraints. From the results achieved, it is possible to notice that both closed-loop stability and feasibility are guaranteed and that this method has a certain level of robustness against disturbances, while the introduction of the integrator state reset is an improvement of the tracking control performance.

Appendices

Appendix A

Euler angles and reference frames transformation matrices

In order to specify the orientation of a rigid body relative to an inertial frame, three angles are required. The Euler angles (ψ, θ, ϕ) give the orientation of a rigid, orthogonal xyz frame relative to the XYZ inertial frame. The orthogonal triad of unit vectors parallel to the inertial axis XYZ are $\hat{\boldsymbol{I}}, \hat{\boldsymbol{J}}$ and $\hat{\boldsymbol{K}}$ respectively, while the orthogonal triad of unit vectors oriented to the axis of the xyz frame are $\hat{\boldsymbol{i}}, \hat{\boldsymbol{j}}$ and $\hat{\boldsymbol{k}}$ respectively.

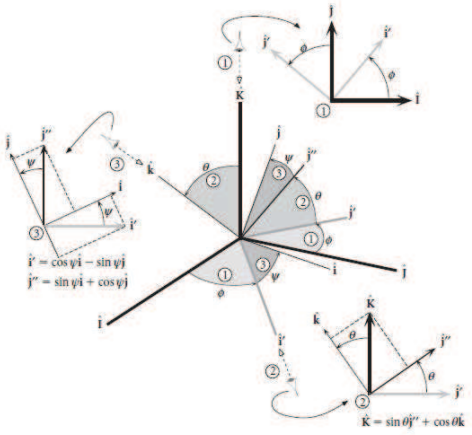


Figure A.1: Euler angles

In Fig. A.1, $\hat{\boldsymbol{I}}\hat{\boldsymbol{J}}\hat{\boldsymbol{K}}$ triad and $\hat{\boldsymbol{i}}\hat{\boldsymbol{j}}\hat{\boldsymbol{k}}$ are shown, along with the three successive rotations required to bring unit vectors initially aligned with $\hat{\boldsymbol{I}}\hat{\boldsymbol{J}}\hat{\boldsymbol{K}}$ into alignment with $\hat{\boldsymbol{i}}\hat{\boldsymbol{j}}\hat{\boldsymbol{k}}$. For simplicity we consider the two frames sharing a common origin.

The xy plane intersects the XY plane along the node line, defined by the unit

vector $\hat{\boldsymbol{i}}$. The first rotation ① is around $\hat{\boldsymbol{K}}$ axis, through the Euler angle ϕ. It rotates the $\hat{\boldsymbol{I}}\hat{\boldsymbol{J}}$ directions to the $\hat{\boldsymbol{i}}'\hat{\boldsymbol{j}}'$ directions. This rotation can be expressed as

$$\hat{\boldsymbol{i}}' = \cos\phi\hat{\boldsymbol{I}} + \sin\phi\hat{\boldsymbol{J}} \tag{A.1}$$

$$\hat{\boldsymbol{j}}' = -\sin\phi\hat{\boldsymbol{I}} + \cos\phi\hat{\boldsymbol{J}} \tag{A.2}$$

$$\hat{\boldsymbol{k}}' = \hat{\boldsymbol{K}} \tag{A.3}$$

$$\begin{Bmatrix} \hat{\boldsymbol{i}}' \\ \hat{\boldsymbol{j}}' \\ \hat{\boldsymbol{k}}' \end{Bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{I}} \\ \hat{\boldsymbol{J}} \\ \hat{\boldsymbol{K}} \end{Bmatrix} \tag{A.4}$$

Thus, the orthogonal transformation matrix associated with this rotation is

$$[\boldsymbol{R}_3(\phi)] = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.5}$$

The second Euler rotation is around the node line $\hat{\boldsymbol{i}}'$, through the angle θ. It rotates the Z axis into alignment with the z axis, and $\hat{\boldsymbol{j}}'$ simultaneously rotates into $\hat{\boldsymbol{j}}''$. We can deduce that

$$\hat{\boldsymbol{i}}'' = \hat{\boldsymbol{i}}' \tag{A.6}$$

$$\hat{\boldsymbol{j}}'' = -\cos\theta\hat{\boldsymbol{j}}' + \sin\theta\hat{\boldsymbol{K}} \tag{A.7}$$

$$\hat{\boldsymbol{k}} = -\sin\theta\hat{\boldsymbol{j}}' + \cos\theta\hat{\boldsymbol{K}} \tag{A.8}$$

$$\begin{Bmatrix} \hat{\boldsymbol{i}}'' \\ \hat{\boldsymbol{j}}'' \\ \hat{\boldsymbol{k}} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{i}}' \\ \hat{\boldsymbol{j}}' \\ \hat{\boldsymbol{K}} \end{Bmatrix} \tag{A.9}$$

The orthogonal transformation matrix is

$$[\boldsymbol{R}_1(\theta)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{A.10}$$

The third Euler rotation is in the xy plane and rotates the unit vectors $\hat{\boldsymbol{i}}'$ and $\hat{\boldsymbol{j}}''$ through the angle ψ around the z axis. The rotation can be expressed as

$$\hat{\boldsymbol{i}} = \cos\psi\hat{\boldsymbol{i}}' + \sin\psi\hat{\boldsymbol{j}}'' \tag{A.11}$$

$$\hat{\boldsymbol{j}} = -\sin\psi\hat{\boldsymbol{i}}' + \cos\psi\hat{\boldsymbol{j}}'' \tag{A.12}$$

$$\hat{\boldsymbol{k}} = \hat{\boldsymbol{k}} \tag{A.13}$$

$$\begin{Bmatrix} \hat{\boldsymbol{i}} \\ \hat{\boldsymbol{j}} \\ \hat{\boldsymbol{k}} \end{Bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{i}}' \\ \hat{\boldsymbol{j}}'' \\ \hat{\boldsymbol{k}} \end{Bmatrix} \tag{A.14}$$

Thus, the orthogonal transformation matrix associated with this rotation is

$$[\boldsymbol{R}_3(\psi)] = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.15}$$

Finally, the transformation matrix $[\boldsymbol{Q}]_{Xx}$ from the inertial XYZ frame into the moving xyz frame is just the product of the three rotation matrices given by Equations A.5, A.10 and A.15

$$[\boldsymbol{Q}_{Xx}] = [\boldsymbol{R}_3(\psi)][\boldsymbol{R}_1(\theta)][\boldsymbol{R}_3(\phi)] \tag{A.16}$$

Substituting the three matrices on the right we can obtain

$$[\boldsymbol{Q}_{Xx}] = \begin{bmatrix} \cos\phi\cos\psi - \sin\phi\sin\psi\cos\theta & \sin\phi\cos\psi + \cos\phi\cos\theta\sin\psi & \sin\theta\sin\psi \\ -\cos\phi\sin\psi - \sin\phi\cos\theta\cos\psi & \sin\theta\cos\psi & \\ \sin\phi\sin\theta & -\cos\phi\sin\theta & \cos\theta \end{bmatrix} \tag{A.17}$$

Appendix B

Simulink implementation of the spacecraft dynamics

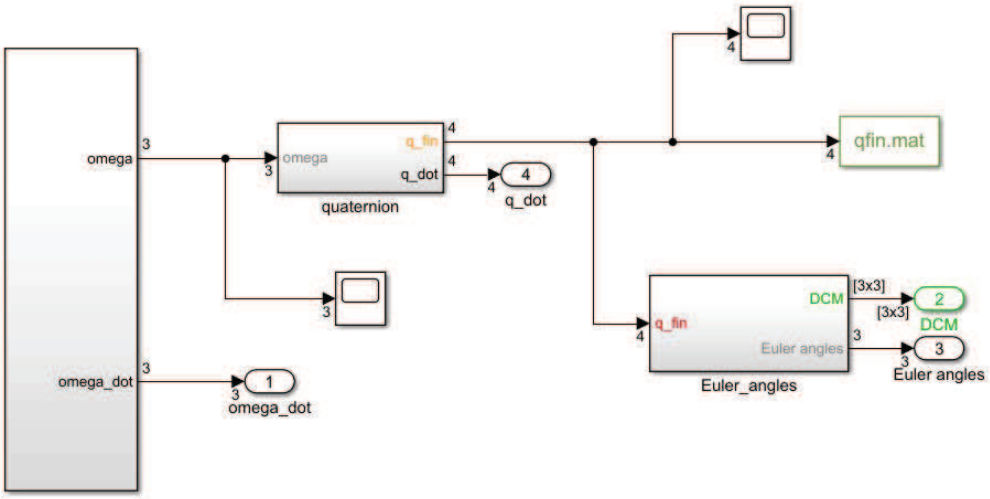


Figure B.1: Nonlinear dynamics implemented on *Simulink*

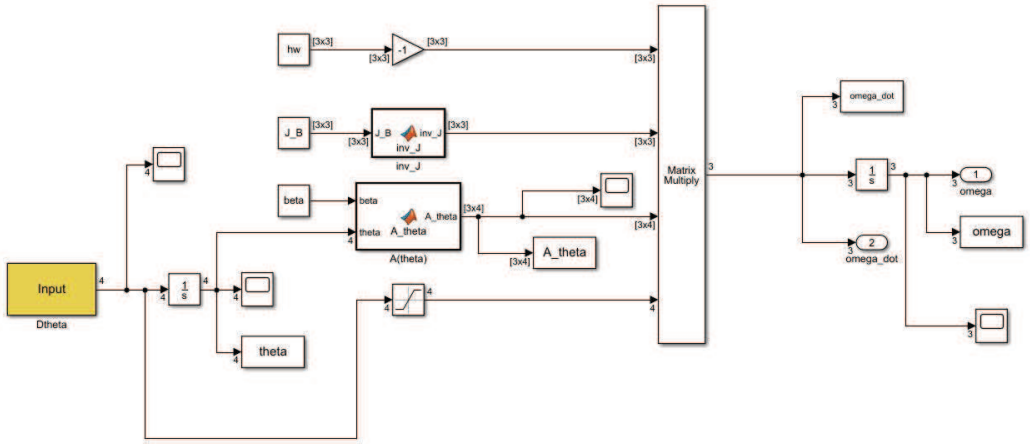


Figure B.2: Particular of ω and θ dynamics implemented on *Simulink*

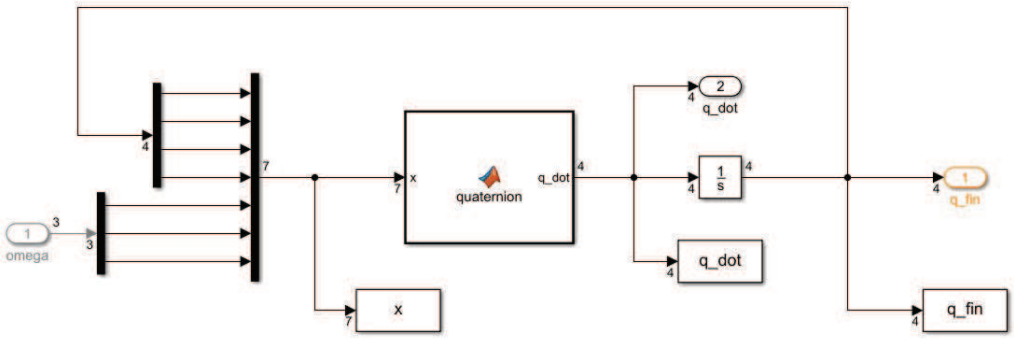


Figure B.3: Particular of q dynamics implemented on *Simulink*

Appendix C

CMG Steering Equation

The equation of motion for a rigid spacecraft with a cluster of N CMGs can be written as

$$\dot{J}\omega + J\dot{\omega} + A_g I_{cg}\ddot{\theta} + A_t I_{ws}[\Omega]^d\dot{\theta} + [\omega^\times](J\omega + A_g I_{cg}\dot{\theta} + A_s I_{ws}\Omega) = g_e \qquad \text{(C.1)}$$

where $\omega = (p, q, r)^T \in \mathbb{R}^3$ is the spacecraft angular velocity vector. As for *Chapter 3*, from this Appendix on the dimensions of vectors and matrices will be clearly explicated without using bold notation. The inertia matrix J of the whole spacecraft is defined as

$$J = {}^B I + A_s I_{cs} A_s^T + A_t I_{ct} A_t^T + A_g I_{cg} A_g^T \qquad \text{(C.2)}$$

where ${}^B I$ is the combined matrix of inertia of the spacecraft platform and the point-masses of the CMGs, while all the vectors and matrices in C.1 are expressed in a body-fixed frame located at the center of rotation of the spacecraft platform. The matrices I_{c*} and I_{w*} are diagonal, with elements the values of the inertias of the gimbal plus wheel structure and wheel-only-structure of the CMGs, respectively. Moreover, the vector $\Omega = (\Omega_1, \ldots, \Omega_N)^T \in \mathbb{R}^N$ represents the wheel speeds of the CMGs with respect to the gimbals while the vector $\theta = (\theta_1, \ldots, \theta_N)^T \in \mathbb{R}^N$ represents the gimbal angles. On each CMG, a frame located at the center of the gimbal/wheel combination having unit vector $\hat{e}_{cj}, \hat{e}_{sj}, \hat{e}_{tj}$, with $j = 1, \ldots, N$, along the gimbal axis, the wheel spin axis and the torque producing axis respectively is attached, so that $\hat{e}_{tj} = \hat{e}_{gj} \times \hat{e}_{sj}$. The matrices $A_* \in \mathbb{R}^{3\times N}$ are used to collect these unit vectors such that $A_* = [e_{*1}, \ldots, e_{*N}]$, with $* = g, s$ or t.

Considering any vector $x = (x_1, x_2, x_3)^T \in \mathbb{R}^3$, the notation $[x^\times]$ denotes the skew-symmetric matrix

$$[x^\times] = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \qquad \text{(C.3)}$$

whereas, for a vector $x \in \mathbb{R}^N$, the notation $[x]^d \in \mathbb{R}^{N\times N}$ denotes the diagonal matrix having as elements the components of the vector x, that is

$$[x^d] = \mathrm{diag}(x_1, \ldots, x_N) \tag{C.4}$$

As a reminder, we have to note that both matrices A_s and A_t are functions of the gimbal angles, so that $A_s = A_s(\theta)$ and $A_t = A_t(\theta)$. Thus, the inertia matrix $J = J(\theta)$ is also a function of the gimbal angles θ, whereas the matrix ${}^B I$ is constant. It is possible to observe that the time variation of the inertia matrix due to the changes of the gimbal angles is reflected in the first term of C.1 and it is generally a small term.

The differential equation that governs the attitude kinematics in terms of the quaternion vector $q = [q_1, q_2, q_3, q_4]^T$ is the one in 2.47, that is

$$\dot{q} = \frac{1}{2}\eta(q)\omega \tag{C.5}$$

Therefore, the term in the right side of C.1 represents the gravity torque owing to the misalignment between the mass center and the center of rotation of the spacecraft simulator platform. Generally, there exists a gravity torque that tends to deteriorate the performance and it has to be compensated by the controller in order to achieve an accurate attitude. This gravity torque acting on the spacecraft can be written as

$$\vec{g}_e = mg\vec{r} \times \vec{n}_0 = -\vec{n}_0 \times mg\vec{r} \tag{C.6}$$

where $\vec{n}_0$ is the inertial unit vector along the local vertical and $\vec{r}$ is the position vector from the center of rotation of the platform to the center of mass. Whether the gravity torque is expressed in the body reference frame it takes the form $g_e = -mg[n_0^\times]r$.

Rearranging equation C.1 by moving all the terms involving the gimbal rates and accelerations onto the right side, it is possible to obtain

$$\begin{aligned} J\dot{\omega} + [\omega^\times]{}^B I\omega + mg[n_0^\times]r = \\ - \dot{J}\omega - A_g I_{cg}\ddot{\theta} - A_t I_{ws}[\Omega]^d\dot{\theta} - [\omega^\times]\big({}^G I(\theta)\omega + A_g I_{cg}\dot{\theta} + A_s I_{ws}\Omega\big) \end{aligned} \tag{C.7}$$

where the inertia term has been split into the constant term ${}^B I$ and a time-varying term ${}^G I(\theta)$ that is

$${}^G I(\theta) = A_s(\theta) I_{cs} A_s^T(\theta) + A_t(\theta) I_{ct} A_t^T(\theta) + A_g I_{cg} A_g^T \tag{C.8}$$

It is now possible to introduce, for any three dimensional vector $v = [v_1, v_2, v_3]^T$, the notation

$${}^B I v = \Gamma_1^T(v)\vartheta_1 \tag{C.9}$$

where

$$\Gamma_1^T(v) = \begin{bmatrix} v_1 & v_2 & v_3 & 0 & 0 & 0 \\ 0 & v_1 & 0 & v_2 & v_3 & 0 \\ 0 & 0 & v_1 & 0 & v_2 & v_3 \end{bmatrix}, \qquad \vartheta_1 = \begin{bmatrix} I_x & I_{xy} & I_{xz} & I_y & I_{yz} & I_z \end{bmatrix}^T$$

Furthermore, by multiplicating

$$[v^\times]^B I v = \Gamma_2^T(v)\vartheta_1 \tag{C.10}$$

we can define

$$\Gamma_2^T(v) = \begin{bmatrix} 0 & -v_1v_3 & v_1v_2 & -v_2v_3 & v_2^2 - v_3^2 & v_2v_3 \\ v_1v_3 & v_2v_3 & v_2^2 - v_1^2 & 0 & -v_1v_2 & -v_1v_3 \\ -v_1v_2 & v_1^2 - v_3^2 & -v_2v_3 & v_1v_2 & v_1v_3 & 0 \end{bmatrix}$$

Lastly, we have

$$mg[v^\times]r = \Gamma_3^T(v)\vartheta_2 \tag{C.11}$$

where $\Gamma_3^T(v) = [v^\times]$ and $\vartheta_2 = mg[r_x, r_y, r_z]^T$.

Therefore, in order to denote all the terms involving the gimbals, we introduce the term u defined as

$$u = -\dot{J}\omega - A_g I_{cg}\ddot{\theta} - A_t I_{ws}[\Omega]^d\dot{\theta} - [\omega^\times]\big({}^G I(\theta)\omega + A_g I_{cg}\dot{\theta} + A_s I_{ws}\Omega\big) \tag{C.12}$$

that can be used to simplify the equation of motion in C.1, using the equations C.9, C.10, C.11, as

$$\dot{\omega} = -J^{-1}\big(\Gamma_2^T(\omega)\vartheta_1 + \Gamma_3^T(\hat{n}_0)\vartheta_2 - u\big) \tag{C.13}$$

If we consider the following dynamic state feedback control law

$$u = -k_1\bar{q} - k_2\omega - \Gamma^T(q,\omega)\hat{\vartheta} - \frac{1}{2}\big({}^G I(\theta)\eta(q)\omega + \dot{J}\bar{q} + \dot{J}\omega\big) \tag{C.14}$$

where $q = [q_1, q_2, q_3, q_4]^T$ and $\bar{q} = [q_1, q_2, q_3]^T$, with the update law

$$\dot{\hat{\theta}} = K\Gamma(q,\omega)(\bar{q} + \omega) \tag{C.15}$$

where

$$\Gamma(q,\omega) = \begin{bmatrix} \Gamma_1(\dot{\bar{q}}) - \Gamma_2(\omega) \\ -\Gamma_3(\hat{n}_0) \end{bmatrix} \tag{C.16}$$

and where $\hat{\vartheta} = [\hat{\vartheta}_1, \hat{\vartheta}_2]$ is the estimate of the unknown vector ϑ, K is a positive definite matrix and k_1, k_2 are positive numbers, the closed loop system of equations C.5, C.12, C.14, is globally asymptotically stable about the equilibrium point

$[\eta_0, \bar{q}, \omega] = 0$, where $\eta_0 = 1 - q_0$ and $\dot{\eta}_0 = -\dot{q}_0$. The proof of this theorem can be found in [34].

Thus, it is possible to obtain the steering equation for a cluster of N CMGs by equating C.12 and C.14, and then arranging the terms with respect to both the gimbal rate $\dot{\theta}$ and the gimbal acceleration $\ddot{\theta}$ as

$$k_1\bar{q} + k_2\omega + \Gamma^T(q,\omega)\hat{\vartheta} + \frac{1}{2}\big({}^G I(\theta)\eta(q)\omega + \dot{J}\bar{q} + \dot{J}\omega\big) = \\ \dot{J}\omega + A_g I_{cg}\ddot{\theta} + A_t I_{ws}[\Omega]^d\dot{\theta} + [\omega^\times]\big({}^G I(\theta)\omega + A_g I_{cg}\dot{\theta} + A_s I_{ws}\Omega\big) \quad \text{(C.17)}$$

Furthermore, after defining the matrices

$$B = A_g I_{cg} \quad \text{(C.18)}$$

$$D\dot{\theta} = A_t I_{ws}[\Omega]^d\dot{\theta} + [\omega^\times]A_g I_{cg}\dot{\theta} + \frac{1}{2}\dot{J}\omega - \frac{1}{2}\dot{J}\bar{q} \quad \text{(C.19)}$$

$$L_r = k_1\bar{q} + k_2\omega + \Gamma^T(q,\omega)\hat{\vartheta} + [\omega^\times]\big({}^G I(\theta)\omega + A_s I_{ws}\Omega\big) + \\ - \frac{1}{2}{}^G I(\theta)\eta(q)\omega \quad \text{(C.20)}$$

where the time derivative of the total inertia matrix is

$$\dot{J} = {}^G\dot{I} = A_t[\dot{\theta}]^d(I_{cs} - I_{ct})A_s^T + A_s[\dot{\theta}]^d(I_{cs} - I_{ct})A_t^T \quad \text{(C.21)}$$

the CMG steering equation becomes

$$B\ddot{\theta} + D\dot{\theta} = L_r \quad \text{(C.22)}$$

Since D matrix can be written as

$$D = A_t I_{ws}[\Omega]^d + [\omega^\times]A_g I_{cg} + \\ \big[(e_{s1}e_{t1}^T + e_{t1}e_{s1}^T)(\omega - \bar{q}) \ldots (e_{sN}e_{tN}^T + e_{tN}e_{sN}^T)(\omega - \bar{q})\big](I_{cs} - I_{ct}) \quad \text{(C.23)}$$

the norm of the matrix $B = A_g I_{cg}$ is very small compared to the norm of matrix D in C.22. For these reason, it is possible to neglect the term $B\ddot{\theta}$ and rewrite the equation C.22 as

$$D\dot{\theta} = L_r \quad \text{(C.24)}$$

Thus, the objective of these CMG steering laws is to solve for $\dot{\theta}$, given any value of L_r.

Appendix D

Experimental tools

Table D.1: CMG components

Tool	Company	Product
Servo motor (Gimbal)	Kondou Kagaku	B3M-SB-1040-A
DC motor (Wheel)	electro kit	41003370
Photo sensor	Akizuki Denshi Tsuushou	TPR-105F
Wheel motor case	Gyroscope.com	CMG Platform

Table D.2: Wheel speed control board components

Tool	Company	Product
Operational amplifier	RS	LM224N
Level shifter	Akizuki Denshi Tsuushou	8-bit conversion module
Schmitt trigger	RS	SN74LS14N
Binary counter	RS	TC74HC4060AP
Motor driver	RS	TB6612FNG (IC)
Motor driver	RS	RE931-04 (SMD conversion)
3-port regulator	RS	TC74HC4060AP
miniUSB	Akizuki Denshi Tsuushou	USB connector
Microcomputer	Renesas	AP-RX62N-0A

Table D.3: Table tools

Tool	Company	Product
PC	Impactics	D3NU2-IR-25-B
9 DOF sensor	YEI technology	3-space sensor
Battery 1	Sanwa Direct	700-BTL011
Battery 2	Anker	Astro E3
Wireless HDMI	Logitec	LDE-WHDI202TR
Wireless Keyboard/Mouse	Microsoft	Wireless Desktop 2000
USB-RS232c converter	Alpha Project	PC-USB-04

Appendix E

Problems involving LMIs

Considering the analysis in [33], given the symmetric matrices $F_i = F_i^T \in \mathbb{R}^{n\times n}$, with $i = 0, \ldots, m$, a Linear Matrix Inequality (LMI) has the form

$$F(x) \triangleq F_0 + \sum_{i=1}^{m} x_i F_i > 0 \tag{E.1}$$

where $x \in \mathbb{R}^m$. The inequality symbol in E.1 means that F_x is positive definite, i.e. $u^T F(x) u > 0$ for all nonzero $u \in \mathbb{R}^n$. This LMI is equivalent to a set of n polynomial inequalities in x.
Moreover, it is possible to find *nonstrict* LMIs, with the form

$$F(x) \geq 0 \tag{E.2}$$

Strict LMIs in E.1 and nonstrict LMIs in E.2 are closely related, though only strict LMIs are considered in the following analysis.

The LMI in E.1 is a convex constraint on x and the set $\{x|F(x) > 0\}$ is convex indeed. Furthermore, it can represent a wide variety of convex constraints on x. In particular, linear inequalities, convex quadratic inequalities, matrix norm inequalities and constraints that arise in control theory, such as Lyapunov and convex quadratic matrix inequalities, can all be cast in the form of an LMI.
Multiple LMIs such as $F^{(1)}(x) > 0, \ldots, F^{(p)}(x) > 0$ can be expressed as the single LMI $\mathbf{diag}(F^{(1)}(x), \ldots, F^{(p)}(x)) > 0$. Thus, no distinction will be made between a set of LMIs and a single LMI.

Whether the matrices F_i are diagonal, the LMI $F(x) > 0$ is just a set of linear inequalities. Nonlinear convex inequalities are represented as an LMI using Schur complements. In particular, the LMI

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{bmatrix} > 0 \tag{E.3}$$

where $Q(x) = Q(x)^T$, $R(x) = R(x)^T$ and $S(x)$ depend affinely on x, is equivalent to

$$R(x) > 0, \qquad Q(x) - S(x)R(x)^{-1}S(x)^T > 0 \tag{E.4}$$

As we can see, the set of nonlinear inequalities can be represented as an LMI.

In the following example, the matrix norm constraint $\|Z(x)\| < 1$, where $Z(x) \in \mathbb{R}^{p\times q}$ and depends affinely on x, is represented as an LMI

$$\begin{bmatrix} I & Z(x) \\ Z(x)^T & I \end{bmatrix} > 0 \tag{E.5}$$

since $\|Z\| < 1$ is equivalent to $I - ZZ^T > 0$. The case $q = 1$ reduces to a general convex quadratic inequality on x.

The constraint $c(x)^T P(x)^{-1} c(x) < 1$, $P(x) > 0$, where $c(x) \in \mathbb{R}^n$ and $P(x) = P(x)^T \in \mathbb{R}^{n\times n}$ depend affinely on x, is expressed as an LMI

$$\begin{bmatrix} P(x) & c(x) \\ c(x)^T & 1 \end{bmatrix} > 0 \tag{E.6}$$

Generally, the constraint $\mathbf{Tr} S(x)^T P(x)^{-1} S(x) < 1$, with $P(x) > 0$, where it is $P(x) = P(x)^T \in \mathbb{R}^{n\times n}$ and $S(x) \in \mathbb{R}^{n\times p}$ which depend affinely on x, is handled by introducing a new matrix variable $X = X^T \in \mathbb{R}^{p\times p}$, and the LMI is

$$\mathbf{Tr} X < 1, \qquad \begin{bmatrix} X & S(x)^T \\ S(x) & P(x) \end{bmatrix} > 0 \tag{E.7}$$

It is often possible to deal with problems in which the variables are matrices as follows

$$A^T P + PA < 0 \tag{E.8}$$

where $A \in \mathbb{R}^{n\times n}$ is given and $P = P^T$ is the variable. We can put the Lyapunov inequality in E.8 in the form of E.1 considering $P_1, \ldots, P_m$ as a basis for symmetric $n \times n$ matrices ($m = n(n+1)/2$) and then taking $F_0 = 0$ and $F_i = -A^T P_i - P_i A$. Furthermore, leaving LMIs in a condensed form as E.8, in addition to saving notation, may lead to more efficient computation.

Consider the quadratic matrix inequality

$$A^T P + PA + PBR^{-1}B^T P + Q < 0 \tag{E.9}$$

where A, B, $Q = Q^T$, $R = R^T > 0$ are given matrices of appropriate sizes and $P = P^T$ is the variable. It can be expressed as the Linear Matrix Inequality

$$\begin{bmatrix} -A^T P - PA - Q & PB \\ B^T P & R \end{bmatrix} > 0 \tag{E.10}$$

This representation shows that the quadratic matrix inequality in E.10 is convex in P.

In some problems it is possible to find linear equality constraints on the variables such as

$$P > 0, \qquad A^T P + PA < 0, \qquad \mathbf{Tr} P = 1 \tag{E.11}$$

where $P \in \mathbb{R}^{k \times k}$ is the variable. In order to write E.11 in the form $F(x) > 0$ we can eliminate the equality constraint considering $P_1, \ldots, P_m$ as a basis for symmetric $k \times k$ matrices with trace zero ($m = (k(k+1)/2) - 1$) and P_0 as a symmetric $k \times k$ matrix with $\mathbf{Tr} P_0 = 1$. Thus, lastly, $F_0 = \mathbf{diag}(P_0, -A^T P_0 - P_0 A)$ and also $F_i = \mathbf{diag}(P_i, -A^T P_i - P_i A)$, for $i = 1, \ldots, m$, should be used in the analysis.

E.1 Standard LMI and EVP problems

Given an LMI $F(x) > 0$, the corresponding LMI problem is to find x^f, with $f = feasible$ such that $F(x^f) > 0$ or determine that the LMI is infeasible. This is a convex feasibility problem and corresponds to find a nonzero $G \geq 0$ such that $\mathbf{Tr} G F_i = 0$, for $i = 1, \ldots, m$ and $\mathbf{Tr} G F_0 \leq 0$.

As an example, consider the given $A_i \in \mathbb{R}^{n \times n}$, with $i = 1, \ldots, L$. We need to find P that satisfies the LMI

$$P > 0, \qquad A_i^T P + PA_i < 0, \qquad i = 1, \ldots, L \tag{E.12}$$

or determine that no P exists. Whether no such P exists, the problem corresponds to finding $Q_0, \ldots, Q_L$, not all zero, such that

$$Q_0 \geq 0, \ldots, Q_L \geq 0, \qquad Q_0 = \sum_{i=1}^{L} (Q_i A_i^T + A_i Q_i) \tag{E.13}$$

which is a nonstrict LMI.

Furthermore, the Eigenvalue Problem (EVP) is to minimize the maximum eigenvalue of a matrix that depends affinely on a variable, subject to an LMI constraint, as follows

$$\begin{array}{ll} \text{minimize} & \lambda \\ \text{subject to} & \lambda I - A(x) > 0, \qquad B(x) > 0 \end{array}$$

where A and B are symmetric matrices that depend on the optimization of the variable x, conducting the analysis on a convex optimization problem.
This kind of problems can be written in the equivalent forms of minimizing a linear function subject to an LMI as

$$\begin{array}{llll} \text{minimize} & c^T x & \text{minimize} & \lambda \\ \text{subject to} & F(x) > 0 & \text{subject to} & A(x,\lambda) > 0 \end{array}$$

where F is an affine function of x and A is affine in (x, λ). Whether the case of the matrices F_i being all diagonal, the problem reduces to the general linear problem of minimizing the linear function $c^T x$ subject to a set of linear inequalities on x.

In addition of the theory explained, an example is shown below. Consider the problem

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \begin{bmatrix} -A^T P - PA - C^T C & PB \\ B^T P & \gamma I \end{bmatrix} > 0, \quad P > 0 \end{array}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$ and $C \in \mathbb{R}^{m\times n}$ are given, and P and γ are the optimization variables. This EVP can also be expressed using an associated quadratic matrix inequality as follows

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & A^T P + PA + C^T C + \gamma^{-1} PBB^T P < 0, \quad P > 0 \end{array}$$

Appendix F

Theorem demonstration for LMI systems

The equation 4.33 implies that the inequality

$$\|\xi(l+i+1|k)\|_P^2 - \|\xi(k+i|k)\|_P^2 \leq -\|\xi(k+i+1|k)\|_Q^2 - \|v(k+i|k)\|_R^2 \quad \text{(F.1)}$$

holds. Therefore, from this inequality and from $\xi(k+H_s|k) \in \mathcal{E}(P,0,\bar{\eta}_2)$, the relations $\xi(k+i|k) \in \mathcal{E}(P,0,\bar{\eta}_2)$, $\forall i \geq H_s$ hold. As a result, since $\mathbb{V}(\xi(k+i|k)) = \xi(k+i|k)^T P \xi(k+i|k)$ is a positive definite function, the conclusion is that $\mathbb{V}(\xi(k+i|k))$ and consequently $\xi(k+i|k)$ converge to zero as i tends to infinity.

Thus, by summing the inequalities

$$\|\xi(k+i+1|k)\|_P^2 - \|\xi(k+i|k)\|_P^2 \leq -\|\xi(k+i+1|k)\|_Q^2 - \|v(k+i|k)\|_R^2 \quad \text{(F.2)}$$

$\forall i \geq H_s$, and using $\|\xi(\infty|k)\|_P \geq 0$, it is possible to obtain

$$J_2(k) < \|\xi(k+H_s|k)\|_P^2 \quad \text{(F.3)}$$

Furthermore, from $\xi(k+i|k) \in \mathcal{E}(P,0,\bar{\eta}_2)$, $\forall i \geq H_s$, the following relation holds

$$\|\xi(k+H_s|k)\|_P^2 \leq \bar{\eta}_2 \quad \text{(F.4)}$$

By performing a congruence transformation to 4.34 by block-diag$[S,I]^{-1}$, and applying the Schur complement to the resulting inequality, we obtain

$$\left(\frac{1}{\bar{\eta}_2}\right) P - \left(\frac{1}{\rho^{(l)2}}\right) F^{(l)T} F^{(l)} \geq 0 \quad \text{(F.5)}$$

Hence, from $\xi(k+i|k) \in \mathcal{E}(P,0,\bar{\eta}_2)$, $\forall i \geq 0$, the relation

$$|v^{(l)}(k+i|k)| \leq \rho^{(l)}, \quad \forall l = 1,\ldots,m, \quad \forall i \geq H_s \quad \text{(F.6)}$$

holds. Moreover, from the analysis of the two relations

$$\rho^{(l)} + \Gamma^{(l)}\bar{w} \leq \rho^{(l)} + |\Gamma^{(l)}\bar{w}| \leq \bar{u}^{(l)} \tag{F.7}$$

$$-\rho^{(l)} + \Gamma^{(l)}\bar{w} \geq -\rho^{(l)} - |\Gamma^{(l)}\bar{w} \geq -\bar{u}^{(l)} \tag{F.8}$$

the inequality $|u^{(l)}(k+i|k)| \leq u^{(l)}$, $\forall i \geq H_s$, $\forall l = 1, \ldots, m$ holds.

Moreover, as we can see from 4.35, the following relation holds

$$\begin{bmatrix} \tau S & * & * \\ 0 & \theta_x^{(j)2} - \tau\bar{\eta}_2 & * \\ \Psi_x^{(j)}(AS + BY) & \Psi_x^{(j)}\Pi w(k|k) & 1 \end{bmatrix} \geq 0, \quad \forall j = 1, \ldots, n_x \tag{F.9}$$

By performing a congruence transformation to F.9, by block-diag$[S, 1, 1]^{-1}$, and applying the Schur complement to the resulting inequality it is possible to obtain

$$\begin{bmatrix} \tau S^{-1} & * \\ 0 & \theta_x^{(j)2} - \tau\bar{\eta}_2 \end{bmatrix} - \begin{bmatrix} \tilde{A}^T\Psi_x^{(j)T} \\ w(k|k)^T\Pi^T\Psi_x^{(j)T} \end{bmatrix} \begin{bmatrix} \Psi_x^{(j)}\tilde{A} & \Psi_x^{(j)}\Pi w(k|k) \end{bmatrix} \geq 0 \tag{F.10}$$

$\forall j = 1, \ldots, n_x$, where $\tilde{A} = A + BF$. Moreover, by multiplying F.10 from the left by $[\xi(k+i|k)^T, 1]$ and from the right by $[\xi(k+i|k)^T, 1]^T$ we obtain

$$\begin{aligned} (\Psi_x^{(j)}x(k+i+1|k))^T\Psi_x^{(j)}x(k+i+1|k) - \theta_x^{(j)2} \leq \\ \tau[\xi(k+i|k)^T S^{-1}\xi(k+i|k) - \bar{\eta}_2] \end{aligned} \tag{F.11}$$

$\forall j = 1, \ldots, n_x$. Thus, this equation implies that

$$\Psi_x x(k+i+1|k) \leq \theta_x, \quad \forall i \geq H_s \tag{F.12}$$

holds.

Bibliography

[1] G. Avanzini, "Spacecraft attitude dynamics and control."

[2] R. E. Kalman, *Contributions to the theory of optimal control.* Bol.Sol.Mat.Mexicana 5, 1960.

[3] D. Bernardini and A. Bemporad, "Stabilizing model predictive control of stochastic constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1468–1480, June 2012.

[4] L. Blackmore, M. Ono, and A. Bektassov, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, June 2010.

[5] A. Mesbah, S. Streif, R. Findeisen, and R. Braatz, "Stochastic nonlinear model predictive control with probabilistic constraints," *American Control Conference*, pp. 2413–2419, June 2014.

[6] Q. Cheng, M. Cannon, B. Kouvaritakis, and M. Evans, "Stochastic MPC for systems with both multiplicative and additive disturbances," *19th IFAC World Congress, Cape Town, SA*, 2014.

[7] M. Mammarella, E. Capello, M. Lorenzen, F. Dabbene, and F. Allgöwer, "A general sampling-based smpc approach to spacecraft proximity operations."

[8] 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*, June 2016.

[9] M. Nagai, S. Yamanaka, and Y. Hirano, "Integrated control of active rear wheel steering and yaw moment control using braking forces," *Transactions of the Japan Society of Mechanical Engineers C*, vol. 64 (622), pp. 2132–2139, 1998.

[10] M. Kale and A. Chipperfield, "Stabilized MPC formulation for robust reconfigurable flight control," *Control Engineering Practice*, vol. 13, pp. 771–788, 2005.

[11] K. Åström and B. Wittenmark, *Computer-controlled systems: theory and design.* Courier Corporation, June 2013.

[12] Y. Yang, "Singularity-free model predictive spacecraft attitude regulation using variable-speed control moment gyroscope model," *IEEE Transactions on*

Aerospace and Electronic Systems, vol. 54, no. 3, pp. 1511–1518, June 2018.

[13] P. Wu, H. Wen, T. Chen, and D. Jin, "Model predictive control of rigid spacecraft with two variable speed control moment gyroscope," *Applied Mathematics and Mechanics*, vol. 38, no. 11, pp. 1551–1564, November 2017.

[14] F. Quagliotti, "Atmospheric flight mechanics."

[15] N. S. Bedrossian, J. Paradiso, E. V. Bergmann, and D. Rowell, "Redundant Single Gimbal Control Moment Gyroscope Singularity Analysis," *Journal of Guidance, Control and Dynamics*, vol. 13, no. 6, pp. 1096–1099, 1990.

[16] H. Kurokawa, "Constrained Steering Law of Pyramid-type Control Moment Gyros and Ground Test," *Journal of Guidance, Control and Dynamics*, vol. 20, no. 3, pp. 445–449, 1997.

[17] B. Wie, "New Singularity Escape/Avoidance Steering Logic for Control Moment Gyro Systems," *AIAA Guidance, Navigation and Control Conference, Austin, Texas*, 2003.

[18] H. Yoon and P. Tsiotras, "Singularity Analysis and Avoidance of Variable Speed Control Moment Gyros - Part I: No Power Constraint Case," *AIAA Guidance, Navigation and Control Conference, Providence, Rhode Island*, 2004.

[19] B. Wie, "Singularity Analysis and Visualization for Single-Gimbal Control Moment Gyro System," *Journal of Guidance, Control and Dynamics*, vol. 27, no. 2, pp. 271–282, 2004.

[20] H. Yoon and P. Tsiotras, "Singularity Analysis and Avoidance of Variable Speed Control Moment Gyros - Part II: Power Constraint Case," *AIAA Guidance, Navigation and Control Conference, Providence, Rhode Island*, 2004.

[21] K. A. Ford and C. D. Hall, "Singular Direction Avoidance Steering for Control-Moment Gyros," *Journal of Guidance, Control and Dynamics*, vol. 23, no. 4, pp. 648–656, 2000.

[22] H. S. Oh and S. R. Vadali, "Feedback Control and Steering Laws for Spacecraft Using Single Gimbal Control Moment Gyros," *Journal of the Astronautical Sciences*, vol. 39, no. 2, pp. 183–203, 1991.

[23] D. Jung and P. Tsiotras, "An Experimental Comparison of CMG Steering Control Laws," *Georgia Institute of Technology, Atlanta.*

[24] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions with Singularity Robustness for Robot Manipulator Control," *Journal of Dynamic Systems, Measurement and Control*, vol. 108, no. 3, pp. 163–171, 1986.

[25] A. Schwarm and M. Nikolaou, "Chance constrained model predictive control," *AIChE J.*, vol. 45, no. 8, pp. 1743–1752, 1999.

[26] D. Van Hessem and O. Bosgra, "Closed-loop stochastic model predictive control in a receding horizon implementation on a continuous polymerization reactor example," *Proceedings of the 2004 American Control Conference, Boston*, pp. 914–919, 2004.

[27] F. Herzog, S. Keel, G. Dondi, L. Schumann, and H. Geering, "Model predictive control for portfolio selection," *Proceedings of the 2006 American Control Conference, Minneapolis*, pp. 1252–1259, 2006.

[28] J. A. Primbs, "Dynamic hedging of basket options under proportional transaction costs using receding horizon control," *Int. J. Control*, vol. 82, no. 10, pp. 1841–1855, 2009.

[29] P. Couchman, B. Kouvaritakis, M. Cannon, and F. Prashad, "Gaming strategy for electric power with random demand," *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1283–1292, 2005.

[30] P. Patrinos, S. Trimboli, and A. Bemporad, "Stochastic MPC for real-time market-based optimal power dispatch," *Proceedings of the 50th Conference on Decision and Control, Orlando*, pp. 7111–7116, 2011.

[31] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari, "Use of model predictive control and weather forecasts for energy efficient building climate control," *Energy Build.*, vol. 45, pp. 15–27, 2012.

[32] J. Yan and R. R. Bitmead, "Incorporating state estimation into model predictive control and its application to network traffic control," *Automatica*, vol. 41, no. 4, pp. 595–604, 2005.

[33] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, "Linear Matrix Inequalities in System and Control Theory," *SIAM, Philadelphia, PA*, 1994.

[34] S. Di Gennaro, "Adaptive robust stabilization of rigid spacecraft in presence of disturbances," *Proceedings of the 34th Conference on Decision and Control, New Orleans, LA*, vol. 2, pp. 1147–1152, 1995.