POLITECNICO DI TORINO
Repository ISTITUZIONALE

# Design and validation of an MPC controller for CMG-based testbed

*Original*
Design and validation of an MPC controller for CMG-based testbed / Facchino, Matteo; Totsuka, Atsushi; Capello, Elisa; Satoh, Satoshi; Guglieri, Giorgio; Yamada, Katsuhiko. - In: OPTIMIZATION AND ENGINEERING. - ISSN 1389-4420. - 24:(2023), pp. 185-221. [10.1007/s11081-021-09633-z]

*Availability:*
This version is available at: 11583/2900532 since: 2021-05-14T12:35:40Z

*Publisher:*
Springer

*Published*
DOI:10.1007/s11081-021-09633-z

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

22 November 2024

Optimization and Engineering (2023) 24:185–221
https://doi.org/10.1007/s11081-021-09633-z

RESEARCH ARTICLE

# Design and validation of an MPC controller for CMG-based testbed

**Matteo Facchino[1] · Atsushi Totsuka[3] · Elisa Capello[2] · Satoshi Satoh[3] · Giorgio Guglieri[1] · Katsuhiko Yamada[3]**

Received: 31 July 2020 / Revised: 29 November 2020 / Accepted: 16 April 2021 / Published online: 13 May 2021
© The Author(s) 2021

**Abstract**
In the last years, Control Moment Gyros (CMGs) are widely used for high-speed attitude control, since they are able to generate larger torque compared to "classical" actuation systems, such as Reaction Wheels . This paper describes the attitude control problem of a spacecraft, using a Model Predictive Control method. The features of the considered linear MPC are: (i) a virtual reference, to guarantee input constraints satisfaction, and (ii) an integrator state as a servo compensator, to reduce the steady-state error. Moreover, the real-time implementability is investigated using an experimental testbed with four CMGs in pyramidal configuration, where the capability of attitude control and the optimization solver for embedded systems are focused on. The effectiveness and the performance of the control system are shown in both simulations and experiments.

**Keywords** MPC Controller · Real-time Implementability · Control moment gyro · Experimental tests on attitude control

## 1 Introduction

The growing interest in space exploration is focusing the research in the development of space architectures able to be independent and self-sustainable. Nowadays, space missions for satellites require the capability to rapidly change the attitude with high slew rate, both to ensure the success of the mission and to avoid catastrophic impacts among crafts (Yadlin 2011; Courie et al. 2018). In general the attitude

✉ Elisa Capello
elisa.capello@polito.it

[1] Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, Italy

[2] Department of Mechanical and Aerospace Engineering, Politecnico di Torino, CNR-IEIIT, Torino, Italy

[3] Graduate School of Engineering, Osaka University, Osaka, Japan

control can be defined as the process aimed to stabilize and adjust the orientation of spacecraft during space maneuvers (National Academies of Sciences and Medicine 2016). Precise attitude tracking control of small satellites is a demanding task due to the limited hardware resources of sensors, actuators, and processors equipped on-board. During the years, attitude control systems based on Control Moment Gyros (CMGs) have been used for a large scale of satellites as they can generate a large torque compared to Reaction Wheels (RWs) systems. 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 the system is equipped with. Various efforts have been made to overcome this troublesome problem, and many CMG singularity avoidance methods have been developed, with different advantages and disadvantages (Kraft 1993; Wie 2005; Lee et al. 2007; Chakravorty 2012).

A configuration with CMGs and steering scheme for rapid maneuvering of Earth observation satellites is presented in paper (Yadlin 2011). The CMG singularity arises when the torque vectors of all the CMGs are aligned on a plane, so that the CMG system cannot produce a control torque in the normal direction of that plane. Different configurations of CMGs systems are compared in Miceli (2007), using a single PID controller with a steering law based on the Moore-Penrose inverse matrix to avoid the singularity condition. The analysis is particularly interesting for the comparison among different actuator system: the first two are CMG systems in pyramidal configuration with a different skew angles, the third system is a cluster of two CMGs and four RWs.

Even if Proportional Integral Derivative (PID)-based controllers are largely used in industrial applications, they are used in cases where the plant has a simple dynamics and where an appropriate proportional and damping gains are able to ensure a stable behavior of the control system. These controllers, combined with a Singular Direction Avoidance (SDA) steering law (Ford and Hall 2000), are usually designed for CMG-based space systems, as in Xin et al. (2015); Bedrossian et al. (2005), Ciavola et al. (2019). However, if the system to be controlled is more complex, the implementation of an optimal control strategy could be better. The optimal control solution is easy and simple to design, and the feedback gains can be computed by solving a Riccati equation. The most successful methodology among the optimal control environment is the Model Predictive Control (MPC).

As clear from literature (Wang and Boyd 2009; Mayne 2014; Mammarella et al. 2018), however, one of the main drawback of the MPC control scheme is related to the online optimization problem and to the difficulty of embedding a real-time solver for the on-board implementation. A practical solution, usually used for MPC implementation on embedded systems, is the offline evaluation of the control law, uploaded on-board by means of lookup tables Bemporad et al. (2002). In literature, gain scheduling design are usually proposed for linear time-varying (LTV) system, but for spacecraft control using CMGs there is a limitation due to the high number of independent time-varying variables. So, gain scheduling design can be unfeasible. Moreover, in space applications, to handle uncertainties and fast changes on the dynamics, this solution is usually not adopted. For this reason, the real-time

implementation (translated in a real-time solver) is a key aspect for the on-board implementation of an MPC control system.

MPC was first applied successfully for plants with slow dynamics (Qin and Badgwell 2003; Yu-Geng et al. 2013). However, in the last years, it was widely applied to different areas, as space applications. Indeed, only few control methodologies are able to address challenging tasks, required by space observations and missions, as deeply explained in Eren et al. (2017). The main advantages of using MPC controllers are related to two main aspects: (1) the MPC controller can directly calculate the angular velocity of the gimbals without calculating the external control torque, (2) input, state and hardware constraints can be included in the implementation. The first aspect enables us to avoid using the conventional steering laws with the singularity problem to drive the CMG system. Besides, the second aspect guarantees computational feasibility, secure operation, and the best possible performance of the system. As a consequence, the stability of the closed-loop system and its feasibility are guaranteed, while even the tracking performance is improved under the constraints.

In Guiggiani et al. (2015) a Model Predictive Controller for spacecraft attitude tracking with RWs actuators is proposed and the controller is designed for desaturation of the RWs, without fuel consumption, compensating the gravity gradient torques. In paper Ashok et al. (2016) the problem of attitude control of spacecraft using CMGs in pyramidal configuration is faced with the implementation of a method to avoid the singularity condition by means of Nonlinear MPC (NMPC). A singularity avoidance term is introduced in the cost function to be minimized, in order to generate a control input that moves away from the singularity when the CMG system approaches the neighbourhood of the singularity, and to rapidly converge to a desired attitude when the CMG system stays away from the singularity. Even if the implementation of an NMPC is treated, this paper still uses a steering law to calculate the value of the gimbal angular velocities. When a steering law is included in the closed-loop scheme, the controller generates the input and the steering law generates the gimbal rates value, avoiding the singularity condition, separately. So, it should happen that the generated torque differs from the controller torque, compromising the control performance in avoiding singularity. This problem can be resolved by generating directly from the controller the control input that does not achieve singularity.

In light of these circumstances, this paper proposes an MPC based attitude controller for the CMG system, which directly generates the gimbal rates so as to avoid the singularity problem possibly caused by the steering laws. The proposed method is based on an MPC method with additional features in Wada and Tsurushima (2016), which are virtual reference state and integrator state. Although the literature provides a linear MPC controller, it optimizes not only a sequence of control inputs, but also a virtual reference signal and an integrator state. These additional optimizations improve attitude tracking performance and reduce the steady-state error with the guarantee of recursive feasibility and constraints satisfaction. The further contribution of this paper is to verify the real-time implementability of the proposed MPC controller using an experimental testbed with four CMGs in pyramidal configuration. By floating the experimental equipment via spherical air bearing and by

matching the center of rotation and the center of gravity of the equipment, the testbed simulates spacecraft attitude dynamics. Since ground experiments of spacecraft attitude control are very difficult, and furthermore MPC controllers require implementation of real-time optimization, there have been a few references, which conduct experimental verification of MPC based spacecraft attitude control. We demonstrate both simulation and experimental results of the proposed method, and also show an optimization solver, which is easy to be translated and computationally efficient for on-board implementation.

The paper is organized as follows. In Sect. 2 the dynamics of the analyzed system is considered, focusing on attitude dynamics. The testbed system is described in Sect. 2.2. The MPC controller is introduced in Sect. 3, including the virtual reference and the integrator state. The real-time implementability of the proposed MPC controller is also discussed in Sect. 3.1. The Simulation results are described in Sect. 4 and the experimental test in Sect. 5. Finally, conclusions are drawn in Sect. 6.

## 2 Preliminaries

Some preliminaries on Linear Matrix Inequalities (LMIs) and on MPC control system are first introduced. The LMI approach is used for the definition of the controller gains and the Yalmip Matlab toolbox Lofberg (2004) is considered.

Given the symmetric matrices $F_i = F_i^T \in \mathbb{R}^{n\times n}$, with $i = 0, \dots, m$, a LMI problem has the form

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

where $x \in \mathbb{R}^m$. The inequality symbol in Eq. (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

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

The LMI problem in Eq. (1) is a convex constraint on $x$ and the set $\{x|F(x) > 0\}$ is convex indeed. Multiple LMIs such as $F^{(1)}(x) > 0, \dots, F^{(p)}(x) > 0$ can be expressed as the single LMI $\mathbf{diag}(F^{(1)}(x), \dots, F^{(p)}(x)) > 0$, where **diag** denotes a diagonal matrix composed by the elements.

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 problem using Schur complements.

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

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

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

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

$$A^T P + PA < 0 \tag{5}$$

where $A \in \mathbb{R}^{n\times n}$ is given and $P = P^T$ is the variable. We can put the Lyapunov inequality in Eq. (5) in the form of Eq. (1) considering $P_1, \dots, P_m$ 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 LMI inequalities in a condensed form as Eq. (5) may lead to more efficient computation.

Consider the quadratic matrix inequality

$$A^T P + PA + PBR^{-1}B^T P + Q < 0, \tag{6}$$

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 LMI problem

$$\begin{bmatrix} -A^T P - PA - Q & PB \\ B^T P & R \end{bmatrix} > 0 \tag{7}$$

This representation shows that the quadratic matrix inequality in Eq. (7) is convex in $P$.

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

$$P > 0, \quad A^T P + PA < 0, \quad \mathrm{Tr}\, P = 1 \tag{8}$$

where $P \in \mathbb{R}^{k\times k}$ is the variable. In order to write Eq. (8) in the form $F(x) > 0$ we can eliminate the equality constraint considering $P_1, \dots, P_m$ 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, $F_0 = \mathrm{diag}\,(P_0, -A^T P_0 - P_0 A)$ and $F_i = \mathrm{diag}\,(P_i, -A^T P_i - P_i A)$, for $i = 1, \dots, m$, should be used in the analysis. The LMIs defined in this paper are solved with the Matlab Toolbox Yalmip Lofberg (2004).

## 3 Model definition

As briefly introduced before, the main objective of this paper is the precise attitude control of a testbed, equipped with four Control Moment Gyros (CMGs) in pyramidal configuration. The testbed is considered as a rigid body, in which Euler's equation and quaternion kinematics are included Markley and Crassidis (2014). After the definition of the dynamic and kinematic equations of the testbed, the Linear Time-Invariant (LTI) mathematical model is derived in a state space formulation.

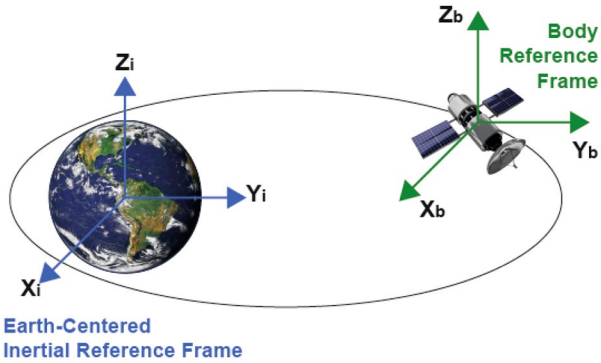


**Fig. 1** Earth-Centered Inertial Reference Frame and Body Reference Frame

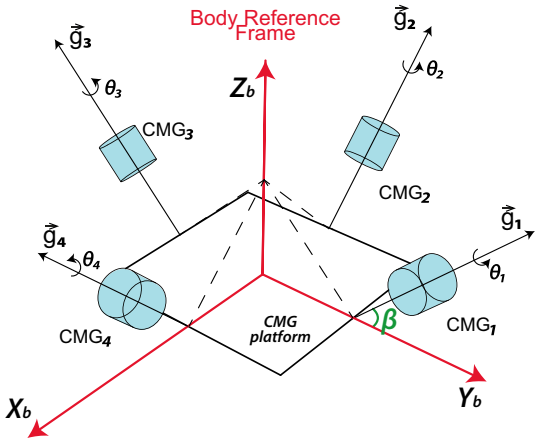


**Fig. 2** Configuration of a pyramidal testbed equipped with 4 CMGs

### 3.1 Attitude dynamics modeling

For the study of spacecraft dynamics, two reference frames are usually considered: an Inertial Reference Frame and a Body Reference Frame, as in Fig. 1.

The Inertial Reference Frame considered in this study is the Earth-Centered Inertial (ECI) that is also called Inertial Geocentric Reference Frame (Fig. 1). It is commonly used to study the motion of a body orbiting around Earth and it has its origin in the Center-of-Mass (CoM) of Earth.

The Body Reference Frame (Fig. 2) has been adopted to describe the attitude dynamics once introduced the CMGs to the equations of motion. The origin of the Body Reference Frame is in the CoM of the spacecraft and it is rigidly attached to the body.

Taking into account the previous assumption, the Euler's equation of motion can be written as Srinivasan et al. (2014)

$$M_B = \dot{h}_B + \omega_B \times h_B, \tag{9}$$

where the subscript $_B$ indicates variables in the Body Reference Frame, $\omega_B \in \mathbb{R}^3$ is the angular velocity of the spacecraft, $h_B \in \mathbb{R}^3$ is the angular momentum, $M_B \in \mathbb{R}^3$ is the torque vector defined as

$$M_B = \sum_{i=1}^{n} r_i \times F_i.$$

$F_i$ is the force which has a momentum with respect to the CoM, and $r_i$ is the distance between the CoM and the considered force. For a generic body, the inertia tensor J directly relates the angular velocity with the angular momentum. The angular momentum $h$ of a rigid body, referred to its CoM, can be defined

$$h = \mathrm{J}\omega. \tag{10}$$

Considering the testbed system, Eq. (10) can be rewritten as

$$h_B = \mathrm{J}_B \omega_B + h_B^C. \tag{11}$$

Thus, the total angular momentum is the sum of two terms: (1) the angular momentum of the spacecraft, and (2) $h_B^C$ is the term given by CMG system, which for a pyramidal configuration is

$$h_B^C = h_\omega \sum_{i=4}^{4} \overline{h_i}(\theta_i), \tag{12}$$

with

$$\overline{h_1}(\theta_1) = \begin{bmatrix} -\sin(\theta_1)\cos\beta \\ \cos(\theta_1) \\ \sin(\theta_1)\sin\beta \end{bmatrix}, \quad \overline{h_2}(\theta_2) = \begin{bmatrix} -\cos(\theta_2) \\ -\sin(\theta_2)\cos\beta \\ \sin(\theta_2)\sin\beta \end{bmatrix}, \tag{13}$$

$$\overline{h_3}(\theta_3) = \begin{bmatrix} \sin(\theta_3)\cos\beta \\ -\cos(\theta_3) \\ \sin(\theta_3)\sin\beta \end{bmatrix}, \quad \overline{h_4}(\theta_4) = \begin{bmatrix} \cos(\theta_4) \\ \sin(\theta_4)\cos\beta \\ \sin(\theta_4)\sin\beta \end{bmatrix}, \tag{14}$$

where $\theta_i \in \mathbb{R}^4$ is the gimbal angle, for $i = 1, ..., 4.$, and $h_\omega = \mathrm{J}_\omega \omega_\omega \in \mathbb{R}$ denotes the angular momentum of each wheel. Since we suppose four identical wheels rotating at the same constant speed, $h_\omega$ is a constant scalar.

Combining the previous equations, the Euler's equation here considered is

$$M_B = \omega_B \times (\mathrm{J}_B \omega_B + h_B^C) + \mathrm{J}_B \dot{\omega}_B + \dot{h}_B^C. \tag{15}$$

For a cluster of 4 CMGs, the internal angular momentum vector $h_B^C$ is a non linear function of the gimbal angles $\theta_i$, with $i = 1, ..., 4$, of each CMG, as in Eq. (12). The time derivative of Eq. (12) is

$$\dot{h}_B^C = h_\omega A(\theta)\dot{\theta}, \tag{16}$$

where the matrix $A(\theta) \in \mathbb{R}^{3\times 4}$, considering a fixed skew angle $\beta$, is

$$A(\theta) = \begin{bmatrix} -\cos(\theta_1)\cos\beta & \sin(\theta_2) & \cos(\theta_3)\cos\beta & -\sin(\theta_4) \\ -\sin(\theta_1) & -\cos(\theta_2)\cos\beta & \sin(\theta_3) & \cos(\theta_4)\cos\beta \\ \cos(\theta_1)\sin\beta & \cos(\theta_2)\sin\beta & \cos(\theta_3)\sin\beta & \cos(\theta_4)\sin\beta \end{bmatrix}. \tag{17}$$

The columns of matrix $A(\theta)$ express the contribution of each CMG to $\dot{h}_B^C$. Finally, the following assumptions are considered

1. The initial angular momentum is zero, and consequently $\omega_B(0) = 0$
2. The initial torque applied to the body is zero, $M_B(0) = 0$

With these hypotheses, starting from Eq. (15) and substituting Eq. (16), we can easily obtain the following simplified equation for the angular acceleration

$$\dot{\omega}_B = -h_\omega \mathrm{J}_B^{-1} A(\theta)\dot{\theta}. \tag{18}$$

This latter equation relates the angular acceleration of the testbed to the angular rate of the gimbals $\dot{\theta}$, which is the control input vector in this system. For the definition of the model, the subscript that indicates the Body Reference Frame is omitted for simplicity.

The kinematic equation of the system using a quaternion representation is

$$\dot{q} = \frac{1}{2}\Sigma(q)\omega, \tag{19}$$

with $q = [q_v^T, q_4]^T = [q_1, q_2, q_3, q_4]^T \in \mathbb{R}^4$, where $q_v = [q_1, q_2, q_3]^T \in \mathbb{R}^3$ is the vectorial part of the quaternion, and $q_4$ is the scalar term.

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

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

Combining Eqs. (18) and (19) obtained previously, the following mathematical model can be obtained

$$\dot{x}_p = \begin{bmatrix} \dot{q} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\Sigma(q)\omega \\ -h_\omega \mathrm{J}^{-1} A(\theta)\dot{\theta} \end{bmatrix}, \tag{20}$$

where the system state vector is defined $x_p = \begin{bmatrix} q \\ \omega \end{bmatrix} \in \mathbb{R}^{n_p}$, with $n_p = 7$.

Rearranging the previous equation, the state equation of the system is

$$\dot{x}_p = \begin{bmatrix} 0_{4\times4} & \frac{1}{2}\Sigma(q) \\ 0_{3\times4} & 0_{3\times3} \end{bmatrix} x_p + \begin{bmatrix} 0_{4\times4} \\ -h_\omega \mathrm{J}^{-1} A(\theta) \end{bmatrix} \dot{\theta} = \bar{A}_p(x_p)x_p + \bar{B}_p(\theta)\dot{\theta} \tag{21}$$

The state of the system is $x_p \in \mathbb{R}^7$ and the control input is $\dot{\theta} \in \mathbb{R}^4$, that is the angular rates of the gimbals. The matrix $\bar{B}_p$ can be considered to be a (external) parameter-dependent matrix as $\bar{B}(\theta)$.

A linearisation is performed assuming that the initial gimbal angles are $\theta_0 = [0, 0, 0, 0]^T$. This fixed value is used to evaluate the matrix $A(\theta)$, so the matrix $\bar{B}_p$ is constant. The desired attitude in terms of $\omega_d$ and $q_d = [q_{1,d}, q_{2,d}, q_{3,d}, q_{4,d}]^T$ is considered as equilibrium point, to have a constant state matrix. We linearize $\bar{A}_p$ matrix around the terminal value of the desired attitude. Thus, more precisely, the constant matrix $\bar{A}_p(q_d(T))$ is used, where "$T$" denotes the terminal time.

So, we have

$$\dot{x}_p = \begin{bmatrix} 0_{4\times4} & \frac{1}{2}\Sigma(q_d(T)) \\ 0_{3\times4} & 0_{3\times3} \end{bmatrix} x_p + \begin{bmatrix} 0_{4\times4} \\ -h_\omega \mathrm{J}^{-1} A(\theta_0) \end{bmatrix} \dot{\theta}, \tag{22}$$

with the following output equation

$$y = \begin{bmatrix} I_3 & 0_{3\times4} \\ 0_{3\times4} & I_3 \end{bmatrix} x_p = \bar{C}_p x_p, \tag{23}$$

where $y = [q_v^T, \omega^T]^T \in \mathbb{R}^6$ and $I_3 \in \mathbb{R}^{3\times3}$ is the identity matrix.

### 3.2 Testbed description

The testbed considered in this work is the one developed in Higashiyama et al. (2020), which is depicted in Fig. 3. Physical parameters of the testbed is summarized in Table 1. The testbed simulates a spacecraft using a metal plate equipped with the four CMGs in pyramidal configuration, a PC, two microcomputers, and three sets of counter weights. The metal plate is floated by compressed air via spherical air bearing, and the center of rotation and the center of gravity of the testbed is matched by manually adjusting the counter weights. The attitude and the angular velocity are estimated via a triaxial gyroscope and accelerometer called the 3-Space Sensor made by YEI Technology, and a low-pass filter. Moreover, the nominal moment of inertia is estimated by applying the least-square method to a regressor representation of the Euler's equations of the uncontrolled rotational motion.

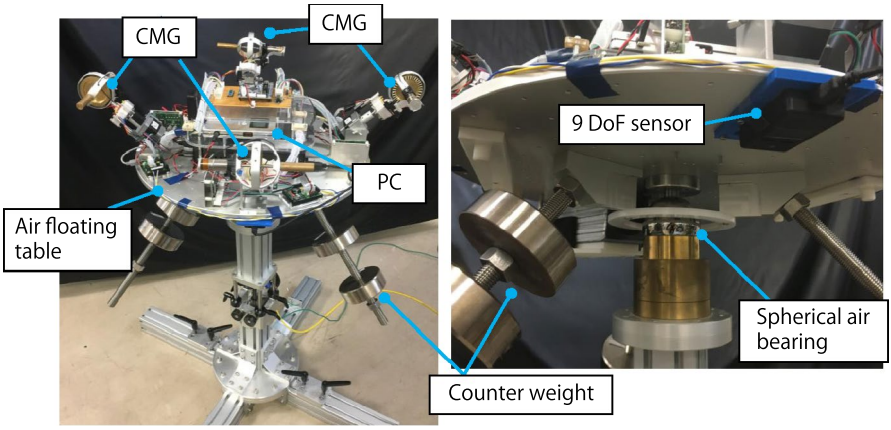


**Fig. 3** Testbed with 4 CMGs in pyramidal configuration

**Table 1** Physical parameters of the testbed

| Variable | Nomenclature | Numerical value | Unit |
|---|---|---|---|
| Mass of the testbed | $m$ | 21.2 | kg |
| Inertia tensor of the testbed | $J$ | $\begin{bmatrix} 0.464 & 0.0 & 0.0 \\ 0.0 & 0.534 & 0.0 \\ 0.0 & 0.0 & 0.610 \end{bmatrix}$ | kg m$^2$ |
| Skew angle | $\beta$ | 45 | deg |
| Wheel angular momentum | $h_w$ | 0.0576 | Nms |

## 4 Design of an MPC controller

Starting from the research of Wada and Tsurushima (2016), a linear MPC with integrator as a servo compensator is considered to control the attitude of a spacecraft/testbed. The integrator is included to reduce the steady-state error. From the optimization procedure, instead of only the control input, a virtual reference signal and an integrator state are also evaluated. The virtual reference signal $w$ is a variable that substitutes the original reference signal $r$ in the control algorithm in order to guarantee input constraints satisfaction. The controller has to decide at each sample time if the integrator state $x_c$ must be optimized or not. If $x_c$ is not optimized, it evolves according to the integrator equations in order to guarantee the steady-state error elimination. Instead, if $x_c$ is included in the optimization problem, the optimized value of this variable gives a rapid decrease of the cost function value.

The plant system is described in the following state space formulation

$$x_p(k+i+1|k) = A_p x_p(k+i|k) + B_p u(k+i|k) \tag{24}$$

$$y(k+i|k) = C_p x_p(k+i|k) + D_p u(k+i|k), \tag{25}$$

where $x_p \in \mathbb{R}^{n_p}$ is the state of the plant, $u \in \mathbb{R}^m$ is the control input and $y \in \mathbb{R}^{n_y}$ is the system output. In our case the matrix $D_p$ is null. In the MPC control system, the matrices of Eq. 21 are translated in discrete time as follows $A_p := I_d + \bar{A}_p dt$, and $B_p := \bar{B}_p dt$. $\bar{C}_p = C_p$. Moreover, following the literature, $(\cdot)(m|k)$ means a value at time m calculated at time k.

In our MPC formulation, both input and state constraints are considered as follows:

$$\Psi_u u(k+i|k) \leq \theta_u, \quad \forall i \geq 0 \tag{26}$$

$$\Psi_{x_p} x_p(k+i+1|k) \leq \theta_x, \quad \forall i \geq 0, \tag{27}$$

where the matrices $\Psi_u \in \mathbb{R}^{2m\times m}$, $\theta_u \in \mathbb{R}^{2m}$, $\theta_x \in \mathbb{R}^{2n_p}$ and $\Psi_{x_p} \in \mathbb{R}^{2n_p\times n_p}$.

Finally, the integrator equations are

$$x_c(k+i+1|k) = x_c(k+i|k) + e(k+i|k) \tag{28}$$

$$e(k+i|k) = w(k|k) - y(k+i|k), \tag{29}$$

with $e \in \mathbb{R}^{n_y}$.

Combining the equations of the plant and the integrator, the error dynamics is defined as

$$\xi(k+i+1|k) = A\xi(k+i|k) + B\nu(k+i|k) \tag{30}$$

$$e(k+i|k) = C\xi(k+i|k), \tag{31}$$

where

$$\xi(k+i|k) = x(k+i|k) - \Pi w(k|k) \tag{32}$$

$$\nu(k+i|k) = u(k+i|k) - \Gamma w(k|k) \tag{33}$$

and $x = [x_p^T, x_c^T]^T$, $C = [-C_p, 0]$,

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

The matrices $\Pi$ and $\Gamma$ are calculated for the tracking control problem in case of linear systems, and to guarantee input and state constraints satisfaction. The calculation procedure and the proofs are presented in Wada and Tsurushima (2016).

In this paper, the following cost function is considered and implemented

$$J(k) = \sum_{i=0}^{H_s-1} \{||\xi(k+i+1|k)||_Q^2 + ||v(k+i|k)||_R^2\} \\ + ||\xi(k+H_s|k)||_P^2 + ||\Pi(w(k|k) - r(k|k))||_M^2, \tag{34}$$

where $H_s$ denotes the prediction horizon, and for a generic vector $x$ and a square matrix $A$ of appropriate dimension, $||.||_A$ denotes the weighted Euclidean norm, namely, $||x||_A := (x^T A x)^{1/2}$. The matrices $Q$, $R$, $M$, and $P$ in (34) are positive-definite. The first three matrices are design matrices, while the matrix $P$ has to be obtained solving an LMI problem for the terminal controller of the dual-mode MPC guaranteeing recursive feasibility and stability, as described in Wada and Tsurushima (2016).

The novelty in the defined cost function is found out in the last term, $||\Pi(w(k|k) - r(k|k))||_M^2$ to include the virtual reference signal $w$ into the optimization procedure. The virtual reference changes from the original reference $r$ in order to satisfy the constraints and to obtain better control performance. The last term prevents $w$ from being unnecessarily away from $r$, and also guarantees that the deviation between $w$ and $r$ remains finite. Under the regulator condition relating to the existence of the matrices $\Pi$ and $\Gamma$, the convergence of $w$ to a step reference $r$ is shown in Wada and Tsurushima (2016).

Using the prediction matrices $S$ and $T$, the state prediction of the error system up to the horizon $H_s$ is defined as

$$\widetilde{\xi} = T\xi(k|k) + S\widetilde{v}, \tag{35}$$

where

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

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

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

$$\widetilde{\Psi}_u(\widetilde{v} + \widetilde{\Gamma} w(k|k)) \le \widetilde{\theta}_u, \tag{36}$$

where $\widetilde{\Psi}_u = \mathbf{diag}[\Psi_u, \cdots, \Psi_u]$, $\widetilde{\theta}_u = [\theta_u^T, \dots, \theta_u^T]^T$ and $\widetilde{\Gamma} = [\Gamma^T, \dots, \Gamma^T]^T$.

The state constraints are rewritten as

$$\widetilde{\Psi}_x(T\xi(k|k) + S\widetilde{v} + \widetilde{\Pi} w(k|k)) \le \widetilde{\theta}_x. \tag{37}$$

$\widetilde{\Psi}_x = \mathbf{diag}[\Psi_x, \cdots, \Psi_x]$, $\widetilde{\theta}_x = [\theta_x^T, \dots, \theta_x^T]^T$ and $\widetilde{\Pi} = [\Pi^T, \dots, \Pi^T]^T$.

As already said, in the considered MPC, the integrator state can be or not optimized. In order to decide if $x_c$ has to be optimized or not, a threshold $\eta_s$ must be

defined. The integrator state $x_c$ is reset at each sampling time so that the cost function value $J(k)$ is minimized while $J(k)$ is greater then $\eta_s$. The control algorithm with integral action is used to achieve zero steady-state error when $J(k)$ is less than or equal to $\eta_s$.

The value of $\eta_s$ is determined by trial and error procedure. If $\eta_s = 0$, the robustness against the disturbances cannot be assured. However, if $\eta_s$ is too large, the robustness against disturbances is improved but the tracking performance are not good. So, the parameter $\eta_s$ has to be chosen by considering a trade-off between tracking performance and robustness against disturbances.

### 4.1 Real-time implementability

For the implementation of the MPC control system, the required resources are correlated with the problem size (i.e., number of constraints, prediction horizon) and, at the same time, the efficiency of the solver algorithm for the optimization of the cost function. In addition to these two computational indices, on-board hardware capabilities and software architectures should also be considered. Implementation issues are discussed in this Section due to hardware constraints and limited computational power of the embedded controller. Some simplifications of the code and analysis of the solvers are required. Moreover, since the GNC hardware is tested on Matlab/Simulink environment, the selection criteria for the solver includes compatibility with this environment. In this paper, in implementing the proposed controller to the testbed, we equivalently convert the optimization problem formulated as an LMI optimization problem in Wada and Tsurushima (2016) into a quadratic programming problem, which has much less computational burden, as

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

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

Moreover, the cost function is re-written using Matlab notation, to simplify the implementations on this environment. So, we have

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

where $\phi(k)$ is a vector composed by the measured state of the system and by the reference signals.

The tested solvers are

- *quadprog*, the interior-point-convex algorithm provided by the Matlab Optimization Toolbox to solve quadratic programming problem,
- *fmincon*, Matlab function that can implement four different algorithms: interior-point, Sequential Quadratic Programming (SQP), active set, and thrust-region-reflective. Also this solver deals quadratic programming problems.

- *quadwright*, a quadratic programming solver developed by J. Currie presented in Currie (2014); Wright (1996), able to speed up the computational capabilities for embedded applications.

All these solvers are tested in simulations. The software *quadprog* can use a thrust-region-reflective method for handling problems with bounds and linear equality constraints. However, it showed an high computational effort and cannot be translated in C code. The solver *fmincon* is usually used for nonlinear constrained problems and it finds a constrained minimum of a scalar function of several variables starting at an initial estimate. This solution is not a global minimum, but it is computational efficient. Finally, the *quadwright* provides the smallest computational times, because it has been developed with a focus on efficient memory use and high speed convergence. For the experimental verification, we create a C++ code with *Eigen* and *eiquadprog* libraries. The *Eigen* library is a C++ template library for linear algebra, and the *eiquadprog* Guennebaud et al. (2011) library is a C++ routine working with Eigen data structures for solving quadratic programming problems based on the Goldfarb-Idnani active-set dual method.

## 5 Simulation results

The simulations are performed in Matlab, with an Intel Core i7 at 2.4GHz and 8GB RAM. The selected sampling time is 25 ms and the prediction horizon $H_s$ is 5 steps. The MPC solver, as previously said, is *quadwright*. The controller parameters are defined in Table 2.

The reference signals for the quaternion are evaluated from the evolution of the angular velocity, starting from Eq. (19), for all different simulation cases. Furthermore, in all the simulations the initial conditions are null states $x$, except for $x_4(0) = q_4 = 1$ and null gimbal angles $\theta$. So, the initial conditions are the ideal quaternion and null angular velocities. Two set of simulations are considered: (1) in the first set of simulations, all the three angular velocities are changed. Trapezoidal variations of the angular velocities are simulated, (2) in the second set of simulations, triangular variations of an angular velocity is examined.

**Table 2** MPC Controller Parameters

| Variable | Nomenclature | Numerical value | Units |
|---|---|---|---|
| Input constraint (gimbal motors speed max value) | $\overline{\dot{\theta}}$ | 1 | rad/s |
| Angular velocity constraint of the testbed | $\overline{\omega}$ | 25 | deg/s |
| Acceleration constraints on the CMG motors | $\overline{\ddot{\theta}}$ | 2 | rad/s$^2$ |
| Prediction horizon | $H_s$ | 5 | – |

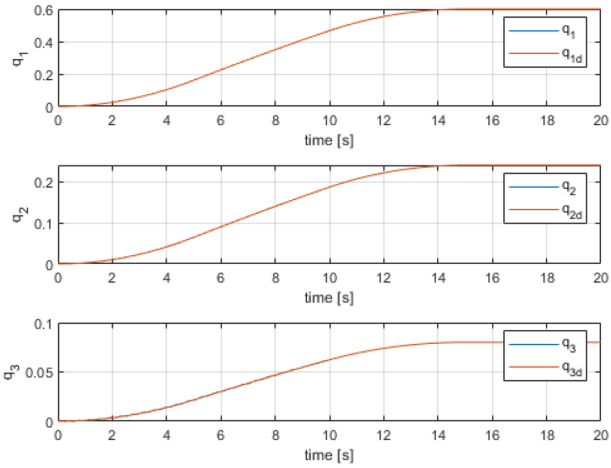


**Fig. 4** Quaternion vectorial components for case 1 with classical MPC

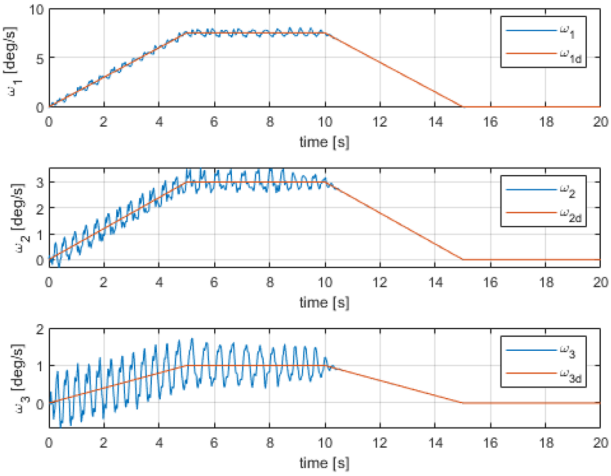


**Fig. 5** Angular velocity components for case 1 with classical MPC

## 5.1 First set of simulations

All the simulations are performed for both of the linear classical MPC and the MPC with optimization of virtual reference signal and integrator state in order to compare their control performance. Here, all the three angular velocities are changed. A simulation time of 20 seconds is considered. The desired angular velocities in the first simulation are $\omega_d = [7, 3, 1]^T$ deg/s. The desired quaternion are evaluated from Eq. (19). Comparing Figures 5, 6 and 7, it is clear that MPC with optimization of integrator state and virtual reference has better performance with respect to the classical MPC, even if all the angular velocities are changed. Moreover, a

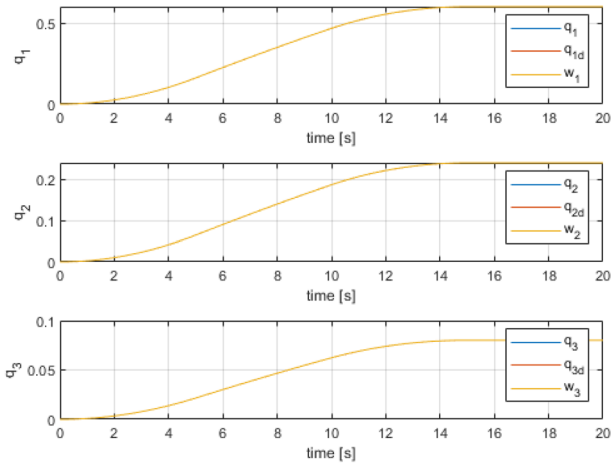


**Fig. 6** Quaternion vectorial components for case 1 with MPC with optimization of virtual reference signal and integrator state

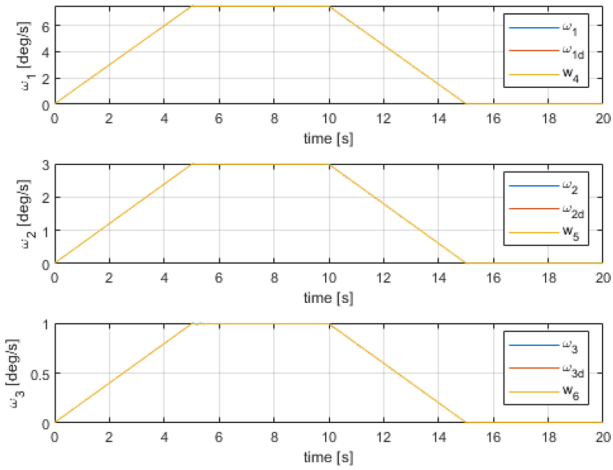


**Fig. 7** Angular velocity components for case 1 with MPC with optimization of virtual reference signal and integrator state

noisy input is obtained by the classical MPC as in Figure 8. The control input of the MPC with integrator state and virtual reference is shown in Figure 9. In a similar way, noisy variation of the gimbal angles can be observed in Figure 10, whereas no noisy response is observed in Figure 11. Instead, no difference can be observed in the quaternion behavior, as in Figures 4, 5 and 6. Finally, to show the effectiveness of the proposed controller, a singularity index is evaluated, which is $\det(AA^T)$, as in Higashiyama et al. (2020). Highest is the singularity index, more effective is the singularity avoidance. The singularity indices are in Figures 12 and 13. As in the

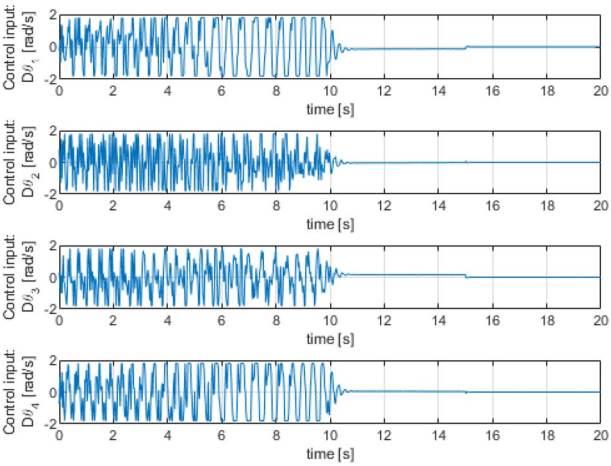


**Fig. 8** Control input of case 1 with classical MPC

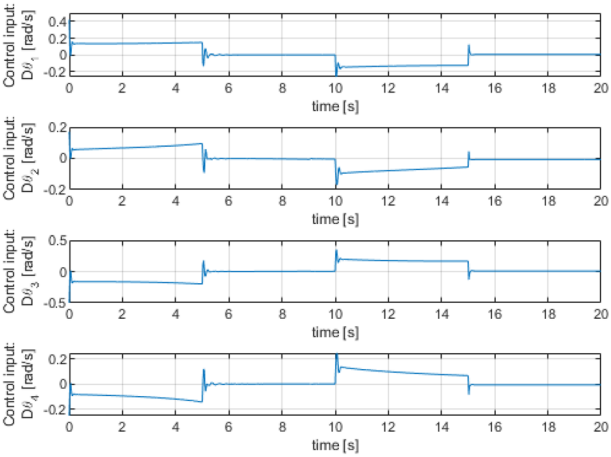


**Fig. 9** Control input of case 1 with MPC with optimization of virtual reference signal and integrator state

previous results, a noisy behavior is observed in Figure 12. Moreover, a small value is obtained if no additional optimizations are included.

In the second simulation, as before, a step variation of the three angular velocities is considered with two variations of the angular velocities. A simulation time of 10 seconds is considered: (i) $\omega_d = [2, 2, 2]^T$ deg/s between 2 and 4 seconds, (ii) $\omega_d = [4, 4, 4]^T$ deg/s between 5.5 and 8 seconds. As for the previous case, the state variables (angular velocities and quaternion) are analyzed for the classical MPC and the one with optimization of virtual reference signal and integrator state. Moreover, control input and gimbal angles are also considered. All the results

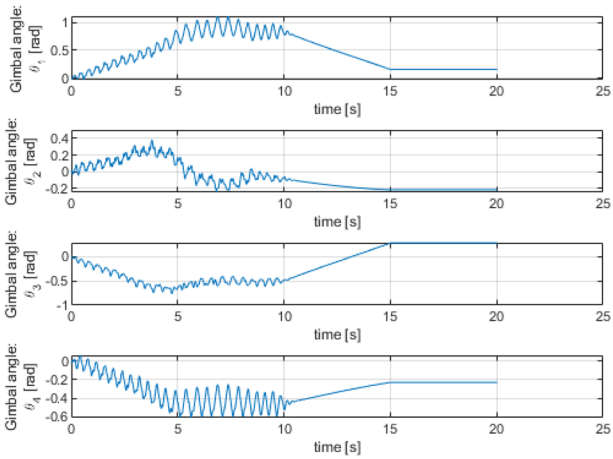


**Fig. 10** Gimbal angles of case 1 with classical MPC

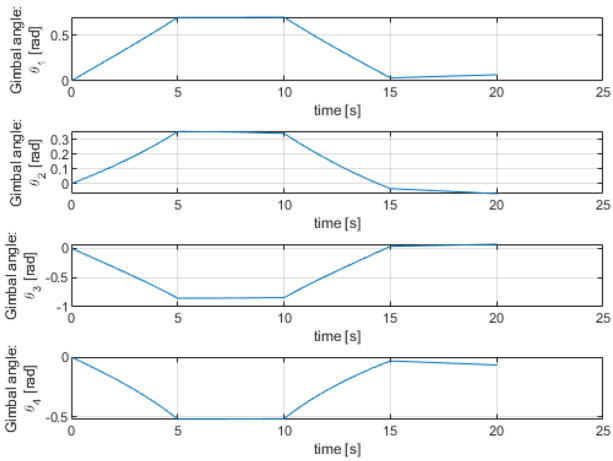


**Fig. 11** Gimbal angles of case 1 with MPC with optimization of virtual reference signal and integrator state

are in Figs. 14,15, 16, 17, 18, 19, 20 21. Finally, the singularity index is analyzed to show the effectiveness of the proposed MPC controller. As in Figs. 22 and 23, even in this simulation the singularity index of the MPC with optimization of virtual reference signal and integrator state is greater than the classical MPC with no additional optimizations.

These simulations demonstrate the effectiveness of the MPC controller with additional features, that optimize the virtual reference signal and the integrator state.

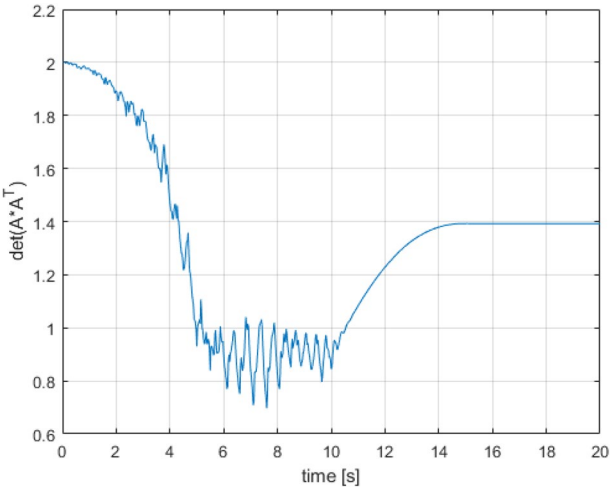


**Fig. 12** Singularity index of case 1 with classical MPC

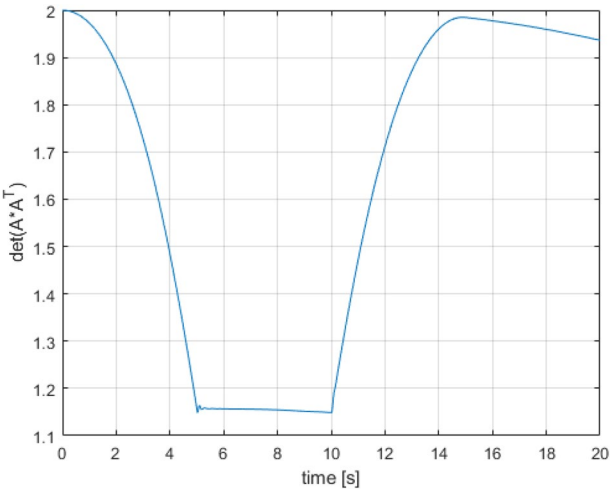


**Fig. 13** Singularity index of case 1 with MPC with optimization of virtual reference signal and integrator state

## 5.2 Second set of simulations

In this set of simulations, triangular variation of one angular velocity is proposed. This set of simulations is performed in simulation and by experimental tests as well (as detailed in Sect. 5). As in the previous section we show that the MPC with integrator state and virtual reference has better performance than the classical one. Thus,

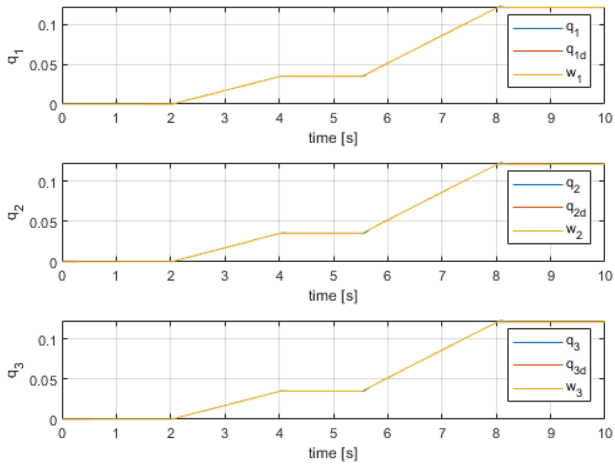


**Fig. 14** Quaternion vectorial components for case 2 with MPC with optimization of virtual reference signal and integrator state

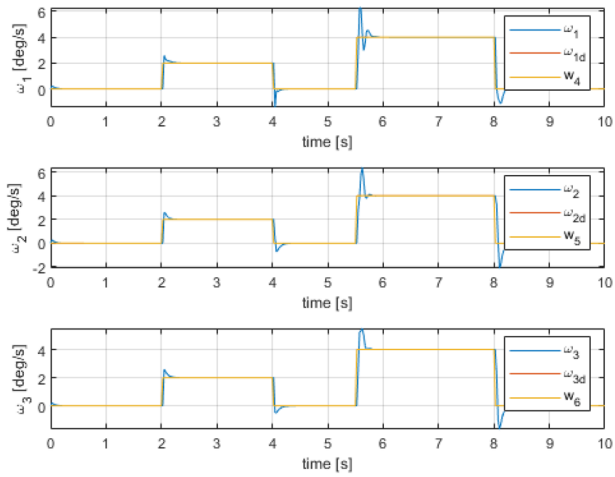


**Fig. 15** Angular velocity components for case 2 with MPC with optimization of virtual reference signal and integrator state

in the following simulations we further investigate the performance of the MPC with integrator state and virtual reference.

As before, two cases are analyzed: (1) a rotation around the $X$-axis, namely the direction of $[1, 0, 0]^T$ of $\omega_d = [5, 0, 0]^T$ deg/s, and (2) a rotation around $Z$ axis of $\omega_d = [0, 0, 5]^T$ deg/s. In both cases a triangular shape of the desired value is considered. Moreover, the desired quaternion is evaluated from Eq. (19).

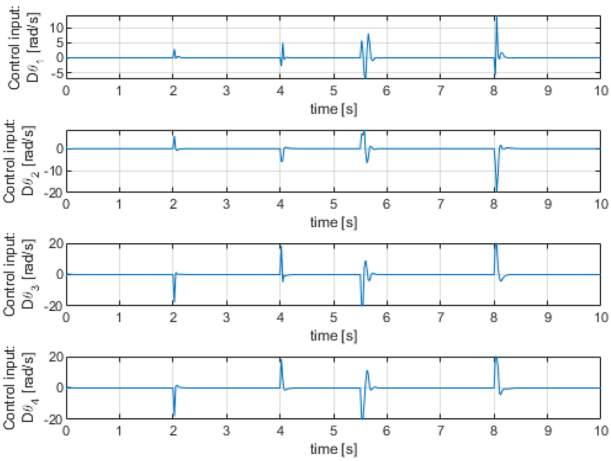


**Fig. 16** Control input of case 2 with MPC with optimization of virtual reference signal and integrator state

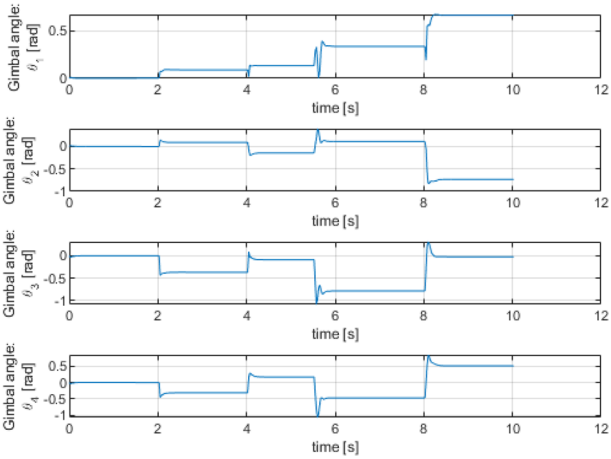


**Fig. 17** Gimbal angles of case 2 with MPC with optimization of virtual reference signal and integrator state

As from Figs. 24 and 25, the controller is able to track the desired behavior, with a small control authority (Fig. 26). Moreover, the singularity index (Fig. 27) shows the ability of the controller to avoid singularity.

Similar results are obtained for the rotation around *Z*-axis (Figs. 28, 29, 30 and 31). The singularity index shown in Fig. 31 implies that if the value of the index is close to zero, the CMG system is close to a singular state. Thus, the higher is this index, the higher is the controller ability to avoid singularity.

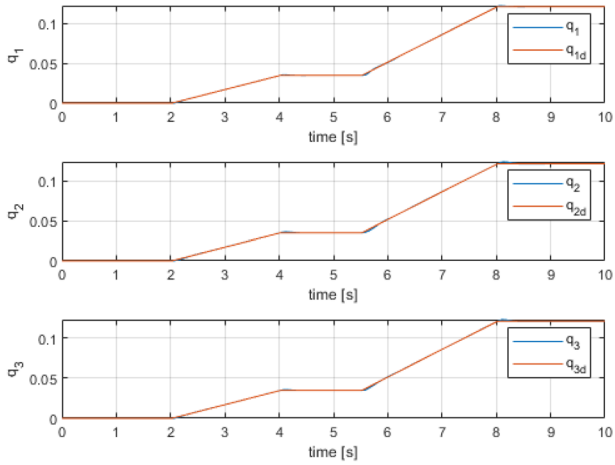


**Fig. 18** Quaternion vectorial components for case 2 with MPC with optimization of virtual reference signal and integrator state

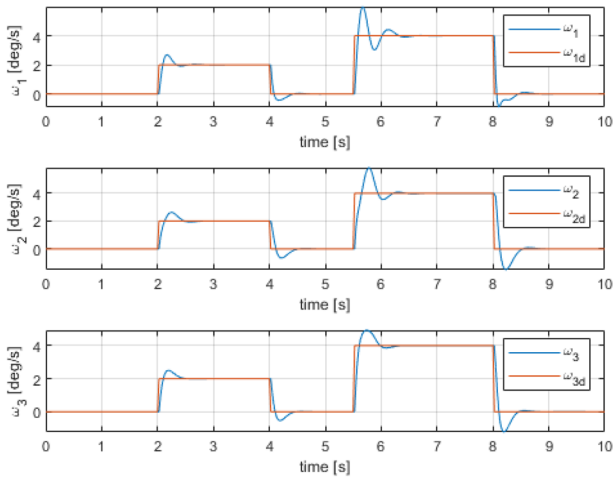


**Fig. 19** Angular velocity components for case 2 with classical MPC

## 6 Experimental results

After performing simulations in Matlab environment, the MPC with optimization of the virtual reference signal is converted in C++ program language, and applied to the testbed in Fig. 3.

Corresponding to the simulations in Sect. 4.2, we perform two cases of experiments: (1) a rotation around $X$ axis of $\omega_d = [5, 0, 0]^T$ deg/s, and (2) a rotation around $Z$ axis of $\omega_d = [0, 0, 5]^T$ deg/s. The desired trajectories of the angular velocity and

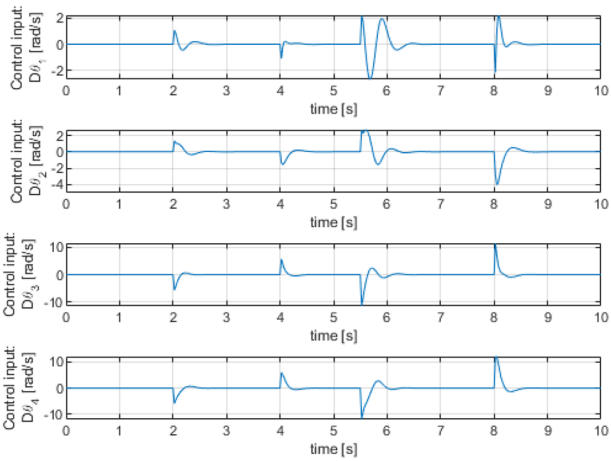


**Fig. 20** Control input of case 2 with classical MPC

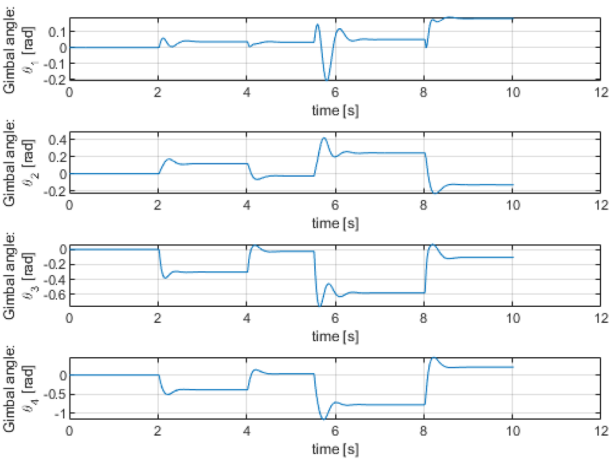


**Fig. 21** Gimbal angles of case 2 with classical MPC

the quaternion are the same as in Sect. 4.2, which are shown in the dotted lines in Figs. 32 and 36, and Figs. 33 and 37, respectively.

The experimental results of the *X*-axis rotation are shown in Figs. 32, 33, 34 and 35. Figs. 32 and 33 exhibit the angular velocity and the vectorial part of the quaternion of the testebed, respectively. Figure 34 shows the control inputs of the system. In addition, the singularity index is indicated in Fig. 35. Those experimental results correspond to the simulation results in Figs. 24, 6, 7 and 27 in Sect. 4.2.

Next, the experimental results of the *Z*-axis rotation are shown in Figs. 36, 37, 38 and 39. Similarly, the angular velocity, the vectorial part of the quaternion, the control inputs, and the singularity index are respectively shown in Figs. 36 37, 38 and

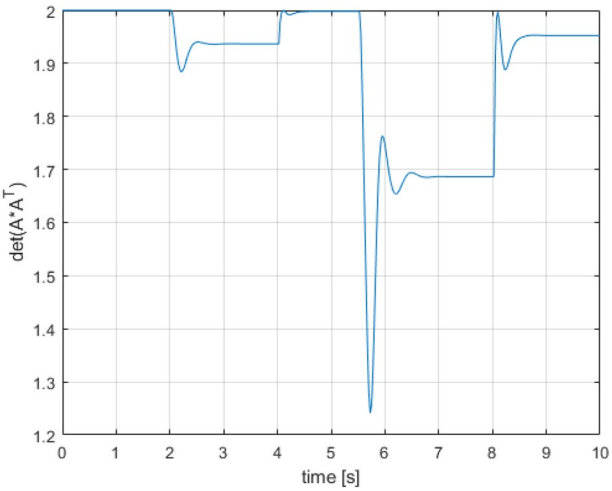


**Fig. 22** Singularity index of case 2 with classical MPC

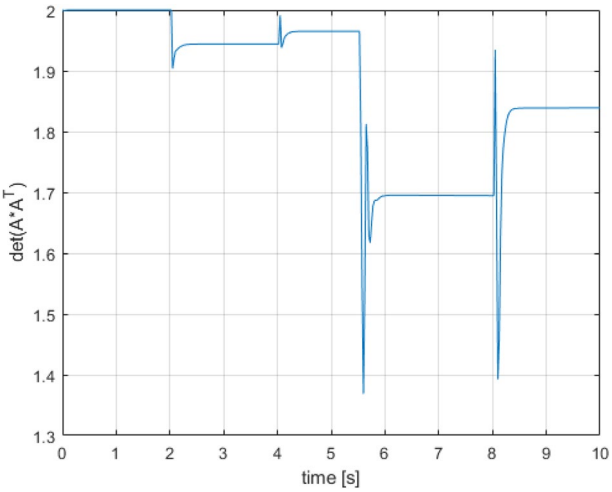


**Fig. 23** Singularity index of case 2 with MPC with optimization of virtual reference signal and integrator state

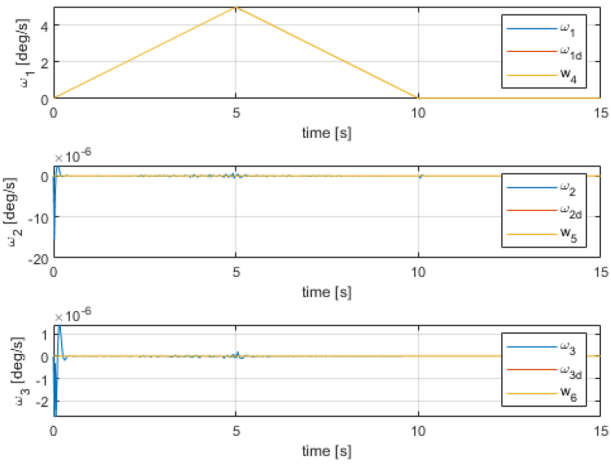


**Fig. 24** Angular velocity components for a rotation around *X* axis

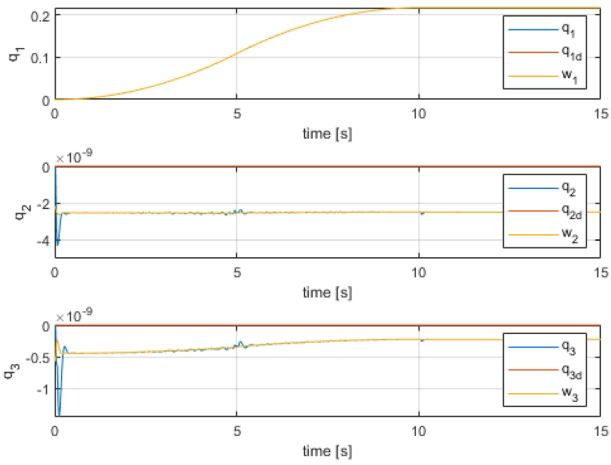


**Fig. 25** Quaternion vectorial components for a rotation around *X* axis

39. Those experimental results correspond to the simulation results in Figs. 28, 29, 30 and 31 in Sect. 4.2.

The unmodeled gravity torque due to the misalignment between the center of rotation and the center of gravity of the testbed inevitably arises in the ground experiments unlike the simulations. In particular, the *X*-axis rotation is seriously affected by the gravity torque disturbance. Moreover, under this setting, since the singular direction of two of the four CMGs is aligned to that of the required torque, the *X*-axis rotation is much difficult than the *Z*-axis rotation. Nonetheless, the present MPC controller compensates the disturbance and successfully achieves attitude

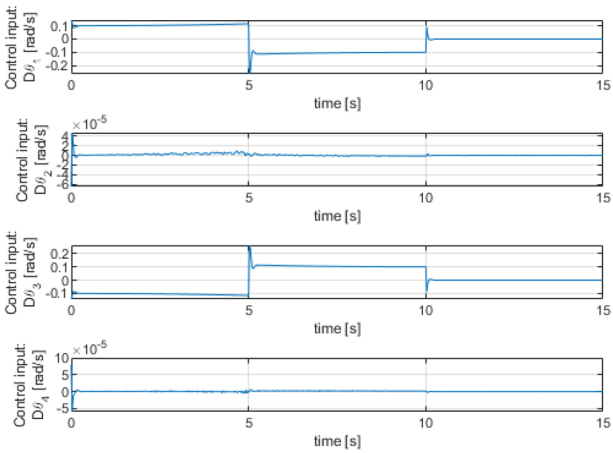


**Fig. 26** Control Input for a rotation around *X* axis

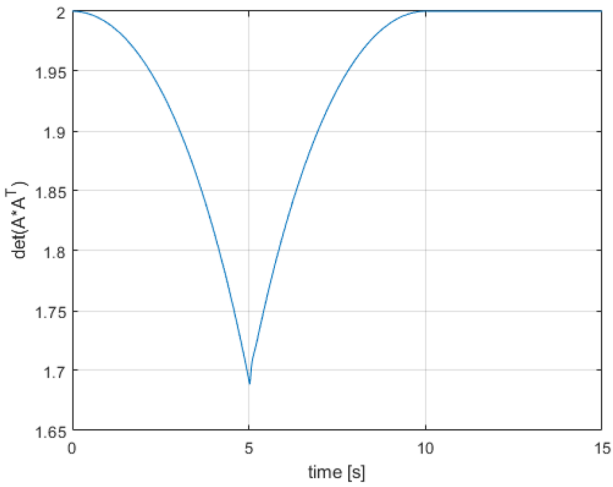


**Fig. 27** Singularity index for a rotation around *X* axis

tracking control in both cases. These experiments verify the effectiveness and feasibility of the proposed method.

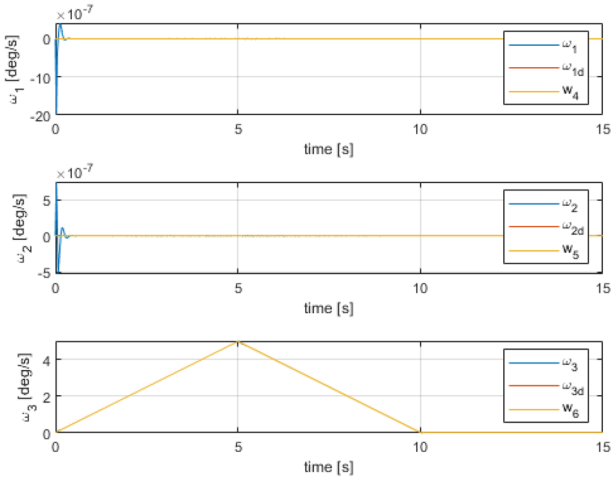


**Fig. 28** Angular velocity components for a rotation around $Z$ axis

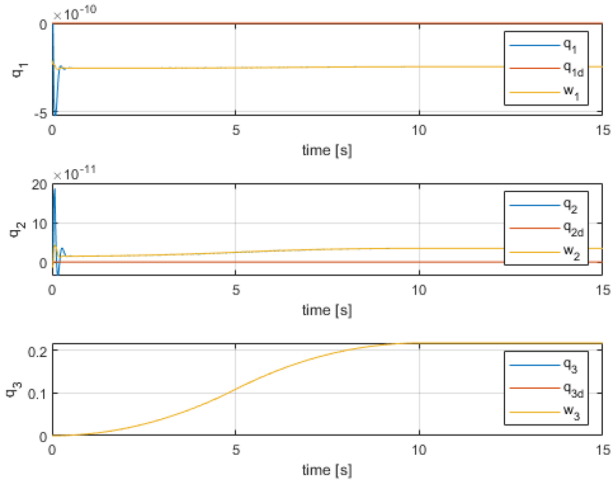


**Fig. 29** Quaternion vectorial components for a rotation around $Z$ axis

## 7 Conclusions

Control Moment Gyros (CMGs) are widely used for precise attitude control. This paper describes the attitude control problem of a spacecraft, using a Model Predictive Control (MPC) systems. The features of the considered linear MPC are: (i) a virtual reference, to guarantee input constraints satisfaction, and (ii) an integrator state as a servo compensator, to reduce the steady-state error. Moreover, the real-time implementability is discussed, considering an optimized solver for embedded systems. Good performance are observed in simulations. However, the

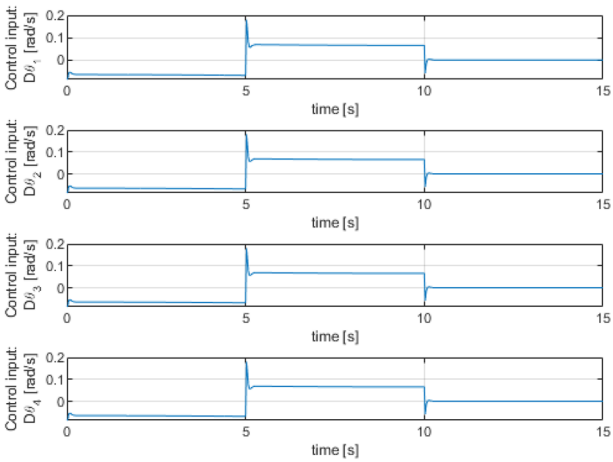


**Fig. 30** Control Input for a rotation around $Z$ axis

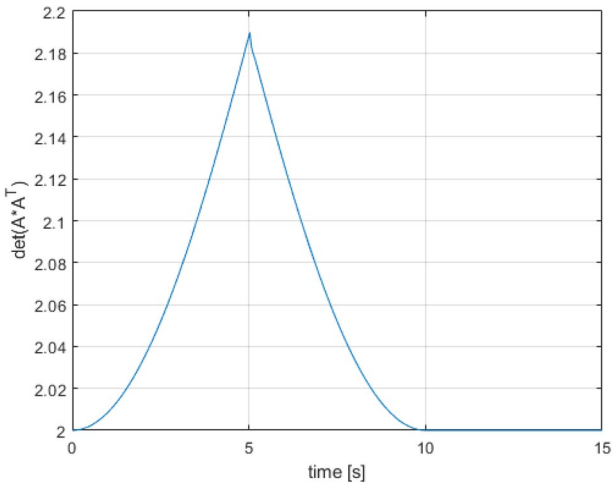


**Fig. 31** Singularity index for a rotation around $Z$ axis

proposed linear MPC control system is not able to handle with additive disturbances acting on the experimental testbed and it is only partially able to compensate this disturbance. In future works a control system able to handle and compensate the gravity gradient torque will be considered. Moreover, the gravity gradient disturbance will be also included in the model of the testbed itself.

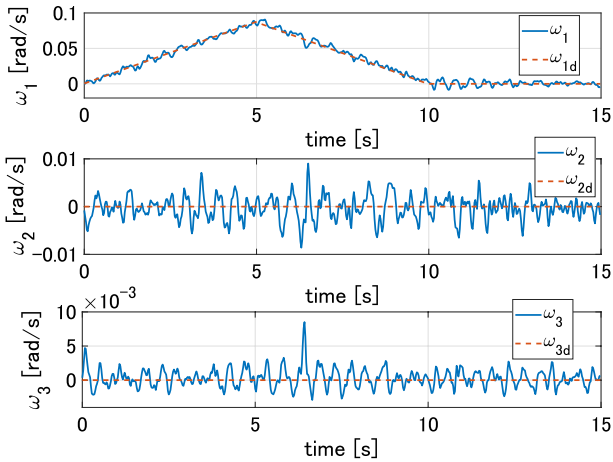


**Fig. 32** Angular velocity components of the testbed in the experiment of $X$-axis rotation

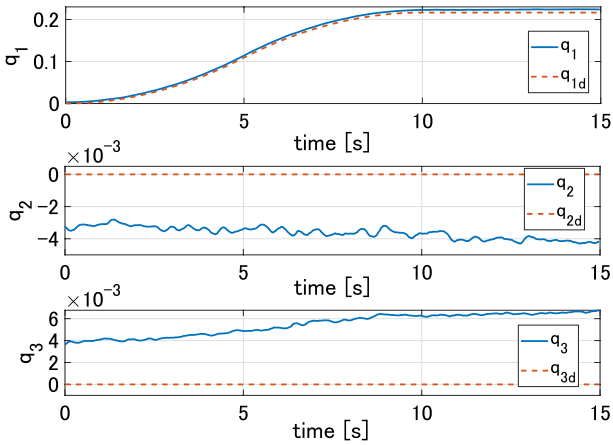


**Fig. 33** Vectorial part of the quaternion in the experiment of $X$-axis rotation

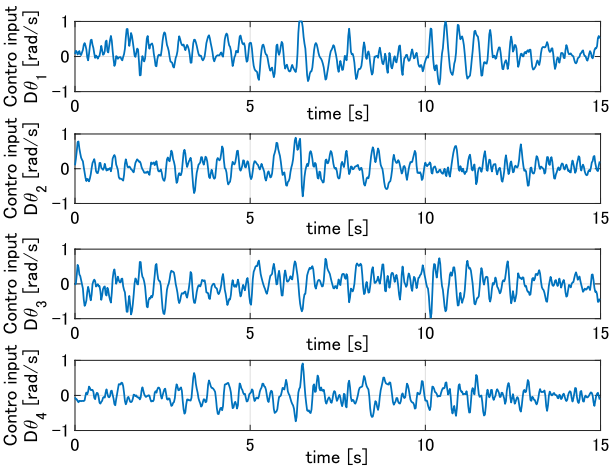


**Fig. 34** Control input signals in the experiment of *X*-axis rotation

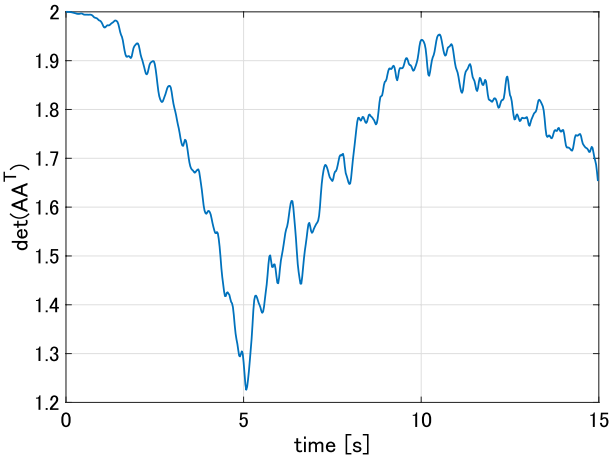


**Fig. 35** Singularity index in the experiment of *X*-axis rotation

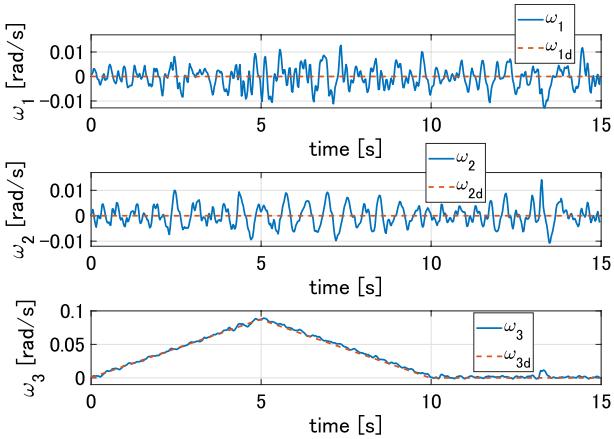


**Fig. 36** Angular velocity components of the testbed in the experiment of $Z$-axis rotation

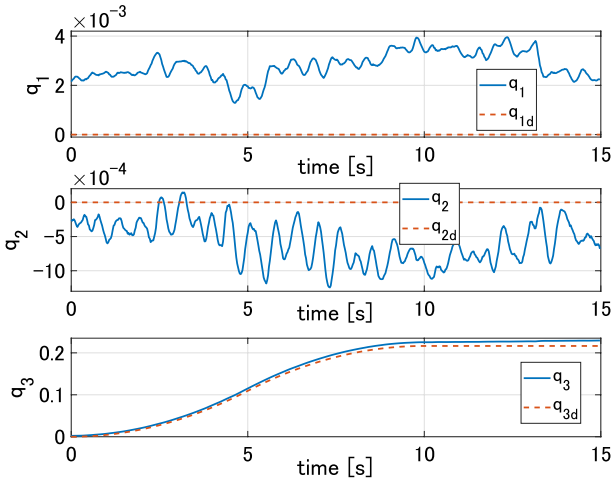


**Fig. 37** Vectorial part of the quaternion in the experiment of $Z$-axis rotation

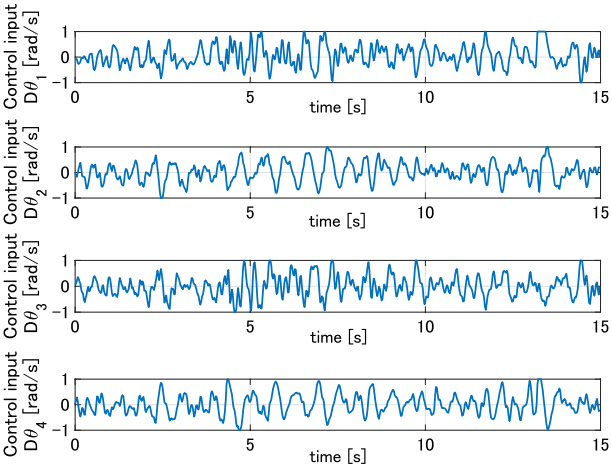


**Fig. 38** Control input signals in the experiment of $Z$-axis rotation

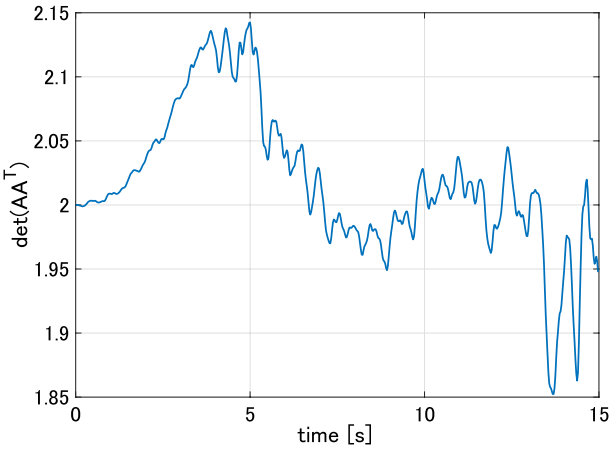


**Fig. 39** Singularity index in the experiment of $Z$-axis rotation

## Appendix: preliminaries on model predictive controller

Model Predictive Control is a control strategy that uses a dynamic model of the system, in order to make a prediction of the future behaviour of the system itself and consequently to "choose" the best control input. Since it is a model-based approach, the definition of the mathematical model of the system is important to obtain a good control performance. From one hand, the model has to be descriptive enough to capture all the most important characteristics of the system, in order to make predictions as much as possible near to the real evolution of the process. On the other hand, if the model is too detailed, the complexity is increased, and the control system reaches a high computational cost, that can be processed just by expensive hardware devices Rawlings and Mayne (2009). At each sample time, the controller computes the sequence of control inputs for the actual and next sample times in open loop fashion, but just the first one of this sequence is applied to the plant. At the next sample time, the procedure is repeated from the beginning, without considering the other control input values obtained in the previous sample time. The number of control inputs, collected in the control sequence, depends on the capability of the controller itself to "look forward" in the future. This characteristic is defined by a parameter called Receding Horizon $H_s$.

The following linear dynamic model of the system in state space formulation is considered:

$$x(k+1|k) = Ax(k|k) + Bu(k|k) \tag{38}$$

$$y(k|k) = Cx(k|k) + Du(k|k). \tag{39}$$

The first equation is the state equation, where $x \in \mathrm{R}^{n_p}$ is the system state, and $u \in \mathrm{R}^m$ is the control input. The second equation is the output equation, and $y \in \mathrm{R}^{n_y}$ is the output vector.

The matrices of the state space system are: (1) the state matrix $A \in \mathrm{R}^{n_p \times n_p}$, (2) the control matrix $B \in \mathrm{R}^{n_p \times m}$, (3) the output matrices $C \in \mathrm{R}^{n_y \times n_p}$ and $D \in \mathrm{R}^{n_y \times m}$. The notation $x(k+1|k)$ indicates the value of the state $x$ at time $k+1$ calculated at time $k$.

The prediction matrices $S$ and $T$ are defined using the defined model of the system, for the evaluation of the optimal sequence of control inputs

$$\begin{bmatrix} x(k+1|k) \\ x(k+2|k) \\ \vdots \\ x(k+H_s|k) \end{bmatrix} = \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{H_s-1}B & A^{H_s-2}B & \cdots & B \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+H_s-1|k) \end{bmatrix}}_{\tilde{u}(k)} + \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_s} \end{bmatrix}}_{T} x(k|k). \tag{40}$$

At each sample time a convex optimization problem is solved, where a cost function is minimized, to obtain the optimal sequence of the control inputs. The cost function at time $k$ is defined as

$$\begin{aligned} J(k) &= x(k+H_s|k)^T P x(k+H_s|k) \\ &+ \sum_{j=0}^{H_s-1} x(k+j|k)^T Q x(k+j|k) + u(k+j|k)^T R u(k+j|k), \end{aligned} \tag{41}$$

where $Q$,$R$ and $P$ are the weighting matrices, defined as follows

$$\begin{aligned} R &= R^T > 0 \\ Q &= Q^T \geq 0 \\ P &= P^T \geq 0. \end{aligned}$$

The cost function can be rewritten as

$$\begin{aligned} J(k) &= x(k|k)^T Q x(k|k) + \begin{bmatrix} x(k+1|k) \\ x(k+2|k) \\ \vdots \\ x(k+H_s|k) \end{bmatrix}^T \underbrace{\begin{bmatrix} Q & 0 & 0 & \cdots & 0 \\ 0 & Q & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & Q & 0 \\ 0 & 0 & \cdots & 0 & P \end{bmatrix}}_{\overline{Q}} \begin{bmatrix} x(k+1|k) \\ x(k+2|k) \\ \vdots \\ x(k+H_s|k) \end{bmatrix} \\ &+ \tilde{u}(k)^T \underbrace{\begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & R \end{bmatrix}}_{\overline{R}} \tilde{u}(k). \end{aligned} \tag{42}$$

Matrix $P$ is the terminal cost function, to ensure feasibility and stability to the MPC. Combining the prediction equation (40) and cost equation (42), we have

$$\begin{aligned} J(k) &= x(k|k)^T Q x(k|k) \\ &\quad + (S\widetilde{u}(k) + Tx(k|k))^T \overline{Q}(S\widetilde{u}(k) + Tx(k|k)) + \widetilde{u}(k)^T \overline{R}\widetilde{u}(k). \end{aligned} \tag{43}$$

Rewriting the previous equation:

$$\begin{aligned} J(k) &= \frac{1}{2}\widetilde{u}(k)^T \underbrace{2(\overline{R} + S^T\overline{Q}S)}_{H}\widetilde{u}(k) \\ &\quad + x(k|k)^T \underbrace{2T^T\overline{Q}S}_{F}\widetilde{u}(k) + \frac{1}{2}x(k|k)^T \underbrace{2(Q + T^T\overline{Q}T)}_{Y}x(k|k). \end{aligned} \tag{44}$$

One of the main feature of the MPC controller is the definition of the input and state constraints, in the optimization problem. The input and state constraints are defined as follows:

$$u_{min} \leqslant u(k+j|k) \leqslant u_{max} \quad j = 0, \dots, H_s - 1 \tag{45}$$

$$x_{min} \leqslant x(k+j|k) \leqslant x_{max} \quad j = 1, \dots, H_s. \tag{46}$$

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

$$\begin{aligned} &\min_{\widetilde{u}(k)} J(k) \\ &\quad \text{subject to (46), (46) and (40).} \end{aligned} \tag{47}$$

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement.

**Declarations**

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

## References

Academies National, of Sciences and Medicine E, (2016) NASA space technology roadmaps and priorities revisited. The National Academies Press, Washington

Ashok K, Aditya R, Joglekar HK, Siva M, Philip N (2016) Cmg configuration and steering approach for spacecraft rapid maneuvers. IFAC-PapersOnLine 49(1):118–123

Bedrossian N, Jang J-WJ, Alaniz A, Johnson M, Sebelius K (2005) International space station us gnc attitude hold controller design for orbiter repair maneuver. 08

Bemporad A, Morari M, Dua V, Pistikopoulos EN (2002) The explicit linear quadratic regulator for constrained systems. Automatica 38(1):3–20

Chakravorty J (2012) Singularity avoidance in single gimbal cmg using the theory of potential functions. In: Proceedings of the 10th World Congress on Intelligent Control and Automation, pp. 1103–1108

Ciavola M, Capello E, Satoh S, Yamada K (2019) Comparison of SDA and MPC controllers of a CMG system for the singularity problem. In: Proceedings of the SICE Annual Conference 2019, September 10-13, 2019, Hiroshima, Japan, pp. 541–544, SICE

Courie I, Jenie Y, Poetro R (2018) Simulation of satellite attitude control using single gimbal control moment gyro (sgcmg) system. In: Journal of Physics: Conference Series, vol. 1130, p. 012003, IOP Publishing

Currie J (2014) Practical applications of industrial optimization: from high-speed embedded controllers to large discrete utility systems. PhD thesis, PhD Thesis, Dept. Elect. Eng., AUT University

Eren U, Prach A, Koçer BB, Raković SV, Kayacan E, Açıkmeşe B (2017) Model predictive control in aerospace systems: current state and opportunities. J Guidance, Control, Dyn 40(7):1541–1566

Ford KA, Hall CD (2000) Singular direction avoidance steering for control-moment gyros. J Guidance, Control, Dyn 23(4):648–656

Guennebaud G, Furfaro A, Gaspero L (2011) "eiquadprog.hh,"

Guiggiani A, Kolmanovsky I, Patrinos P, Bemporad A (2015) Constrained model predictive control of spacecraft attitude with reaction wheels desaturation. In: 2015 European Control Conference (ECC), pp. 1382–1387, IEEE

Higashiyama D, Shoji Y, Satoh S, Jikuya I, Yamada K (2020) Attitude control for spacecraft using pyramid-type variable-speed control moment gyros. Acta Astronautica 173:252–265

Kraft RH (1993) Cmg singularity avoidance in attitude control of a flexible spacecraft. In: 1993 American Control Conference, pp. 56–58, IEEE

Lee H, Lee D, Bang H, Tahk M-J (2007) Singularity avoidance law for cmg using virtual actuator. IFAC Proc Vol 40(7):305–310

Lofberg J (2004) Yalmip: A toolbox for modeling and optimization in matlab. In: 2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508), pp. 284–289, IEEE

Mammarella M, Lorenzen M, Capello E, Park H, Dabbene F, Guglieri G, Romano M, Allgöwer F (2018) An offline-sampling smpc framework with application to autonomous space maneuvers. IEEE Trans Control Syst Technol 17:859–865

Markley FL, Crassidis JL (2014) Fundamentals of spacecraft attitude determination and control. Springer, New York

Mayne DQ (2014) Model predictive control: Recent developments and future promise. Automatica 50(12):2967–2986

Miceli D (2007) Analisi di un sistema avionico basato su control moment gyro

Qin S, Badgwell TA (2003) A survey of industrial model predictive control technology. Control Eng Pract 11(7):733–764

Rawlings JB, Mayne DQ (2009) Model predictive control: theory and design. Nob Hill Pub, Portland

Srinivasan K, Gandhi D, Venugopal M (2014) Spacecraft attitude control using control moment gyro reconfiguration. Stud Inform Control 23(3):286

Wada N, Tsurushima S (2016) Constrained mpc to track time-varying reference signals: online optimization of virtual reference signals and controller states. IEEJ Trans Electri Electron Eng 11:S65–S74

Wang Y, Boyd S (2009) Fast model predictive control using online optimization. IEEE Trans Control Syst Technol 18(2):267–278

Wie B (2005) New singularity escape/avoidance steering logic for control moment gyro systems. J Guidance Control Dyn 28(948–956):09

Wright SJ (1996) Applying new optimization algorithms to more predictive control. tech. rep., Argonne National Lab., IL (United States)

Xin X, Li Z, Liu X, Chen Z (2015) Hybrid control and steering logic for the cmg-based spacecraft attitude control system. In: 2015 IEEE Conference on Control Applications (CCA), pp. 780–785

Yadlin R (2011) Stochastic model predictive control and its application for small satellite attitude control. PhD thesis, Oxford University, UK

Yu-Geng X, De-Wei L, Shu L (2013) Model predictive control-status and challenges. Acta Automatica Sinica 39(3):222–236

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.