Aerospace Science and Technology 128 (2022) 107778

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Multiplexed MPC attitude control of a moving mass satellite using dual-rate piecewise affine model

Yuandong Hu [a], Zhengliang Lu [a], K.V. Ling [b], Wenhe Liao [a,*], Xiang Zhang [a]

[a] School of Mechanical Engineering, Nanjing University of Science and Technology, No. 200 Xiaolingwei Street, Nanjing 210094, Jiangsu, China
[b] School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 17 April 2022
Received in revised form 15 June 2022
Accepted 18 July 2022
Available online 25 July 2022
Communicated by Choon Ki Ahn

Keywords:
Attitude control
Aerodynamic torques
Moving mass
Piecewise affine
Model predictive control
Hardware-in-the-loop simulation

ABSTRACT

This paper proposes a Multiplexed MPC algorithm for the dual-rate piecewise affine (PWA) model to control the attitude of a low-Earth-orbit (LEO) moving mass satellite. Although still facing an inherent inertial disturbance problem, the moving mass technology is a promising way to counter the strong aerodynamic torques in LEO. To make the satellite attitude fully actuated, a magnetorquer is used to cooperate with the moving mass system, resulting in a two-input control system. Through a strategy that updates inputs asynchronously, the proposed algorithm effectively balances the inertial disturbances caused by mass motions, the system's dynamic performance, and the controller's computational complexity. Error attitude kinematic and dynamic equations are derived, to convert the tracking problem into an equivalent regulation problem. The PWA approximation is used to transform the nonlinear attitude control equations described by error modified Rodrigues parameter to a linear state-space model for MPC design. Then, a new form of Multiplexed MPC, namely dual-rate PWA-MMPC, is developed for this dual-rate PWA model, which updates two control inputs at different rates. Despite only finding sub-optimal solutions to the original problem by stability analysis, this algorithm outperforms other control strategies through its ability to coordinate multiple inputs, deal with constraints, and reduce computational complexity. Hardware-in-the-loop simulations are conducted to demonstrate the effectiveness of the proposed methodology for the attitude control.

1. Introduction

Low Earth Orbit (LEO) satellites have received much attention for their essential application value in civilian and military domains [1]. Compared with other satellites, LEO satellites are closer to the Earth with many advantages, including more powerful signals, higher resolution images, and lower launch costs. However, the lower the orbit, the more severe the impacts of aerodynamic torques on the satellite attitude [2]. Conventional attitude control actuators of small satellites, that is, reaction wheels, can ensure the high-precision control of the attitude [3]. But when facing the strong aerodynamic torques in LEO, reaction wheels have to compensate these torques in reverse all the time. The usual magnetorquer can no longer unload angular momentum in time, and a propulsion system is required instead. Or even the propulsion system is directly used to control the attitude [4]. By comparison, the moving mass device (MMD) is to actively exploit aerodynamic torques for attitude control, and thus it will have stronger ability as the orbit descends. Therefore, this paper considers using the MMD to actively control the attitude of LEO satellites.

The satellite's aerodynamic torque is caused by the non-parallelism between the aerodynamic force and the vector from the satellite center of mass (CoM) to the center of pressure (CoP) [5]. Previous works [6,7] investigated novel ways that use deployable/retractable or rotatable plates to stabilize the satellite by relocating the CoP. Furthermore, Sun et al. [8] proposed to use rotatable and retractable plates attached on the satellite to change aerodynamic forces and torques, simultaneously controlling the translational and rotational motions. However, the increasing windward area would further accelerate the unwanted orbit decay of the LEO satellite.

* Corresponding author.
E-mail addresses: huyuandong@njust.edu.cn (Y. Hu), 112010115@njust.edu.cn (Z. Lu), EKVLING@ntu.edu.sg (K.V. Ling), hu18362903019@163.com (W. Liao), zhxiang2002@126.com (X. Zhang).

https://doi.org/10.1016/j.ast.2022.107778
1270-9638/

Alternatively, aerodynamic forces can be altered by changing the satellite CoM through the internal MMD. Using the MMD has the advantage that the satellite geometry remains unchanged. He et al. [9] adopted four movable masses distributed in a plane to stabilize the satellite using aerodynamic torques and internal momentum exchange torques. Lu [10] designed an underactuated control law to stabilize a 2U LEO CubeSat. In particular, Chesi et al. [11] and Virgili-Llop et al. [12] proposed a strategy that complements the MMD with one reaction wheel or a magnetorquer, achieving the full-actuation of the satellite attitude. The main challenge in designing attitude control algorithms for such a moving mass satellite is how to deal with the system disturbances caused by uncertainties of aerodynamic forces and inertial torques. Previous works mostly focused on designing advanced algorithms with robustness to handle the uncertainties of aerodynamic forces [13,14], but did not draw enough attention to the dynamic effects of inertial disturbances [11,12]. Several previous works developed suitable control algorithms to slow down the mass motions to reduce inertial torques, such as the discrete PID algorithm [15] and the backstepping algorithm with input constraints [16]. These algorithms all took the ideal torque as input, leading to the degradation of the system's dynamic performance. Hence for such a two-input system in [11], this paper employs the MPC algorithm to coordinate the two inputs to balance the inertial disturbances and the system's dynamic performance.

Model predictive control (MPC) is an established technology in a wide range of process industries, and has attracted interest in aerospace applications, such as orbit tracking [17,18], attitude maneuver [19,20], and attitude-orbit coupling control [21,22]. In these applications, MPC offers good performances, especially for coordinating multiple inputs and satisfying system constraints. In this paper, MPC is adopted to control the attitude of a LEO satellite with two actuators, the MMD and the magnetorquer. However, computational complexity, particularly computation time, is an outstanding issue when facilitating applications of conventional MPC in small satellites.

Direct application of nonlinear MPC (NMPC) in the satellite attitude control system would result in an extremely complex finite-horizon, nonconvex optimization [23,24]. The alternative of linear MPC (LMPC) is particularly effective since it only involves solving tractable convex quadratic programs [25]. Many previous works [26–28] linearized nonlinear systems and formulated a LMPC algorithm instead, such as the typical nonlinear system of the satellite attitude control [28]. Imura et al. [29,30] proposed a piecewise affine (PWA) linearization approach for the real-time onboard implementation, avoiding computing the real-time Jacobian matrix. This PWA approximation enables using LMPC in strongly nonlinear systems such as those encountered in attitude maneuvers [20].

Conventional LMPC updates all control inputs simultaneously, which could result in a high computational complexity [31]. For a more rapid response to disturbances, Ling et al. [32,33] proposed a Multiplexed MPC (MMPC), which updates only one input at a time and moves inputs in a periodic order. This MMPC algorithm not only reduces the optimization complexity at each time step, but also offers a chance of updating multiple inputs asynchronously. The MMPC idea has been implemented in some applications, such as the aircraft engine control [34] and the cutterhead control of tunnel boring machines [35]. In this paper, we employ the MMPC idea and extend it to the dual-rate PWA system to update two inputs at different rates, thereby balancing the inertial disturbances and the system's dynamic performance.

The main contributions of this paper are as follows:

(1) The idea of operating the MMD and magnetorquer actuators asynchronously is innovatively proposed to control the attitude of a LEO satellite. Asynchronously updating inputs balances the effects of inertial disturbances and the system's dynamic performance.
(2) A PWA model in state-space form is derived based on the error modified Rodrigues parameter, rather than the usual error quaternion [20,21].
(3) The MMPC algorithm is extended to the dual-rate PWA system, achieving a low computational complexity algorithm, namely dual-rate PWA-MMPC. This algorithm is suitable for the strongly nonlinear system and facilitates the real-time onboard implementation on small satellites.

The remainder of this paper is organized as follows. Section 2 describes dynamic equations of the LEO satellite with MMD based on the generalized theorem of moment of momentum. We then derive a PWA model to approximate the nonlinear error kinematic and dynamic equations in Section 3. Section 4 details the proposed dual-rate PWA-MMPC algorithm and its stability analysis. Section 5 presents results from the hardware-in-the-loop simulations, and Section 6 concludes this paper.

2. Modelling

Aerodynamic forces can be leveraged for the attitude control of LEO satellites, and the moving mass technology provides an approach to actively control aerodynamic torques by changing the satellite CoM. However, due to the inability to exert a torque parallel to the aerodynamic force, the satellite system with purely aerodynamic actuation is instantaneously underactuated with the unactuated axis not fixed to the body [11]. Thus, a magnetorquer is introduced to generate a torque parallel to the aerodynamic force, resulting in a two-input control system.

Consider a satellite including n moving masses, and the body implies the part of the satellite other than these moving masses. Fig. 1 illustrates the coordinate systems and the pertinent variables required to derive dynamic equations:

(1) $O_E X_I Y_I Z_I$ is the inertial coordinate system; $O_C X_B Y_B Z_B$ is the body-fixed coordinate system;
(2) The points O_S, O_P, and O_C are the satellite CoM, the satellite CoP, and the satellite geometric center respectively; The point O_M is the body CoM;
(3) The vectors $\boldsymbol{R}$ and $\boldsymbol{R}_S$ locate O_C and O_S in the $O_E X_I Y_I Z_I$ system respectively; The vectors $\boldsymbol{r}_B$, $\boldsymbol{r}_S$, $\boldsymbol{r}_P$, and $\boldsymbol{p}_i$ locate O_M, O_S, O_P, and the moving mass i in the $O_C X_B Y_B Z_B$ system respectively;
(4) The force $\boldsymbol{F}_a$ is the aerodynamic force; $\boldsymbol{T}_m$ and $\boldsymbol{T}_e$ are the magnetic torque and the unknown environmental disturbance;
(5) $\boldsymbol{\omega}_{bi} = [\omega_{bi1}, \omega_{bi2}, \omega_{bi3}]^{\mathrm{T}}$ is the satellite inertial angular velocity, and $\boldsymbol{\omega}_m^i$ is the angular velocity of the moving mass i in the $O_C X_B Y_B Z_B$ system.

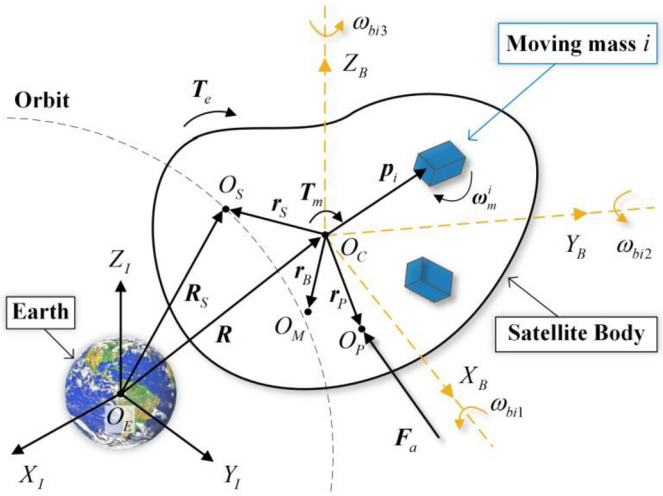

Fig. 1. Force diagram of the satellite with moving masses.

2.1. Attitude model of a satellite with MMD

First, for any vector $\boldsymbol{v}$ and any coordinate system E, define the operations ${}^{E}\dot{\boldsymbol{v}}$ and ${}^{E}\ddot{\boldsymbol{v}}$ as the first and second derivative with respect to E. Based on the generalized theorem of moment of momentum [12], the attitude dynamic equation, which considers O_C as the reference point, is given by

$$\boldsymbol{M}_T = {}^{I}\dot{\boldsymbol{H}} + \boldsymbol{S} \times \boldsymbol{a} \tag{1}$$

where, $\boldsymbol{M}_T$ denotes the external torque; $\boldsymbol{H}$ denotes the total angular momentum; $\boldsymbol{S}$ is the first moment to the reference point O_C; and, $\boldsymbol{a}$ is the inertial acceleration of the reference point.

From the vector differentiation rule between coordinate systems [36], ${}^{I}\dot{\boldsymbol{H}}$ can be written as

$$\begin{aligned}
{}^{I}\dot{\boldsymbol{H}} &= \boldsymbol{J}_B {}^{B}\dot{\boldsymbol{\omega}}_{bi} + m_B {}^{B}\dot{\boldsymbol{r}}_B \times {}^{I}\dot{\boldsymbol{r}}_B + m_B \boldsymbol{r}_B \times {}^{BI}\ddot{\boldsymbol{r}}_B + \boldsymbol{\omega}_{bi} \times \left[\boldsymbol{J}_B \boldsymbol{\omega}_{bi} + m_B \boldsymbol{r}_B \times {}^{I}\dot{\boldsymbol{r}}_B\right] \\
&\quad + \boldsymbol{J}_i\left({}^{B}\dot{\boldsymbol{\omega}}_{bi} + {}^{B}\dot{\boldsymbol{\omega}}_m^i\right) + m_i {}^{B}\dot{\boldsymbol{p}}_i \times {}^{I}\dot{\boldsymbol{p}}_i + m_i \boldsymbol{p}_i \times {}^{BI}\ddot{\boldsymbol{p}}_i + \boldsymbol{\omega}_{bi} \times \left[\boldsymbol{J}_i\left(\boldsymbol{\omega}_{bi} + \boldsymbol{\omega}_m^i\right) + m_i \boldsymbol{p}_i \times {}^{I}\dot{\boldsymbol{p}}_i\right] \\
&= \left(\boldsymbol{J}_B + \boldsymbol{J}_i\right){}^{B}\dot{\boldsymbol{\omega}}_{bi} + \boldsymbol{J}_i {}^{B}\dot{\boldsymbol{\omega}}_m^i + m_B {}^{B}\dot{\boldsymbol{r}}_B \times \left(\boldsymbol{\omega}_{bi} \times \boldsymbol{r}_B\right) + m_B \boldsymbol{r}_B \times \left({}^{B}\ddot{\boldsymbol{r}}_B + \boldsymbol{\omega}_{bi} \times {}^{B}\dot{\boldsymbol{r}}_B + {}^{B}\dot{\boldsymbol{\omega}}_{bi} \times \boldsymbol{r}_B\right) \\
&\quad + \boldsymbol{\omega}_{bi} \times \left[\boldsymbol{J}_B \boldsymbol{\omega}_{bi} + m_B \boldsymbol{r}_B \times \left({}^{B}\dot{\boldsymbol{r}}_B + \boldsymbol{\omega}_{bi} \times \boldsymbol{r}_B\right)\right] + m_i \boldsymbol{p}_i \times \left({}^{B}\ddot{\boldsymbol{p}}_i + \boldsymbol{\omega}_{bi} \times {}^{B}\dot{\boldsymbol{p}}_i + {}^{B}\dot{\boldsymbol{\omega}}_{bi} \times \boldsymbol{p}_i\right) \\
&\quad + m_i {}^{B}\dot{\boldsymbol{p}}_i \times \left(\boldsymbol{\omega}_{bi} \times \boldsymbol{p}_i\right) + \boldsymbol{\omega}_{bi} \times \left[\boldsymbol{J}_i\left(\boldsymbol{\omega}_{bi} + \boldsymbol{\omega}_{mbi}^b\right) + m_i \boldsymbol{p}_i \times \left({}^{B}\dot{\boldsymbol{p}}_i + \boldsymbol{\omega}_{bi} \times \boldsymbol{p}_i\right)\right]
\end{aligned} \tag{2}$$

where, $\boldsymbol{J}_B$ and $\boldsymbol{J}_i$ are the inertia matrices of the body and the moving mass i; m_B and m_i are the masses of the body and the moving mass i.

The inertial acceleration of the satellite is caused by the aerodynamic force $\boldsymbol{F}_a$ and the satellite gravity $\boldsymbol{G}$, and hence

$$\boldsymbol{a} = {}^{I}\ddot{\boldsymbol{R}} = {}^{I}\ddot{\boldsymbol{R}}_S - {}^{I}\ddot{\boldsymbol{r}}_S = \left(\boldsymbol{F}_a + \boldsymbol{G} - m_B {}^{I}\ddot{\boldsymbol{r}}_B - \sum_{i=1}^{n} m_i {}^{I}\ddot{\boldsymbol{p}}_i\right) \Big/ \left(m_B + \sum_{i=1}^{n} m_i\right) \tag{3}$$

Moreover, $\boldsymbol{M}_T$ and $\boldsymbol{S}$ can be expressed as

$$\boldsymbol{M}_T = \boldsymbol{r}_P \times \boldsymbol{F}_a + \boldsymbol{r}_S \times \boldsymbol{G} + \boldsymbol{T}_m + \boldsymbol{T}_e \tag{4}$$

$$\boldsymbol{S} = m_B \boldsymbol{r}_B + \sum_{i=1}^{n} m_i \boldsymbol{p}_i \tag{5}$$

In this section, we default that the derivative of a vector without left superscript is with respect to the $O_C X_B Y_B Z_B$ system. Then, substituting Eqs. (2), (3), (4), and (5) into Eq. (1), the attitude dynamic equation can be written as

$$\begin{aligned}
&\left(\boldsymbol{J}_B + \sum_{i=1}^{n} \boldsymbol{J}_i + \boldsymbol{J}_m\right)\dot{\boldsymbol{\omega}}_{bi} + \sum_{i=1}^{n} \boldsymbol{J}_i \dot{\boldsymbol{\omega}}_m^i + \boldsymbol{\omega}_{bi} \times \left[\left(\boldsymbol{J}_B + \sum_{i=1}^{n} \boldsymbol{J}_i + \boldsymbol{J}_m\right)\boldsymbol{\omega}_{bi} + \sum_{i=1}^{n} \boldsymbol{J}_i \boldsymbol{\omega}_m^i\right] \\
&\quad = \left(\boldsymbol{r}_P - \boldsymbol{r}_S\right) \times \boldsymbol{F}_a + \boldsymbol{T}_m + \boldsymbol{T}_a + \boldsymbol{T}_c + \boldsymbol{T}_e
\end{aligned} \tag{6}$$

where

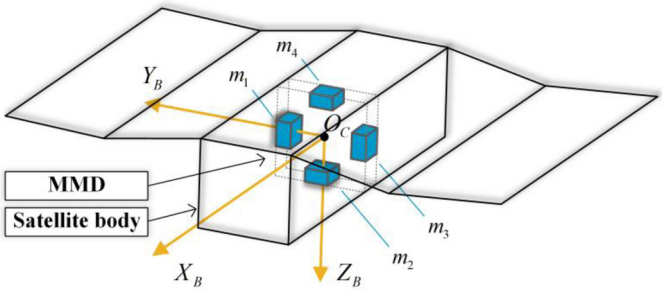

Fig. 2. The doubly symmetric mass distribution of MMD.

$$\begin{cases} \boldsymbol{J}_m = \left(m_B + \sum_{i=1}^{n} m_i\right)[\boldsymbol{r}_S]^{\times}[\boldsymbol{r}_S]^{\times} - m_B[\boldsymbol{r}_B]^{\times}[\boldsymbol{r}_B]^{\times} - \sum_{i=1}^{n}\left(m_i[\boldsymbol{p}_i]^{\times}[\boldsymbol{p}_i]^{\times}\right) \\ \boldsymbol{T}_a = \boldsymbol{r}_S \times \sum_{i=1}^{n} m_i\ddot{\boldsymbol{p}}_i - \sum_{i=1}^{n}(m_i\boldsymbol{p}_i \times \ddot{\boldsymbol{p}}_i) \\ \boldsymbol{T}_c = 2\boldsymbol{r}_S \times \left(\boldsymbol{\omega}_{bi} \times \sum_{i=1}^{n} m_i\dot{\boldsymbol{p}}_i\right) - \sum_{i=1}^{n}\left[2m_i\boldsymbol{p}_i \times (\boldsymbol{\omega}_{bi} \times \dot{\boldsymbol{p}}_i)\right] \end{cases} \tag{7}$$

in which the operation $[\cdot]^{\times}$ represents the cross-product matrix.

Referring to [37], we adopt the doubly symmetric mass distribution of MMD, as shown in Fig. 2, to reduce the additional torques $\boldsymbol{T}_a$ and $\boldsymbol{T}_c$ caused by mass motions. These two pairs of moving masses are symmetrically distributed about the O_CY_B and O_CZ_B axes, and each pair moves completely synchronously.

Moreover, the following three assumptions are made:

(1) The MMD is mounted in the plane $O_CY_BZ_B$;
(2) Four moving masses have the same mass, and each mass is considered a point mass, that is, $m_i = m_s (i = 1, 2, 3, 4)$, and $\boldsymbol{J}_i = 0$;
(3) Moving masses have no rotational motions relative to the satellite body, that is, $\boldsymbol{\omega}_m^i = 0$ and $\dot{\boldsymbol{\omega}}_m^i = 0$.

Thus, the position vectors of moving masses are $\boldsymbol{p}_1 = [0, l_1, \delta]^{\mathrm{T}}$, $\boldsymbol{p}_2 = [0, \delta, l_2]^{\mathrm{T}}$, $\boldsymbol{p}_3 = [0, l_1, -\delta]^{\mathrm{T}}$, and $\boldsymbol{p}_4 = [0, -\delta, l_2]^{\mathrm{T}}$, where l_1 and l_2 are controllable positions, and δ is the constant deviation. Then, the attitude dynamic equation can be simplified to be

$$\begin{aligned} &(\boldsymbol{J}_B + \boldsymbol{J}_m)\dot{\boldsymbol{\omega}}_{bi} + \boldsymbol{\omega}_{bi} \times (\boldsymbol{J}_B + \boldsymbol{J}_m)\boldsymbol{\omega}_{bi} \\ &= \frac{2m_s}{m_B + 4m_s}[\boldsymbol{F}_a]^{\times}\boldsymbol{H}_m\boldsymbol{L} - [\boldsymbol{B}_{mag}]^{\times}\boldsymbol{m}_{mag} + \boldsymbol{T}_a + \boldsymbol{T}_c + \boldsymbol{T}_e + \left(\boldsymbol{r}_P - \frac{m_B\boldsymbol{r}_B}{m_B + 4m_s}\right) \times \boldsymbol{F}_a \end{aligned} \tag{8}$$

where, $\boldsymbol{L} = [l_1, l_2]^{\mathrm{T}}$ is the controllable position vector; $\boldsymbol{m}_{mag}$ is the controllable magnetic moment; $\boldsymbol{B}_{mag}$ is the geomagnetic vector in the $O_CX_BY_BZ_B$ system; and, the constant matrix $\boldsymbol{H}_m$ is

$$\boldsymbol{H}_m = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{9}$$

2.2. Error kinematics and dynamics

To simplify the stability analysis of the controller, we consider error kinematics and dynamics to convert the tracking problem into an equivalent regulation problem.

First, the other two coordinate systems are defined here: an orbit coordinate system $O_CX_OY_OZ_O$ and a desired coordinate system $O_CX_DY_DZ_D$. Define $\boldsymbol{\omega}_{bo}$, $\boldsymbol{\omega}_{oi}$, $\boldsymbol{\omega}_d$, and $\boldsymbol{\omega}_e$ as the satellite attitude to the $O_CX_OY_OZ_O$ system, the orbital angular velocity, the desired angular velocity, and the error angular velocity, respectively.

Then, it yields $\boldsymbol{\omega}_{bi} = \boldsymbol{\omega}_e + \boldsymbol{A}_e\boldsymbol{\omega}_d + \boldsymbol{A}_{bo}\boldsymbol{\omega}_{oi}$ where $\boldsymbol{A}_{bo}$ and $\boldsymbol{A}_e$ are the direction cosine matrices from the $O_CX_BY_BZ_B$ system and the $O_CX_DY_DZ_D$ system to the $O_CX_OY_OZ_O$ system. Therefore, the error dynamic equation can be written as

$$\begin{aligned} (\boldsymbol{J}_B + \boldsymbol{J}_m)\dot{\boldsymbol{\omega}}_e = &(\boldsymbol{J}_B + \boldsymbol{J}_m)(\boldsymbol{\omega}_e \times \boldsymbol{A}_e\boldsymbol{\omega}_d - \boldsymbol{A}_e\dot{\boldsymbol{\omega}}_d + \boldsymbol{\omega}_{bo} \times \boldsymbol{A}_{bo}\boldsymbol{\omega}_{oi} - \boldsymbol{A}_{bo}\dot{\boldsymbol{\omega}}_{oi}) - \boldsymbol{\omega}_{bi} \times (\boldsymbol{J}_B + \boldsymbol{J}_m)\boldsymbol{\omega}_{bi} \\ &+ \frac{2m_s}{m_B + 4m_s}[\boldsymbol{F}_a]^{\times}\boldsymbol{H}_m\boldsymbol{L} - [\boldsymbol{B}_{mag}]^{\times}\boldsymbol{m}_{mag} + \boldsymbol{T}_a + \boldsymbol{T}_c + \boldsymbol{T}_e + \left(\boldsymbol{r}_P - \frac{m_B\boldsymbol{r}_B}{m_B + 4m_s}\right) \times \boldsymbol{F}_a \end{aligned} \tag{10}$$

Moreover, the modified Rodrigues parameter (MRP) $\boldsymbol{\sigma}_e$ is used to represent the attitude mapping from the $O_CX_DY_DZ_D$ system to the $O_CX_BY_BZ_B$ system. The corresponding error kinematic equation is given by

$$\dot{\boldsymbol{\sigma}}_e = \boldsymbol{G}(\boldsymbol{\sigma}_e)\boldsymbol{\omega}_e \tag{11}$$

where $\boldsymbol{G}(\boldsymbol{\sigma}_e) = \frac{1}{2}(\boldsymbol{I}_{3\times3} + [\boldsymbol{\sigma}_e]^{\times} + \boldsymbol{\sigma}_e\boldsymbol{\sigma}_e^{\mathrm{T}} - \frac{1}{2}(1 + \boldsymbol{\sigma}_e^{\mathrm{T}}\boldsymbol{\sigma}_e)\boldsymbol{I}_{3\times3})$ with $\boldsymbol{I}_{3\times3} \in \mathbb{R}^{3\times3}$ denoting the identity matrix.

Remark 1. From Eq. (8), the control matrices of two inputs, $\boldsymbol{L}$ and $\boldsymbol{m}_{mag}$, depend on predictions of the space environment where the satellite is. Even with the state-of-art prediction models, there are still unknowns in the predicted aerodynamic torque and magnetic torque, such as NRLMSISE-00 atmospheric density model [38], Horizontal Wind Model 14 (HWM-14) [39], and International Geomagnetic Reference Field model (IGRF-13) [40]. Thus, previous works [12,14] have developed various control algorithms with robustness for the satellite to improve the attitude control performance when it encounters environmental torques.

3. PWA model in state-space form

This section aims to obtain a discrete linear model in state-space form for LMPC design. The PWA approximation is essentially to linearize the nonlinear attitude control equations. In principle, as long as the state region is properly partitioned, the PWA model can approach any nonlinear system using multiple linear models [29], even including discontinuous systems. We here offer an alternative description of the satellite attitude PWA model, which is based on error MRP rather than the usual error quaternion [20,21].

First, nonlinear terms in the error dynamics in Eq. (10) are reformed to be related to attitude parameters. Defining $\boldsymbol{L}$ as $\boldsymbol{u}_1 \in \mathbb{R}^{n_1}$, $\boldsymbol{m}_{mag}$ as $\boldsymbol{u}_2 \in \mathbb{R}^{n_2}$, it yields

$$(\boldsymbol{J}_B + \boldsymbol{J}_m)\dot{\boldsymbol{\omega}}_e = \boldsymbol{M}_1\boldsymbol{\omega}_e + \boldsymbol{M}_2 + \boldsymbol{M}_3\boldsymbol{u}_1 + \boldsymbol{M}_4\boldsymbol{u}_2 + \boldsymbol{d} \tag{12}$$

where

$$\begin{aligned}
&\boldsymbol{M}_1 = \left[(\boldsymbol{J}_B + \boldsymbol{J}_m)(\boldsymbol{\omega}_e + \boldsymbol{A}_e\boldsymbol{\omega}_d + \boldsymbol{A}_{bo}\boldsymbol{\omega}_{oi})\right]^\times + [\boldsymbol{A}_e\boldsymbol{\omega}_d + \boldsymbol{A}_{bo}\boldsymbol{\omega}_{oi}]^\times(\boldsymbol{J}_B + \boldsymbol{J}_m) - (\boldsymbol{J}_B + \boldsymbol{J}_m)\left([\boldsymbol{A}_e\boldsymbol{\omega}_d]^\times + [\boldsymbol{A}_{bo}\boldsymbol{\omega}_{oi}]^\times\right),\\
&\boldsymbol{M}_2 = -(\boldsymbol{J}_B + \boldsymbol{J}_m)\left(\boldsymbol{A}_e\dot{\boldsymbol{\omega}}_d + \boldsymbol{A}_e\boldsymbol{A}_d\dot{\boldsymbol{\omega}}_{oi} + [\boldsymbol{A}_{bo}\boldsymbol{\omega}_{oi}]^\times\boldsymbol{A}_e\boldsymbol{\omega}_d\right) - (\boldsymbol{A}_e\boldsymbol{\omega}_d + \boldsymbol{A}_e\boldsymbol{A}_d\boldsymbol{\omega}_{oi}) \times (\boldsymbol{J}_B + \boldsymbol{J}_m)(\boldsymbol{A}_e\boldsymbol{\omega}_d + \boldsymbol{A}_e\boldsymbol{A}_d\boldsymbol{\omega}_{oi}),\\
&\boldsymbol{M}_3 = \frac{2m_s}{m_B + 4m_s}[\boldsymbol{F}_a]^\times\boldsymbol{H}_m,\ \boldsymbol{M}_4 = -[\boldsymbol{B}_{mag}]^\times, \qquad \boldsymbol{d} = \boldsymbol{T}_a + \boldsymbol{T}_c + \boldsymbol{T}_e + \left(\boldsymbol{r}_P - \frac{m_B\boldsymbol{r}_B}{m_B + 4m_s}\right) \times \boldsymbol{F}_a
\end{aligned} \tag{13}$$

in which the disturbances given in $\boldsymbol{d}$ are partly known and partly unknown. Specifically, only the part of additional inertial torques $\boldsymbol{T}_a$ and $\boldsymbol{T}_c$ are known, and the other terms in $\boldsymbol{d}$ are unknown due to the existence of model uncertainties, $\boldsymbol{r}_P$, $\boldsymbol{r}_P$, and $\boldsymbol{T}_e$.

The continuous-time error kinematics and dynamics in Eqs. (11) and (12) are discretized with the sampling period Δt to be

$$\boldsymbol{\sigma}_e(k+1) = \boldsymbol{\sigma}_e(k) + \Delta t\boldsymbol{G}\big(\boldsymbol{\sigma}_e(k)\big)\boldsymbol{\omega}_e(k+1) \tag{14}$$

$$\begin{aligned}
\big(\boldsymbol{J}_B + \boldsymbol{J}_m(k)\big)\boldsymbol{\omega}_e(k+1) = {}& \big(\boldsymbol{J}_B + \boldsymbol{J}_m(k)\big)\boldsymbol{\omega}_e(k) + \Delta t\boldsymbol{M}_1(k)\boldsymbol{\omega}_e(k)\\
&+ \Delta t\boldsymbol{M}_2(k) + \Delta t\boldsymbol{M}_3(k)\boldsymbol{u}_1(k) + \Delta t\boldsymbol{M}_4(k)\boldsymbol{u}_2(k) + \Delta t\boldsymbol{d}(k)
\end{aligned} \tag{15}$$

Note that $\boldsymbol{M}_2$ is related to $\boldsymbol{\sigma}_e$ since it involves $\boldsymbol{A}_e$, and thus

$$\boldsymbol{M}_2(k) = \boldsymbol{M}_2(k)\frac{\boldsymbol{\sigma}_e(k)^{\mathrm{T}}}{\|\boldsymbol{\sigma}_e(k)\|^2}\boldsymbol{\sigma}_e(k) \tag{16}$$

Then, define a new state variable as $\boldsymbol{x}(k) = [\boldsymbol{\sigma}_e(k)^{\mathrm{T}}, \boldsymbol{\omega}_e(k)^{\mathrm{T}}]^{\mathrm{T}}$. The nonlinear attitude control model can be rewritten in state-space form as

$$\boldsymbol{x}(k+1) = \boldsymbol{A}(k)\boldsymbol{x}(k) + \boldsymbol{B}_{u1}(k)\boldsymbol{u}_1(k) + \boldsymbol{B}_{u2}(k)\boldsymbol{u}_2(k) + \boldsymbol{B}_d(k)\boldsymbol{d}(k) \tag{17}$$

where

$$\begin{aligned}
&\boldsymbol{A}(k) = \boldsymbol{M}_t^{-1}(k)\begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \Delta t\boldsymbol{M}_2(k)\frac{\boldsymbol{\sigma}_e(k)^{\mathrm{T}}}{\|\boldsymbol{\sigma}_e(k)\|^2} & \boldsymbol{J}_B + \boldsymbol{J}_m(k) + \Delta t\boldsymbol{M}_1(k) \end{bmatrix}, \qquad \boldsymbol{B}_{u1}(k) = \boldsymbol{M}_t^{-1}(k)\begin{bmatrix} \boldsymbol{0}_{3\times2} \\ \Delta t\boldsymbol{M}_3(k) \end{bmatrix},\\
&\boldsymbol{B}_{u2}(k) = \boldsymbol{M}_t^{-1}(k)\begin{bmatrix} \boldsymbol{0}_{3\times3} \\ \Delta t\boldsymbol{M}_4(k) \end{bmatrix}, \qquad \boldsymbol{B}_d(k) = \boldsymbol{M}_t^{-1}(k)\begin{bmatrix} \boldsymbol{0}_{3\times3} \\ \Delta t\boldsymbol{I}_{3\times3} \end{bmatrix}, \qquad \boldsymbol{M}_t(k) = \begin{bmatrix} \boldsymbol{I}_{3\times3} & -\Delta t\boldsymbol{G}(\boldsymbol{\sigma}_e(k)) \\ \boldsymbol{0}_{3\times3} & \boldsymbol{J}_B + \boldsymbol{J}_m(k) \end{bmatrix}
\end{aligned} \tag{18}$$

Note that $\boldsymbol{M}_t^{-1}(k)$ always exists and is given by

$$\boldsymbol{M}_t^{-1}(k) = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \Delta t\boldsymbol{G}(\boldsymbol{\sigma}_e(k))(\boldsymbol{J}_B + \boldsymbol{J}_m(k))^{-1} \\ \boldsymbol{0}_{3\times3} & (\boldsymbol{J}_B + \boldsymbol{J}_m(k))^{-1} \end{bmatrix} \tag{19}$$

Next is to transform the state-space equation in Eq. (17) into a PWA model. Consider that the state compact set $\boldsymbol{\Omega}_{\boldsymbol{x}} = \{\boldsymbol{x}(k) | \boldsymbol{x}(k) = [\boldsymbol{\sigma}_e(k)^{\mathrm{T}}, \boldsymbol{\omega}_e(k)^{\mathrm{T}}]^{\mathrm{T}}\}$ is partitioned into n subsets, that is, $\boldsymbol{\Omega}_{\boldsymbol{x}} = \{\boldsymbol{\Omega}_{\boldsymbol{x},1}, \boldsymbol{\Omega}_{\boldsymbol{x},2}, \cdots, \boldsymbol{\Omega}_{\boldsymbol{x},n}\}$, and a state variable $\boldsymbol{x}_{0,i} \in \boldsymbol{\Omega}_{\boldsymbol{x},i}$ is chosen as an approximation point. If the current state $\boldsymbol{x}(k)$ is in the subset $\boldsymbol{\Omega}_{\boldsymbol{x},i}$, the state and control matrices are computed by substituting $\boldsymbol{x}_{0,i}$ into $\boldsymbol{A}(k)$, $\boldsymbol{B}_{u1}(k)$, $\boldsymbol{B}_{u2}(k)$, and $\boldsymbol{B}_d(k)$. Then, we obtain a PWA model

$$\boldsymbol{x}(k+1) = \boldsymbol{A}(\boldsymbol{x}_{0,i})\boldsymbol{x}(k) + \boldsymbol{B}_{u1}(\boldsymbol{x}_{0,i})\boldsymbol{u}_1(k) + \boldsymbol{B}_{u2}(\boldsymbol{x}_{0,i})\boldsymbol{u}_2(k) + \boldsymbol{B}_d(\boldsymbol{x}_{0,i})\boldsymbol{d}(k), \quad \text{for } \boldsymbol{x}(k), \boldsymbol{x}_{0,i} \in \boldsymbol{\Omega}_{\boldsymbol{x},i} \tag{20}$$

Aiming to reduce static errors, the above PWA model is rewritten in incremental form as

$$\Delta\boldsymbol{x}(k+1) = \boldsymbol{A}(\boldsymbol{x}_{0,i})\Delta\boldsymbol{x}(k) + \boldsymbol{B}_{u1}(\boldsymbol{x}_{0,i})\Delta\boldsymbol{u}_1(k) + \boldsymbol{B}_{u2}(\boldsymbol{x}_{0,i})\Delta\boldsymbol{u}_2(k) + \boldsymbol{B}_d(\boldsymbol{x}_{0,i})\Delta\boldsymbol{d}(k), \quad \text{for } \boldsymbol{x}(k), \boldsymbol{x}_i^0 \in \boldsymbol{\Omega}_{\boldsymbol{x},i} \tag{21}$$

where, $\Delta\boldsymbol{x}(k+1) = \boldsymbol{x}(k+1) - \boldsymbol{x}(k)$; $\Delta\boldsymbol{u}_1(k) = \boldsymbol{u}_1(k) - \boldsymbol{u}_1(k-1)$; $\Delta\boldsymbol{u}_2(k) = \boldsymbol{u}_2(k) - \boldsymbol{u}_2(k-1)$; and, $\Delta\boldsymbol{d}(k) = \boldsymbol{d}(k) - \boldsymbol{d}(k-1)$.

Remark 2. Since the attitude dynamics in Eq. (10) is strongly nonlinear, the real-time Jacobian matrix in the conventional linearization method will be extremely complex [41]. It would impose a high real-time computational load, which is unsuitable for implementations, especially on small satellites. Hence, this paper adopts the PWA approximation, linearizing the model without computing the Jacobian matrix. Furthermore, the control matrices in the resulting PWA systems are more stable since they are computed using a given linearization state, rather than the measured state at each time step.

4. Controller design

From Eq. (7), additional inertial disturbances $\boldsymbol{T}_a$ and $\boldsymbol{T}_c$ are related to the acceleration and velocity of moving masses. Previous works [37] proposed a solution of reducing the update rate of MMD, but at the expense of the system's dynamic performance. Therefore, we develop a dual-rate PWA-MMPC algorithm for this two-input control system to achieve the asynchronous update of these two inputs. This algorithm is subsequently proved to be asymptotically stable.

4.1. Dual-rate PWA-MMPC

Since the inertial disturbances are affected by the acceleration and velocity of mass motions, the control frequency of the MMD is better to be appropriately lower. At the same time, to avoid a decrease in the system's dynamic performance, the magnetorquer can be controlled more frequently. Thus, we consider a control input order $(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_2, \cdots)$ with m steps in one control cycle. This implies that only after the magnetorquer is controlled $m-1$ times, the MMD will be controlled once, cooperating and called in sequence. With such an input order, these two inputs are updated at different rates, thereby achieving a balance of the inertial disturbances and the system's dynamic performance.

Define N_u and N_p as the control horizon and the prediction horizon with $N_p = (N_u - 1)m + 1$. At each time step, only the inputs to be updated will be implemented with the other inputs considered as known. The indexing function of the input to be updated is defined as $\varepsilon(k) = (k \bmod m) + 1$. Then, the N-step prediction model at time k for the system described by Eq. (21) is

$$\boldsymbol{X}_{k+1|k} = \mathcal{I}\boldsymbol{x}_{k|k} + \boldsymbol{S}_{x,k}\Delta\boldsymbol{x}_{k|k} + \boldsymbol{S}_{u,k}\Delta\boldsymbol{U}_k + \boldsymbol{S}_{d,k}\Delta\boldsymbol{D}_k \tag{22}$$

where

$$\boldsymbol{X}_{k+1|k} = \begin{bmatrix} \boldsymbol{x}_{k+1|k} \\ \boldsymbol{x}_{k+2|k} \\ \vdots \\ \boldsymbol{x}_{k+N_p|k} \end{bmatrix}_{N_p\times 1}, \quad \Delta\boldsymbol{U}_k = \begin{bmatrix} \Delta\boldsymbol{u}_{k|k} \\ \Delta\boldsymbol{u}_{k+1|k} \\ \vdots \\ \Delta\boldsymbol{u}_{k+N_p-1|k} \end{bmatrix}_{N_p\times 1}, \quad \Delta\boldsymbol{D}_k = \begin{bmatrix} \Delta\boldsymbol{d}_{k|k} \\ \Delta\boldsymbol{d}_{k+1|k} \\ \vdots \\ \Delta\boldsymbol{d}_{k+N_p-1|k} \end{bmatrix}_{N_p\times 1},$$

$$\boldsymbol{S}_{x,k} = \begin{bmatrix} \boldsymbol{A}_k \\ \boldsymbol{A}_k^2 + \boldsymbol{A}_k \\ \vdots \\ \sum_{i=1}^{N_p} \boldsymbol{A}_k^i \end{bmatrix}_{N_p\times 1}, \quad \boldsymbol{S}_{u,k} = \begin{bmatrix} \boldsymbol{B}_{u,k}^{\varepsilon(k)} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \sum_{i=1}^{2} \boldsymbol{A}_k^{i-1}\boldsymbol{B}_{u,k}^{\varepsilon(k)} & \boldsymbol{B}_{u,k}^{\varepsilon(k+1)} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N_p} \boldsymbol{A}_k^{i-1}\boldsymbol{B}_{u,k}^{\varepsilon(k)} & \sum_{i=1}^{N_p-1} \boldsymbol{A}_k^{i-1}\boldsymbol{B}_{u,k}^{\varepsilon(k+1)} & \cdots & \boldsymbol{B}_{u,k}^{\varepsilon(k+N_p-1)} \end{bmatrix}_{N_p\times N_p},$$

$$\boldsymbol{S}_{d,k} = \begin{bmatrix} \boldsymbol{B}_{d,k}^{\varepsilon(k)} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \sum_{i=1}^{2} \boldsymbol{A}_k^{i-1}\boldsymbol{B}_{d,k}^{\varepsilon(k)} & \boldsymbol{B}_{d,k}^{\varepsilon(k+1)} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N_p} \boldsymbol{A}_k^{i-1}\boldsymbol{B}_{d,k}^{\varepsilon(k)} & \sum_{i=1}^{N_p-1} \boldsymbol{A}_k^{i-1}\boldsymbol{B}_{d,k}^{\varepsilon(k+1)} & \cdots & \boldsymbol{B}_{d,k}^{\varepsilon(k+N_p-1)} \end{bmatrix}_{N_p\times N_p}, \quad \mathcal{I} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{I} \\ \vdots \\ \boldsymbol{I} \end{bmatrix}_{N_p\times 1} \tag{23}$$

in which $\Delta\boldsymbol{x}_{k+i|k}$, $\Delta\boldsymbol{u}_{k+i|k}$, and $\Delta\boldsymbol{d}_{k+i|k}$ denote the state increment, the input increment, and the known disturbance increment at time $k+i$ predicted at time k, respectively. Note that $\Delta\boldsymbol{d}_{k+i|k}$ only includes the increment of the known part in $\boldsymbol{d}$, that is, $\boldsymbol{T}_a$ and $\boldsymbol{T}_c$.

For the problem of regulation to the origin, this dual-rate PWA-MMPC is to minimize the following cost function

$$J_k = p(\boldsymbol{x}_{k+N_p|k}) + \sum_{i=0}^{N_p-1} \left[q(\boldsymbol{x}_{k+i|k}) + r(\Delta\boldsymbol{u}_{k+i|k})\right] \tag{24}$$

with

$$\begin{cases} q(\boldsymbol{x}_{k+i|k}) = \boldsymbol{x}_{k+i|k}^{\mathrm{T}}\boldsymbol{Q}^{\varepsilon(k+i)}\boldsymbol{x}_{k+i|k} \\ r(\boldsymbol{u}_{k+i|k}) = \Delta\boldsymbol{u}_{k+i|k}^{\mathrm{T}}\boldsymbol{R}^{\varepsilon(k+i)}\Delta\boldsymbol{u}_{k+i|k} \\ p(\boldsymbol{x}_{k+N_p|k}) = \boldsymbol{x}_{k+N_p|k}^{\mathrm{T}}\boldsymbol{P}_k\boldsymbol{x}_{k+N_p|k} \end{cases} \tag{25}$$

where $\boldsymbol{Q}^{\varepsilon(k+i)}$ and $\boldsymbol{R}^{\varepsilon(k+i)}$ are weight matrices on state and input for each input; $\boldsymbol{P}_k$ is the variant weight matrix on final state.

Then, this dual-rate PWA-MMPC solves the following finite horizon constrained control problem:

$$\begin{aligned} \min \quad & J_k(\boldsymbol{x}_{k|k}, \Delta \boldsymbol{U}_k) \\ \text{wrt} \quad & \Delta \boldsymbol{U}_k \\ \text{s.t.} \quad & \boldsymbol{u}_{k+i|k} \in \mathrm{U} \\ & \boldsymbol{x}_{k+i|k} \in X, \ (i = 1, 2, \cdots, N_p - 1) \\ & \boldsymbol{x}_{k+N_p|k} \in X_f \\ & \Delta \boldsymbol{x}_{k+i+1|k} = \boldsymbol{A}_k \Delta \boldsymbol{x}_{k+i|k} + \boldsymbol{B}_{u,k}^{\varepsilon(k+i)} \Delta \boldsymbol{u}_{k+i|k} + \boldsymbol{B}_{d,k}^{\varepsilon(k+i)} \Delta \boldsymbol{d}_{k+i|k} \end{aligned} \tag{26}$$

where X_f denotes the terminal state set, and $\boldsymbol{x}_{k+N_p|k} \in X_f$ is required for stability guarantees.

If this optimization problem is convex, this constrained linear optimization problem can still be transformed into a tractable quadratic program (QP) problem. The QP problem requires input constraints to be transformed to the form of

$$\begin{bmatrix} \boldsymbol{T}_{mpc} \\ -\boldsymbol{T}_{mpc} \end{bmatrix} \Delta \boldsymbol{U}_k \leq \begin{bmatrix} \Delta \boldsymbol{u}_{k|k,\max} \\ \vdots \\ \Delta \boldsymbol{u}_{k+N_p-1|k,\max} \\ -\Delta \boldsymbol{u}_{k|k,\min} \\ \vdots \\ -\Delta \boldsymbol{u}_{k+N_p-1|k,\min} \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{L}_{mpc} \\ -\boldsymbol{L}_{mpc} \end{bmatrix} \Delta \boldsymbol{U}_k \leq \begin{bmatrix} \boldsymbol{u}_{k|k,\max} - \boldsymbol{u}_{k-1|k} \\ \vdots \\ \boldsymbol{u}_{k+N_p-1|k,\max} - \boldsymbol{u}_{k-1|k} \\ \boldsymbol{u}_{k-1|k} - \boldsymbol{u}_{k|k,\min} \\ \vdots \\ \boldsymbol{u}_{k-1|k} - \boldsymbol{u}_{k+N_p-1|k,\min} \end{bmatrix} \tag{27}$$

where, $\Delta \boldsymbol{u}_{k|k,\min}, \cdots, \Delta \boldsymbol{u}_{k+N_p-1|k,\min}$ and $\Delta \boldsymbol{u}_{k|k,\max}, \cdots, \Delta \boldsymbol{u}_{k+N_p-1|k,\max}$ are limitations of input rate at each time; $\boldsymbol{u}_{k|k,\min}, \cdots, \boldsymbol{u}_{k+N_p-1|k,\min}$ and $\boldsymbol{u}_{k|k,\max}, \cdots, \boldsymbol{u}_{k+N_p-1|k,\max}$ are limitations of input at each time; and, $\boldsymbol{T}_{mpc}$ and $\boldsymbol{L}_{mpc}$ are input constraint matrices.

Note that, compared with conventional LMPC, input constraint matrices no longer remain constant, but depend on the index $\varepsilon(k)$. Fig. 3 illustrates how to determine these two input constraint matrices.

Given the initial state $\boldsymbol{x}_{k|k}$, if the optimization problem is feasible, it will yield a set of optimal controls $\Delta \boldsymbol{U}_k^* = [\Delta \boldsymbol{u}_{k|k}^{*\mathrm{T}}, \Delta \boldsymbol{u}_{k+1|k}^{*\mathrm{T}}, \cdots, \Delta \boldsymbol{u}_{k+N_p-1|k}^{*\mathrm{T}}]^{\mathrm{T}}$, an optimal predicted-state trajectory $\boldsymbol{X}_{k+1|k}^* = [\boldsymbol{x}_{k|k}^{*\mathrm{T}}, \cdots, \boldsymbol{x}_{k+N_p|k}^{*\mathrm{T}}]^{\mathrm{T}}$, and the optimized cost function $J_k^*(\boldsymbol{x}_{k|k}, \Delta \boldsymbol{U}_k^*)$. Then, the first element $\Delta \boldsymbol{u}_{k|k}^*$ will be implemented for the actual plant.

Remark 3. The MMPC in [33] optimizes only over the input about to be moved. Those inputs which have already been planned before the current time are assumed to be executed as planned, that is, $\Delta \boldsymbol{u}_{k+i|k} = \Delta \boldsymbol{u}_{k+i|k-1}$. Essentially, the MMPC in [33] and the propose MMPC have the same idea of updating one input at a time and only differ in the prediction strategy.

4.2. Stability analysis of dual-rate PWA-MMPC

To ensure the closed-loop stability of the proposed dual-rate PWA-MMPC algorithm, we establish several additional conditions and carefully parameterize the terminal cost through its weight matrix $\boldsymbol{P}_k$. The lemma of the uniform asymptotic stability for discrete-time systems is given here.

Lemma 1 *([42]: uniform asymptotic stability for discrete-time systems). Given the closed-loop system Eq. (22), with* $\mathrm{X} \in \mathbb{R}^n$ *as the state constraint set containing the origin as its equilibrium, suppose there exists a continuous Lyapunov function* $J(\boldsymbol{x}_{k|k}) : \mathbb{R}^n \times \mathbb{Z}_+ \to \mathbb{R}_+$ *for* $\boldsymbol{x}_{k|k} \in X$*, such that 1)* $J(0) = 0$ *and* $J(\boldsymbol{x}_{k|k}) > 0$ *when* $\boldsymbol{x}_{k|k} \neq 0$*, and 2)* $J(\boldsymbol{x}_{k+1|k+1}) - J(\boldsymbol{x}_{k|k}) \leq -\alpha \boldsymbol{x}_{k|k}$*, where* $\alpha(\cdot)$ *denotes a positive definite function. Then, the origin is uniformly asymptotically stable.*

Before addressing the stability issue, two assumptions are made here.

Assumption 1. There exists a terminal controller $\boldsymbol{u}_f^{\varepsilon(k)} \in U$ such that $\boldsymbol{x}_{k+N_p|k} \in X_f$ and $\boldsymbol{A}_k \boldsymbol{x}_{k+N_p|k} + \boldsymbol{B}_{u,k}^{\varepsilon(k+N_p)} \boldsymbol{u}_f^{\varepsilon(k)} \in X_f$. Consider that the optimized control sequence at time k is modified by omitting the initial term $\Delta \boldsymbol{u}_{k|k}^*$ and adding the terminal controller $\Delta \boldsymbol{u}_f^{\varepsilon(k)}$ in the end of the prediction horizon. If this new control sequence is implemented on the subsequent PWA system at time $k+1$, it is always satisfied that $\boldsymbol{x}_{k+1+i|k+1} \in X$ for $i = 0, 1, \cdots, N_p - 1$.

This assumption guarantees the persistent feasibility of the dual-rate PWA-MMPC, which implies that the optimal control problem is always feasible at its finite horizon [21].

Considering that the prediction model is time-varying, attention will be paid to the distance between two subsequent PWA systems of $(\boldsymbol{A}_k, \boldsymbol{B}_{u,k}^{\varepsilon(k)})$ and $(\boldsymbol{A}_{k+1}, \boldsymbol{B}_{u,k+1}^{\varepsilon(k+1)})$.

Assumption 2. As the PWA system evolves over time, the difference between two subsequent systems at time k and $k+1$ in terms of a bound on the state error should be quantified. If $\boldsymbol{u}_{k+i|k}, \boldsymbol{u}_{k+1+i|k+1} \in U$ for $i = 0, 1, \cdots, N_p - 1$, the variation of the PWA system over the time interval Δt satisfies that the change of predicted states is bounded. That is,

$$\left\| \boldsymbol{Q}^{\varepsilon(k+1)\frac{1}{2}} \boldsymbol{x}_{k+i+1|k+1} \right\|^2 = \left\| \boldsymbol{Q}^{\varepsilon(k)\frac{1}{2}} \boldsymbol{x}_{k+i+1|k} \right\|^2 + \delta_i \| \boldsymbol{x}_{k+N_p|k} \|^2, \quad i = 0, 1, \cdots, N_p \tag{28}$$

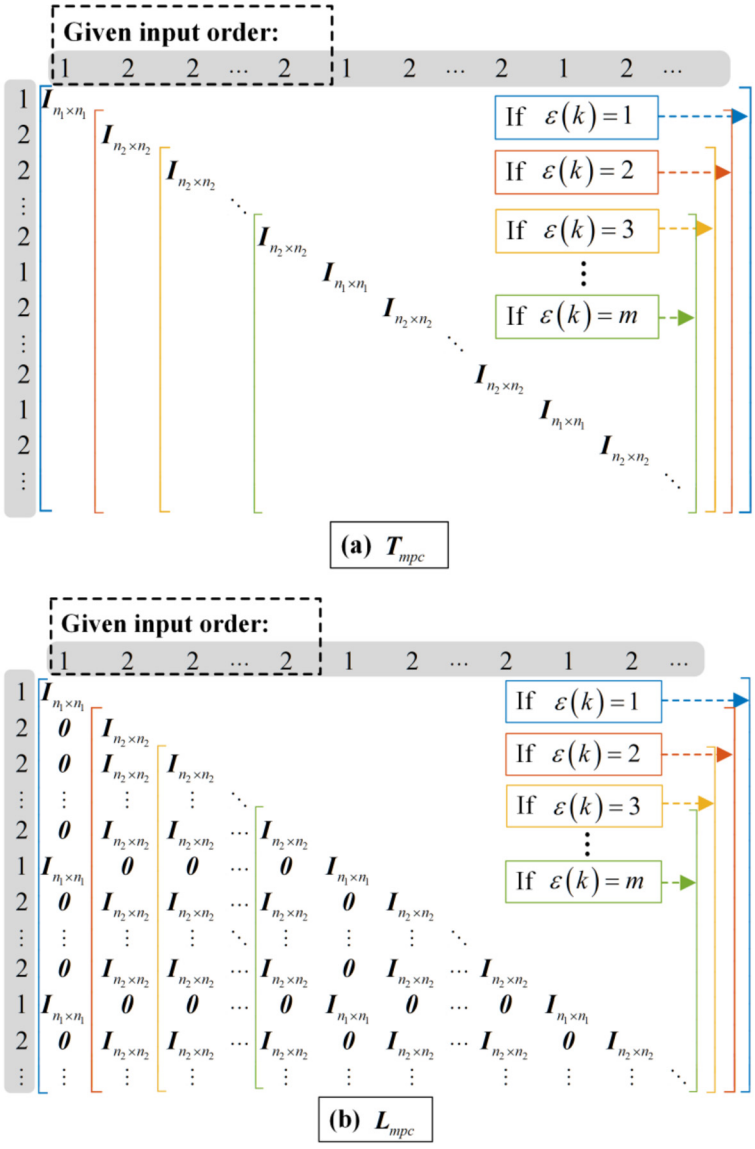

Fig. 3. Input constraint matrices $\boldsymbol{T}_{mpc}$ and $\boldsymbol{L}_{mpc}$.

$$\left\| \boldsymbol{P}_{k+1}^{\frac{1}{2}} \boldsymbol{x}_{k+N_p+1|k+1} \right\|^2 = \left\| \boldsymbol{P}_{k}^{\frac{1}{2}} \boldsymbol{x}_{k+N_p+1|k} \right\|^2 + \delta_{N_p} \| \boldsymbol{x}_{k+N_p|k} \|^2 \tag{29}$$

Using above two equations, the difference between two predicted states at time k and $k+1$ can be stated as

$$\begin{aligned} \sum_{i=0}^{N_p-1} \left[q(\boldsymbol{x}_{k+1+i|k+1}) - q(\boldsymbol{x}_{k+i|k}) \right] &= \sum_{i=0}^{N_p-1} \left[q(\boldsymbol{x}_{k+1+i|k+1}) - q(\boldsymbol{x}_{k+i+1|k}) \right] - q(\boldsymbol{x}_{k|k}) + q(\boldsymbol{x}_{k+N_p|k}) \\ &= \sum_{i=0}^{N_p-1} \delta_i \| \boldsymbol{x}_{k+N_p|k} \|^2 - q(\boldsymbol{x}_{k|k}) + q(\boldsymbol{x}_{k+N_p|k}) \end{aligned} \tag{30}$$

$$p(\boldsymbol{x}_{k+1+N_p|k+1}) - p(\boldsymbol{x}_{k+N_p+1|k}) = \delta_{N_p} \| \boldsymbol{x}_{k+N_p|k} \|^2 \tag{31}$$

Then, the closed-loop stability of this dual-rate PWA-MMPC algorithm is given by the following theorem.

Theorem 1. *Consider the PWA system Eq. (22) subject to constraints in Eq. (26). Suppose that assumptions in Lemma 1, Assumption 1, and Assumption 2 are all satisfied. If the set of initial conditions $\boldsymbol{x}_{k|k}$ is feasible in N_p steps and the terminal cost function $p(\cdot)$ satisfies*

$$p(\boldsymbol{x}_{k+N_p+1|k}) + \delta_m \| \boldsymbol{x}_{k+N_p|k} \|^2 + q(\boldsymbol{x}_{k+N_p|k}) - p(\boldsymbol{x}_{k+N_p|k}) + r\left(\Delta \boldsymbol{u}_f^{\varepsilon(k+1+N_p)}\right) \leq 0 \tag{32}$$

in which $\delta_m = \max\{|\delta_0|, |\delta_1|, \cdots, |\delta_{N_p}|\}$, then the closed-loop system Eq. (22) with the optimal control $\Delta \boldsymbol{u}_{k|k}^*$ is asymptotically stable.

Proof. To find an upper bound on J^*_{k+1}, the optimized input sequence $\Delta \boldsymbol{U}^*_k$ at time k is used to organize a candidate input sequence $\Delta \boldsymbol{U}_{k+1}$ imposed to the PWA system at time $k+1$ as

$$\Delta \boldsymbol{U}_{k+1} = \left[\Delta \boldsymbol{u}^{*\mathrm{T}}_{k+1|k}, \Delta \boldsymbol{u}^{*\mathrm{T}}_{k+2|k}, \cdots, \Delta \boldsymbol{u}^{*\mathrm{T}}_{k+N_p-1|k}, \Delta \boldsymbol{u}_f^{\varepsilon(k+N_p)}\right]^{\mathrm{T}} \tag{33}$$

Note that, since two systems are sampled at time k and $k+1$, their trajectories generated by $\boldsymbol{U}_k$ do not overlap. Therefore, $\boldsymbol{U}_{k+1}$ is just feasible, but not optimal, and thus, it yields

$$J^*_{k+1}\left(\boldsymbol{x}_{k+1|k+1}, \Delta \boldsymbol{U}^*_{k+1}\right) \le J_{k+1}(\boldsymbol{x}_{k+1|k+1}, \Delta \boldsymbol{U}_{k+1}) \tag{34}$$

with

$$\begin{aligned} &J_{k+1}(\boldsymbol{x}_{k+1|k+1}, \Delta \boldsymbol{U}_{k+1}) - J^*_k\left(\boldsymbol{x}_{k|k}, \Delta \boldsymbol{U}^*_k\right) \\ &\quad = p(\boldsymbol{x}_{k+1+N_p|k+1}) - p(\boldsymbol{x}_{k+N_p|k}) + \sum_{i=0}^{N_p-1}\left[q(\boldsymbol{x}_{k+1+i|k+1}) - q(\boldsymbol{x}_{k+i|k})\right] + r(\Delta \boldsymbol{u}_f^{\varepsilon(k+1)}) - r\left(\Delta \boldsymbol{u}^*_{k|k}\right) \end{aligned} \tag{35}$$

Substituting Eqs. (30), (31), (32) and (35) into Eq. (34), the following is obtained:

$$J^*_{k+1}\left(\boldsymbol{x}_{k+1|k+1}, \Delta \boldsymbol{U}^*_{k+1}\right) - J^*_k\left(\boldsymbol{x}_{k|k}, \Delta \boldsymbol{U}^*_k\right) \le -q(\boldsymbol{x}_{k|k}) - r\left(\Delta \boldsymbol{u}^*_{k|k}\right) \tag{36}$$

The weight matrices $\boldsymbol{Q}^{\varepsilon(k+i)}$ and $\boldsymbol{R}^{\varepsilon(k+i)}$ ensure that $J^*_k(\boldsymbol{x}_{k|k}, \Delta \boldsymbol{U}^*_k)$ is monotonically decreasing. The cost function can be considered as a Lyapunov function, and thus, from Lemma 1, the closed-loop system Eq. (22) with the optimal control $\boldsymbol{u}^*_{k|k}$ is asymptotically stable.

Next is to find an appropriate terminal cost $\boldsymbol{P}_k$ that satisfies the inequality in Eq. (32). Consider the infinite horizon LQ optimal controller $\Delta \boldsymbol{u}_f^{\varepsilon(k+N_p)} = -\boldsymbol{K}_\infty^{\varepsilon(k+N_p)} \boldsymbol{x}_{k+N_p|k}$ as a linear periodic terminal input after the prediction horizon. It will not be executed but be used for the stability analysis. Under the assumption 1, it is satisfied that

$$\boldsymbol{x}_{k+N_p+1|k} = \boldsymbol{A}_k \boldsymbol{x}_{k+N_p|k} + \boldsymbol{B}_{u,k}^{\varepsilon(k+N_p)} \boldsymbol{u}_f^{\varepsilon(k+N_p)} = \left(\boldsymbol{A}_k - \boldsymbol{B}_{u,k}^{\varepsilon(k+N_p)} \boldsymbol{K}_\infty^{\varepsilon(k+N_p)}\right) \boldsymbol{x}_{k+N_p|k} = \boldsymbol{A}_k^{\varepsilon(k+N_p)} \boldsymbol{x}_{k+N_p|k} \in X_f \tag{37}$$

The coefficient matrix is computed via

$$\boldsymbol{K}_\infty^{\varepsilon(k+N_p)} = -\left(\boldsymbol{B}_{u,k}^{\varepsilon(k+N_p)} \boldsymbol{P}_\infty^{\varepsilon(k+N_p)} \boldsymbol{B}_{u,k}^{\varepsilon(k+N_p)} + \boldsymbol{R}^{\varepsilon(k+N_p)}\right)^{-1} \boldsymbol{B}_{u,k}^{\varepsilon(k+N_p)} \boldsymbol{P}_\infty^{\varepsilon(k+N_p)} \boldsymbol{A}_k \tag{38}$$

where $\boldsymbol{P}_\infty^{\varepsilon(k+N_p)}$ satisfies the discrete-time algebraic Riccati equation (ARE):

$$\begin{aligned} \boldsymbol{P}_\infty^{\varepsilon(k+N_p)} &= \boldsymbol{Q}^{\varepsilon(k+N_p)} + \boldsymbol{A}_k^{\mathrm{T}} \boldsymbol{P}_\infty^{\varepsilon(k+N_p)} \boldsymbol{A}_k \\ &\quad - \boldsymbol{A}_k^{\mathrm{T}} \boldsymbol{P}_\infty^{\varepsilon(k+N_p)} \boldsymbol{B}_{u,k}^{\varepsilon(k+N_p)} \left(\boldsymbol{R}^{\varepsilon(k+N_p)} + \boldsymbol{B}_{u,k}^{\varepsilon(k+N_p)\mathrm{T}} \boldsymbol{P}_\infty^{\varepsilon(k+N_p)} \boldsymbol{B}_{u,k}^{\varepsilon(k+N_p)}\right)^{-1} \boldsymbol{B}_{u,k}^{\varepsilon(k+N_p)\mathrm{T}} \boldsymbol{P}_\infty^{\varepsilon(k+N_p)} \boldsymbol{A}_k \end{aligned} \tag{39}$$

Substituting the terminal closed-loop system into Eq. (32), the following is obtained:

$$\boldsymbol{x}^{\mathrm{T}}_{k+N_p|k}\left[\boldsymbol{A}_k^{\varepsilon(k+N_p)\mathrm{T}} \boldsymbol{P}_k \boldsymbol{A}_k^{\varepsilon(k+N_p)} + \boldsymbol{Q}^{\varepsilon(k+N_p)} - \boldsymbol{P}_k + \delta_m \boldsymbol{I} + \boldsymbol{K}_\infty^{\varepsilon(k+N_p)\mathrm{T}} \boldsymbol{R}^{\varepsilon(k+N_p)} \boldsymbol{K}_\infty^{\varepsilon(k+N_p)}\right] \boldsymbol{x}_{k+N_p|k} \le 0 \tag{40}$$

By solving the following discrete Lyapunov equation, it yields:

$$\boldsymbol{A}_k^{\varepsilon(k+N_p)\mathrm{T}} \boldsymbol{P}_k \boldsymbol{A}_k^{\varepsilon(k+N_p)} + \boldsymbol{Q}^{\varepsilon(k+N_p)} - \boldsymbol{P}_k + \delta_m \boldsymbol{I} + \boldsymbol{K}_\infty^{\varepsilon(k+N_p)\mathrm{T}} \boldsymbol{R}^{\varepsilon(k+N_p)} \boldsymbol{K}_\infty^{\varepsilon(k+N_p)} = \boldsymbol{0} \tag{41}$$

Thus, the desired weighting matrix $\boldsymbol{P}_k$ can be computed using Eq. (41). As the closed-loop stability analysis is given above, the error state will converge to the terminal region X_f in finite time.

Compared with the nominal MMPC algorithm in [33], implementing the proposed algorithm has to update the control matrices and determine the weighting matrix before computing the MPC algorithm at each time step. For clarity, the pseudo code of this dual-rate PWA-MMPC is presented in Table 1.

Remark 4. The parameter δ_m does not require to be chosen too conservatively to avoid the significant degradation of the controller's performance. The situation of $\delta_m < \delta_i$ only occurs occasionally with extremely bad conditions of the initial state, and even if it occurs, the current input can be considered an unknown disturbance and will not affect the state convergence. Besides, this stability analysis does not depend on the optimality of this initial solution, only its feasibility.

Remark 5. The computational complexity of typical MPC algorithm is $O((n_{ctrl} \times N_{ctrl})^3)$ where n_{ctrl} is the dimension of the input and N_{ctrl} is the control horizon length [33]. Thus, the strategy of updating only one input at a time significantly reduces the optimization complexity. By redefining the indexing function $\varepsilon(k)$ appropriately, this algorithm can actually be generalized to various multi-input MPC schemes with a fixed periodic input order.

5. Hardware-in-the-loop simulation

In this section, hardware-in-the-loop simulations are conducted to demonstrate the effectiveness of the proposed methodology for the attitude control. This control algorithm is supposed to be applied in a LEO 2U CubeSat, called NJUST-2, as shown in Fig. 4. It is a part of the QB50 project and was deployed from the International Space Station (ISS) on 25 May 2017.

Orbital parameters in the simulation are propagated using actual Two Line Elements (TLE) observed at 03:36:24.669 in 26 May 2019 (UTCG) and listed in Table 2.

Table 1
Pseudo code of dual-rate PWA-MMPC.

Attitude control algorithm based on dual-rate PWA-MMPC	
1.	**Initialization:** Partition the state compact set $\boldsymbol{\Omega}_x$ into n subsets by Eq. (20); Give the input order and the feasible input sequences for all inputs according to Eq. (26); Obtain δ_m via sampling method; Set initial time k_0 and initial input
2.	**While** (mission not over)
3.	Measure the attitude, obtain $\boldsymbol{\sigma}_{bo}$, $\boldsymbol{\omega}_{bo}$
4.	Compute the states $\boldsymbol{\sigma}_e$, $\boldsymbol{\omega}_e$, and determine the linearization point $\boldsymbol{x}_{0,i}$
5.	Compute control matrices $\boldsymbol{A}_k$, $\boldsymbol{B}_{u,k}^{\varepsilon(k)}$, $\boldsymbol{B}_{d,k}^{\varepsilon(k)}$ by Eq. (18)
6.	Solve the LQ problem to obtain $\boldsymbol{K}_{\infty}^{\varepsilon(k+N_p)}$, $\boldsymbol{A}_k^{\varepsilon(k+N_p)}$ by Eqs. (37), (38)
7.	Compute $\boldsymbol{P}_k$ by Eq. (41)
8.	Prepare the matrices $\boldsymbol{S}_{x,k}$, $\boldsymbol{S}_{u,k}$, $\boldsymbol{S}_{d,k}$, $\mathcal{I}$ by Eq. (23)
9.	Solve the MPC optimization problem in Eq. (26), obtain input sequence
10.	Implement the first part of input sequence on the satellite
11.	Pause for one time step, $k = k + 1$, move to next input according to the given order
12.	**End while**

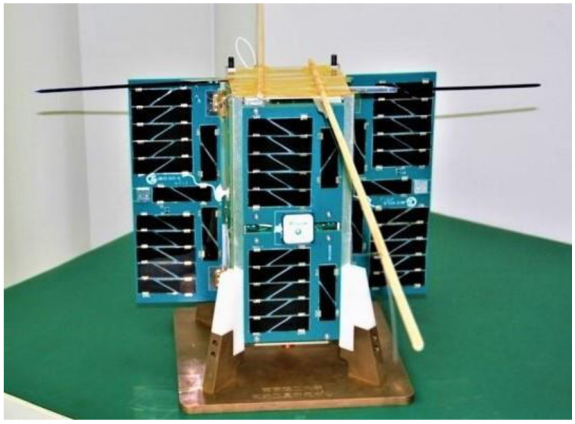

Fig. 4. A LEO CubeSat called NJUST-2.

Table 2
Orbital parameters.

Parameters	Values
Orbit height	300 km
Inclination	51.705°
Eccentricity	0
RAAN	31.437°
UTC time	26 May 2019 12:00:00.000
Initial latitude	−17.414°
Initial longitude	−46.432°

5.1. Hardware-in-the-loop simulation platform

This hardware-in-the-loop simulation platform involves a prototype design of MMD with doubly symmetric mass distribution, as shown in Fig. 5(a). These masses are driven by four linear DC-servomotors from FAULHABER LM0830-040-01 series. According to its technical manual, its movement can be approximated as half distance with the uniform acceleration and the other half with the uniform deceleration. Each moving mass is 0.1 kg in this design and can be actuated within ±0.04 m. As presented in Fig. 5(b), this simulation uses a three-axis magnetorquer completely designed and manufactured by Nanjing University of Science and Technology (NJUST). Its maximum magnetic moment is up to $0.1\mathrm{A}\cdot\ \mathrm{m}^2$. Moreover, the control system for these two actuators is shown in Fig. 5(c). The ADCS CPU running in the C environment is responsible for computing the control laws using attitude measurements and communicating with two actuators.

This control platform for the actuators is connected to an attitude-orbit coupling dynamic simulation system shown in Fig. 6. This simulation system is built by Micro/Nano Satellite Research Center of NJUST, composed of real-time dynamics simulation system, measurement simulation system, orbit information displayer, and attitude information displayer. Note that, in this simulation, the ADCS CPU, the MMD, and the three-axis magnetorquer are fully physically accessed. Thus, there exists an unknown error between the desired torque computed by the control algorithm and the measured torque, resulting in model uncertainties. The procedure of this hardware-in-the-loop simulation is described in Fig. 7.

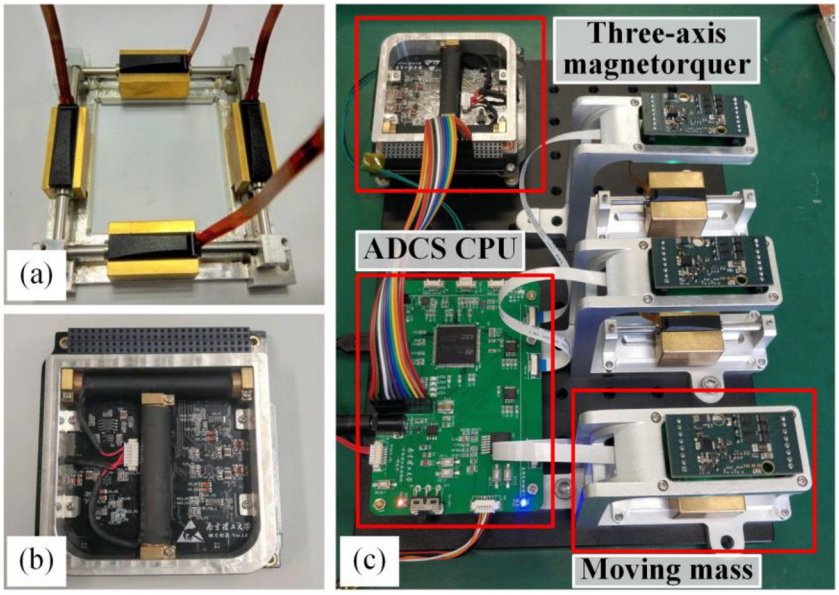

Fig. 5. The control system for two actuators. (a) MMD. (b) Three-axis magnetorquer. (c) The control system.

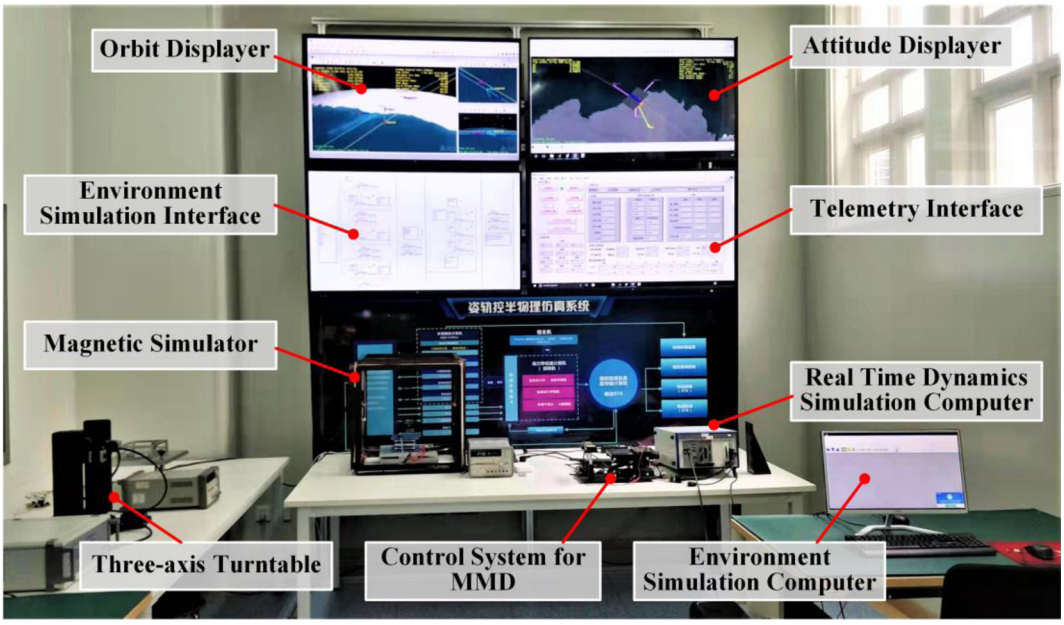

Fig. 6. The attitude-orbit coupling dynamic simulation system.

5.2. Simulation results

In this simulation, the NRLMSISE-00 Atmospheric Model [38] and the Horizontal Wind Model [39] (HWM 14) are used to build the true aerodynamic environment running in the dynamics simulation computer. Whereas, when computing control laws, it is assumed that the aerodynamic force is constant as its estimated average, also resulting in model uncertainties. Besides, the International Geomagnetic Reference Field [40] (IGRF-13) model is used to provide the true geomagnetic field information. Associated simulation parameters are listed in Table 3.

In the PWA system, the state compact set of MRP is partitioned by the interval of 0.04 and the state compact set of angular velocity by the interval of 0.001. This simulation uses Euler angles to give the initial and desired conditions. The roll, pitch, and yaw angles are defined as φ, θ, and ψ, respectively. The satellite attitude described by Euler angles follows the $\psi - \varphi - \theta$ order. Controller parameters are listed in Table 4.

The dynamics updating step size is set to 0.2 s, and the sampling period Δt is set to 2 s. The simulation results are reported in Figs. 8∼12.

In this simulation, the synchronous nonlinear MPC algorithm in [33] and the synchronous nonlinear MPC algorithm in [43] are employed in this PWA system for comparison, namely PWA-SLMPC and PWA-SNMPC. Unlike the proposed dual-rate PWA-MMPC, these two algorithms update two inputs synchronously at each time step. The PWA system used by PWA-SNMPC at each prediction time step will vary with the prediction state, that is, PWA-SNMPC uses a nonlinear model to predict the state.

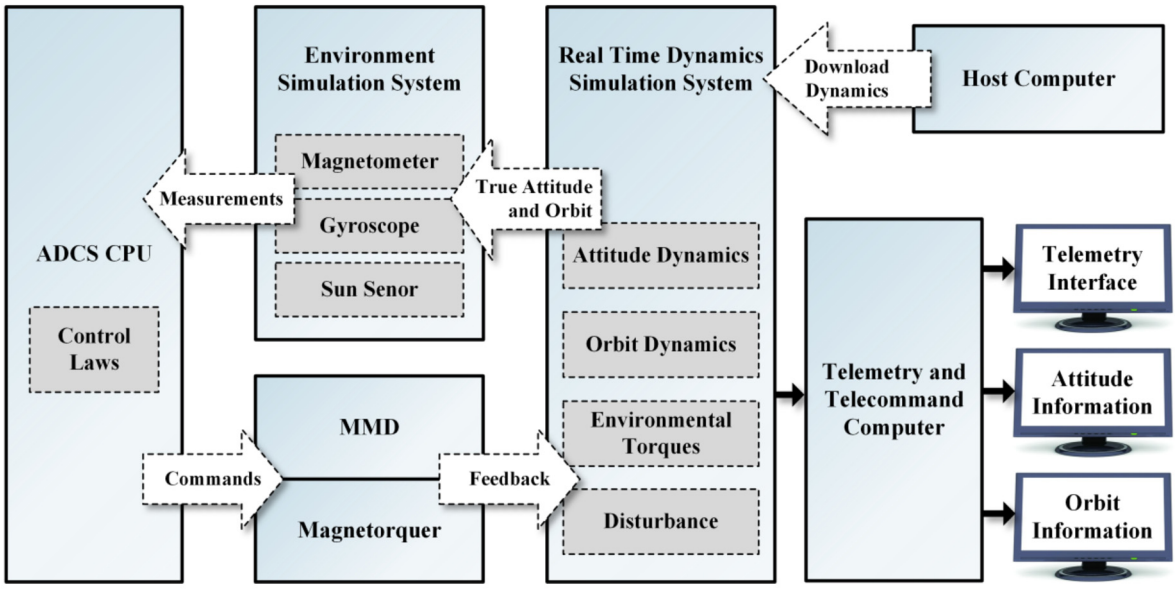

Fig. 7. Flow diagram of the hardware-in-the-loop simulation.

Table 3
Simulation parameters.

Parameters	Values
Mass	$m_B = 2$ kg, $m_s = 0.1$ kg
Moment of inertia	$\boldsymbol{J}_B = \text{diag}(3, 8, 8) \times 10^{-3}$ kg $\cdot$ m^2
Satellite size	$110 \times 110 \times 220$ mm
Body CoM vector	$\boldsymbol{r}_B = [4, 1, 3]^{\mathrm{T}} \times 10^{-3}$ m
Satellite CoP vector	$\boldsymbol{r}_P = [-2, 1, -1]^{\mathrm{T}} \times 10^{-3}$ m
Unknown disturbance	$\boldsymbol{T}_e = [0.5, -1.5, 1]^{\mathrm{T}} \times 10^{-7}$ Nm
Constant deviation	$\delta = 0.04$ m
Atmospheric Model	Average F10.7: 140, Daily F10.7: 140,
	Geomagnetic index: 4
HWM 14	Apindex: 80, Model: total

Table 4
Controller parameters.

Parameters	Values
Initial and desired Euler angle ($\psi \ -\varphi \ -\theta$ order)	$[\varphi_0, \theta_0, \psi_0] = [10°, -20°, 15°]$, $[\varphi_d, \theta_d, \psi_d] = [-5°, 5°, 0°]$
Initial and desired angular velocity	$\boldsymbol{\omega}_{bo,0} = [-2, -1, -2]^{\mathrm{T}} \times 10^{-3}$ rad/s, $\boldsymbol{\omega}_{bo,d} = [0, 0, 0]^{\mathrm{T}}$ rad/s
MPC horizon	$N_u = 4$, $m = 5$
Weight matrices	$\boldsymbol{Q}^{\varepsilon(k)} = \text{diag}(20, 20, 20, 200, 200, 200)$, $\boldsymbol{R}^1 = \text{diag}(2.5, 2.5)$, $\boldsymbol{R}^2 = \cdots = \boldsymbol{R}^m = \text{diag}(2.5, 2.5, 2.5)$
Input constraints	$\Delta u_{1,i} = 0.02$ m, $\Delta u_{2,i} = 0.2$A $\cdot$ m^2

Fig. 8 presents the simulation results using these three algorithms. There are apparent jitters on error MRP and error angular velocity due to random errors in the measurements and additional inertial disturbances in the dynamics. These three controllers show different performances on the attitude stability. Specifically, since this paper adopts the idea of updating inputs asynchronously, additional inertial disturbances are significantly reduced. The attitude stability in case of dual-rate PWA-MMPC is much better than that of PWA-SLMPC, even better than that of PWA-SNMPC using a nonlinear prediction model.

Furthermore, numerical simulations are conducted to compare the computational complexity of these three algorithms. We did these numerical simulations on an AMD core, Ryzen-5800H@3.20Ghz, with Matlab R2021a software. The *quadprog* function is used to solve QP problems, the *fmincon* function to solve nonlinear optimization problems, and *tic*/*toc* function to count the time. Table 5 lists the time costs of these numerical simulations with the simulation time of 2500 s. As expected, the proposed dual-rate PWA-MMPC costs less time than PWA-SLMPC and much less than PWA-SNMPC. Therefore, the proposed algorithm has a good balance of the control performance and the computational complexity, and thus it is a promising attitude algorithm for the small satellite with MMD.

Fig. 9 depicts the time responses of Euler angle and Euler angular velocity. Under control of the proposed dual-rate PWA-MMPC algorithm, the satellite can be driven to the desired attitude within 1300 s with the Euler angle accuracy being $\pm 0.4°$ and the Euler angular velocity accuracy being $\pm 0.04°/s$. Due to the characteristic of the MPC algorithm, there is only a very small static error on the

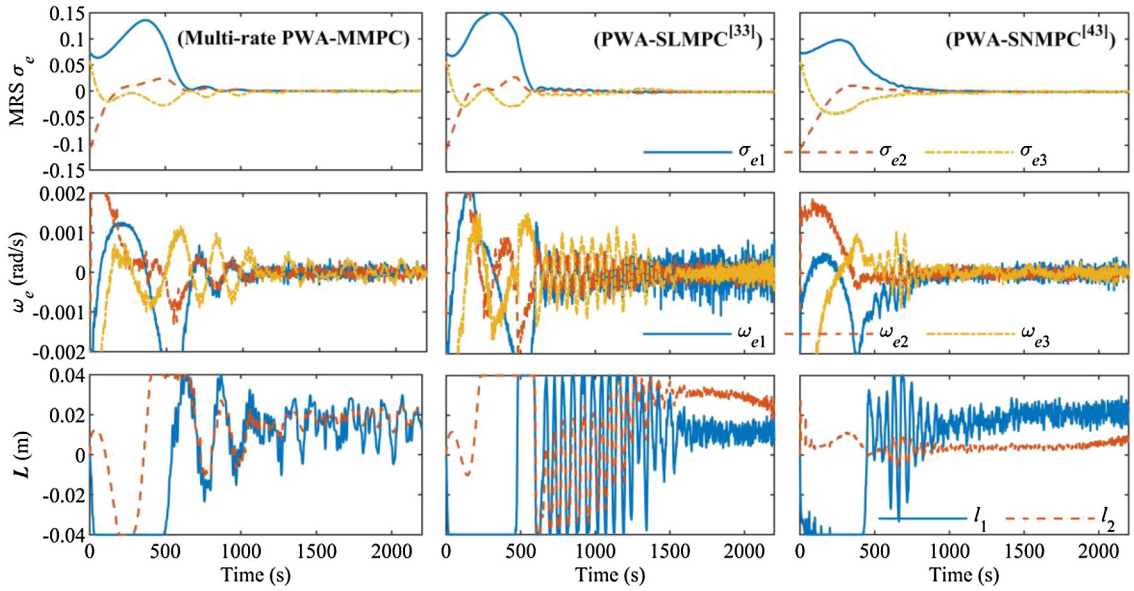

Fig. 8. Comparison of dual-rate MMPC, SLMPC in [33], and SNMPC in [43].

Table 5
Comparison of time costs of numerical simulations.

N_p	Dual-rate PWA-MMPC	PWA-SLMPC	PWA-SNMPC
3	\	\	310.45 s
6	35.83 s	41.24 s	1396.86 s
11 ($N_u = 3$)	37.22 s	50.81 s	\
16 ($N_u = 4$)	39.29 s	62.39 s	\
21 ($N_u = 5$)	44.06 s	75.46 s	\
26 ($N_u = 6$)	48.14 s	94.73 s	\

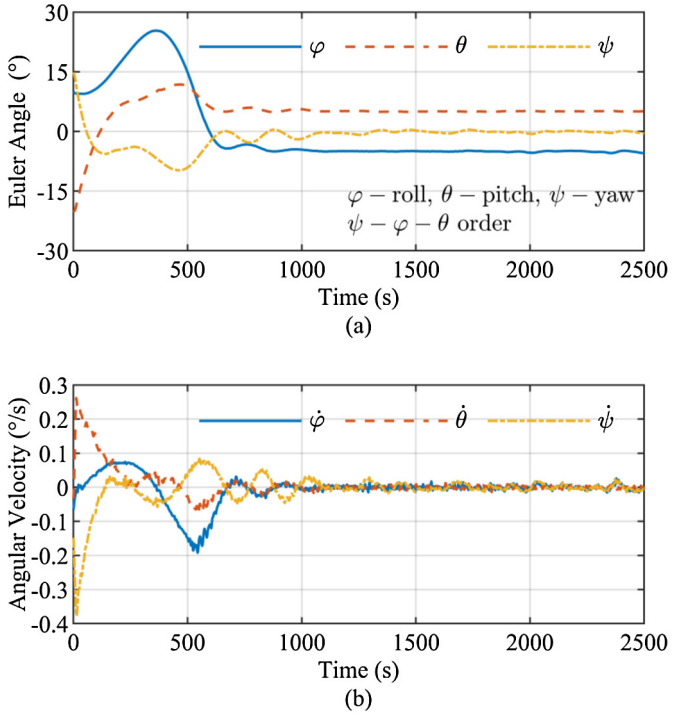

Fig. 9. Curves of Euler angle and Euler angular velocity.

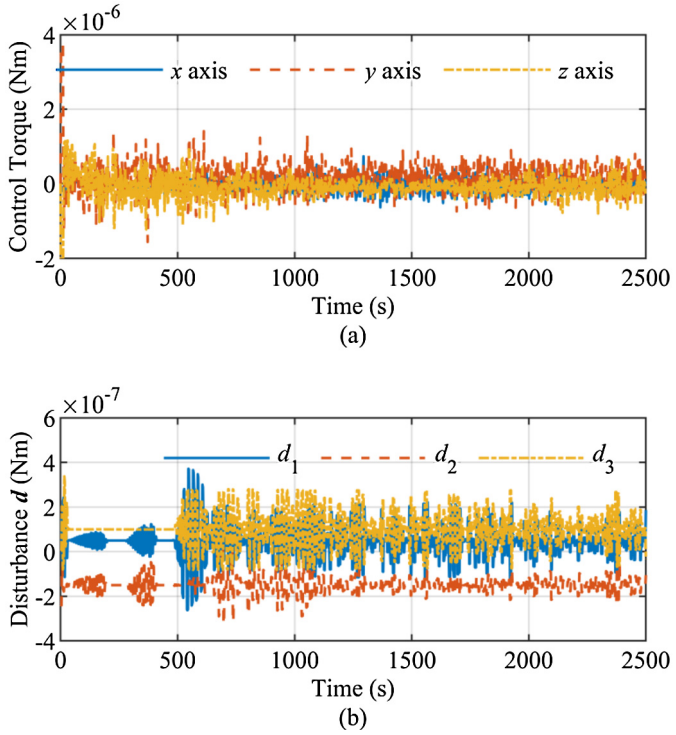

Fig. 10. Curves of control torque and disturbance.

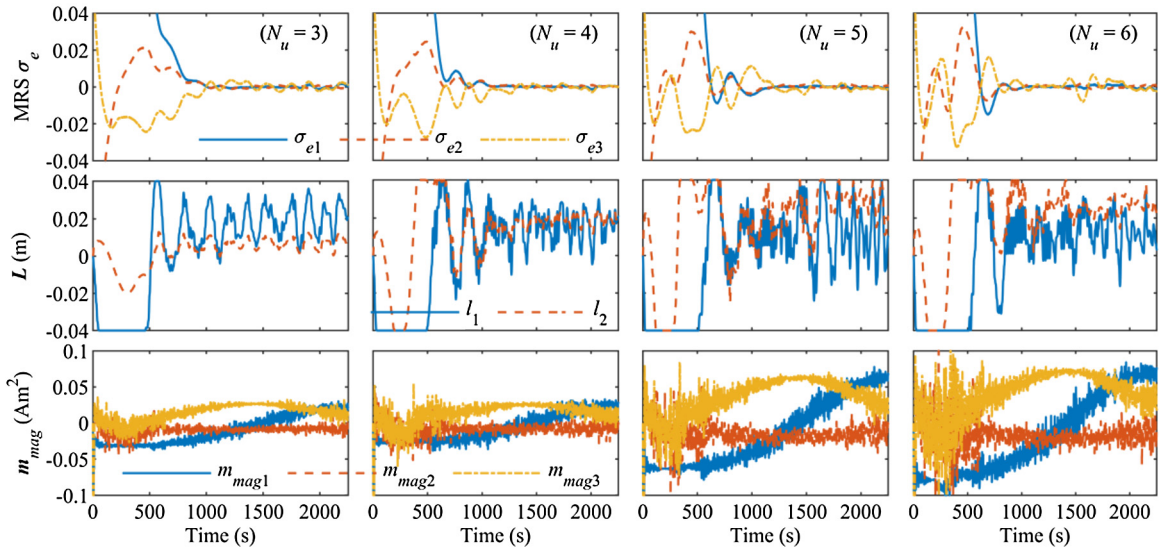

Fig. 11. Comparison of different control horizons.

final attitude despite internal parametric uncertainties, inaccurate aerodynamic forces, and other unknown disturbances. Fig. 10 depicts the time responses of the control torque and the system disturbances during the simulation. It shows that this dual-rate controller can reduce the system disturbance to be an order of magnitude smaller than the control torque, ensuring the attitude stability of the satellite.

To illustrate the performance of this dual-rate PWA-MMPC algorithm more clearly, we simulate the attitude control with different parameter conditions for comparison. Fig. 11 presents the simulation curves with different cases of control horizons. When extending the control horizon, the attitude control accuracy decreases instead. It is because the error of the state predicted by the linear PWA system at time k will be enormous if this prediction state is far from the approximation point. Besides, two inputs also tend to vary more dramatically as the state prediction error increases. Seen from Table 5, as with conventional MPC, the computational complexity of the proposed dual-rate PWA-MMPC algorithm will also increase when the prediction horizon is longer. It is caused by the increase of the dimension of variable remained to be optimized in QP problem.

Fig. 12 presents the simulation curves with different cases of input orders. The parameter m is a given parameter which determines the update rates of two inputs. The attitude control system tends to the strategy that synchronously updates inputs as m decreases, and to the strategy of single magnetic control as m increases. Therefore, the tradeoff principle must be used to choose these controller parameters.

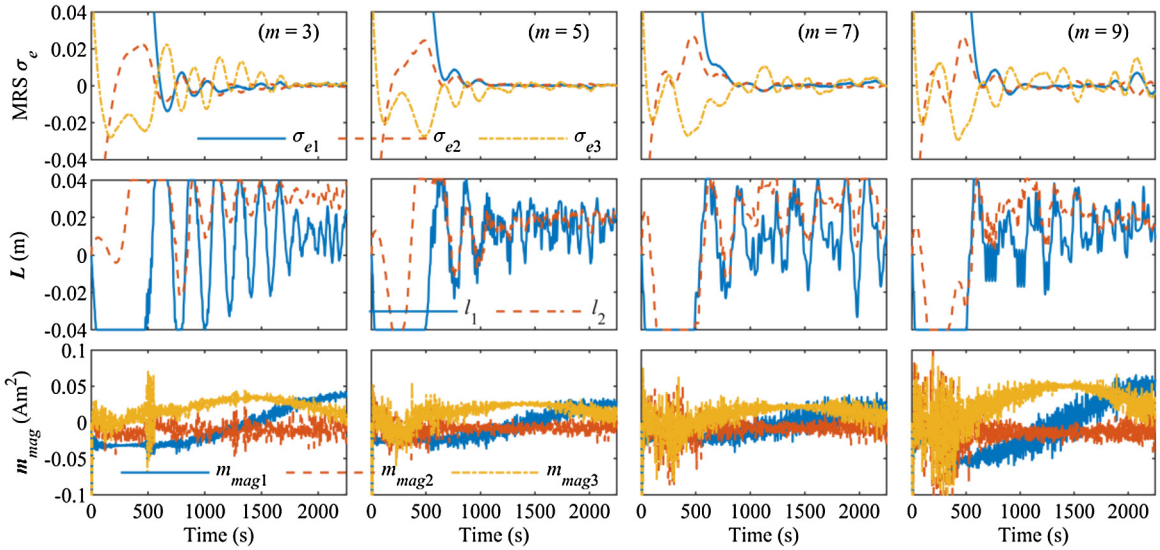

Fig. 12. Comparison of different input orders.

6. Conclusions

The inherent inertial disturbance problem remains to be solved when using the MMD to actively control aerodynamic forces. This paper proposed a dual-rate PWA-MMPC attitude control algorithm for the satellite with the MMD and the magnetorquer as actuators. This algorithm updates only one input at a time according to a given order, achieving the asynchronous update of control inputs. It effectively balances the inertial disturbances and the system's dynamic performance. Error kinematic and dynamic equations of this attitude control system are derived, converting the tracking problem into an equivalent regulation problem. Subsequently, the nonlinear attitude control equations are approximated using multiple linear state-space models, thereby achieving a PWA model suitable for LMPC design. The dual-rate PWA-MMPC algorithm is then designed, and its stability is analyzed. Finally, hardware-in-the-loop simulations are carried out and reveal that, compared with the other two algorithms, the proposed dual-rate PWA-MMPC algorithm offers a better performance on attitude stability and costs less time on optimization problem computation. Furthermore, the authors' future works will be devoted to generalizing this MMPC algorithm to a multi-input control system with a real-time input order.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J.R. Kopacz, R. Herschitz, J. Roney, Small satellites an overview and assessment, Acta Astronaut. 170 (2020) 93–105, https://doi.org/10.1016/j.actaastro.2020.01.034.
[2] Y. Hu, Z. Lu, W. Liao, X. Zhang, Differential aerodynamic force-based formation control of nanosatellites using yaw angle deviation, J. Guid. Control Dyn. 44 (12) (2021) 2199–2213, https://doi.org/10.2514/1.G006141.
[3] J.T. King, Increasing agility in orthogonal reaction wheel attitude control systems, Acta Astronaut. 177 (2020) 673–683, https://doi.org/10.1016/j.actaastro.2020.08.027.
[4] S. Ahmed khan, S. Yang, A. Anwar, S. Rao, S. Fahad, J. Wang, J. Tong, M. Tahir, Active attitude control for microspacecraft: a survey and new embedded designs, Adv. Space Res. 69 (10) (2022) 3741–3769, https://doi.org/10.1016/j.asr.2022.02.020.
[5] S. Firuzi, S. Gong, Attitude control of a flexible solar sail in low Earth orbit, J. Guid. Control Dyn. 41 (8) (2018) 1715–1730, https://doi.org/10.2514/1.G003178.
[6] M.I. Martinelli, R.S. Sanchez-Peña, Passive 3 axis attitude control of MSU-1 pico-satellite, Acta Astronaut. 56 (2005) 507–517, https://doi.org/10.1016/j.actaastro.2004.10.007.
[7] R. Sutherland, I. Kolmanovsky, A.R. Girard, Attitude control of a 2U CubeSat by magnetic and air drag torques, IEEE Trans. Control Syst. Technol. 27 (3) (2019) 1047–1059, https://doi.org/10.1109/TCST.2018.2791979.
[8] R. Sun, J. Wang, D. Zhang, Q. Jia, X. Shao, Roto-translational spacecraft formation control using aerodynamic forces, J. Guid. Control Dyn. 40 (10) (2017) 2556–2568, https://doi.org/10.2514/1.G003130.
[9] L. He, X. Chen, K.D. Kumar, T. Sheng, C. Yue, A novel three-axis attitude stabilization method using in-plane internal mass-shifting, Aerosp. Sci. Technol. 92 (2019) 489–500, https://doi.org/10.1016/j.ast.2019.06.019.
[10] Z. Lu, Study on mass moment attitude control for fast orbit maneuver satellite, Ph.D. Thesis, Nanjing University of Science and Technology, 2017.
[11] S. Chesi, Q. Gong, M. Romano, Aerodynamic three-axis attitude stabilization of a spacecraft by center-of-mass shifting, J. Guid. Control Dyn. 40 (7) (2017) 1613–1626, https://doi.org/10.2514/1.G002460.
[12] J. Virgili-Llop, H.C. Polat, M. Romano, Attitude stabilization of spacecraft in very low Earth orbit by center-of-mass shifting, Front. Robot. AI 6 (2019) 1–11, https://doi.org/10.3389/frobt.2019.00007.
[13] R. Sun, C. Riano-Rios, R. Bevilacqua, N.G. Fitz-Coy, W.E. Dixon, CubeSat adaptive attitude control with uncertain drag coefficient and atmospheric density, J. Guid. Control Dyn. 44 (2) (2021) 379–388, https://doi.org/10.2514/1.G005515.
[14] C. Riano-Rios, R. Sun, R. Bevilacqua, W.E. Dixon, Aerodynamic and gravity gradient based attitude control for CubeSats in the presence of environmental and spacecraft uncertainties, Acta Astronaut. 180 (2021) 439–450, https://doi.org/10.1016/j.actaastro.2020.12.038.
[15] Z. Lu, Y. Hu, W. Liao, X. Zhang, Attitude control of the low Earth orbit CubeSat using a moving mass actuator, Proc. Inst. Mech. Eng., G J. Aerosp. Eng. 236 (10) (2022), https://doi.org/10.1177/09544100211048275.
[16] Y. Hu, Z. Lu, W. Liao, Speed-adaptive dynamic surface attitude control for a satellite with moving masses under input constraints, Trans. Inst. Meas. Control 42 (16) (2020) 3091–3109, https://doi.org/10.1177/0142331220940427.

[17] C. Zagaris, H. Park, J. Virgili-Llop, R. Zappulla, M. Romano, I. Kolmanovsky, Model predictive control of spacecraft relative motion with convexified keep-out-zone constraints, J. Guid. Control Dyn. 41 (9) (2018) 2054–2062, https://doi.org/10.2514/1.G003549.
[18] Q. Hu, J. Xie, C. Wang, Dynamic path planning and trajectory tracking using MPC for satellite with collision avoidance, ISA Trans. 84 (2019) 128–141, https://doi.org/10.1016/j.isatra.2018.09.020.
[19] M. Mirshams, M. Khosrojerdi, Attitude control of an underactuated spacecraft using tube-based MPC approach, Aerosp. Sci. Technol. 48 (2016) 140–145, https://doi.org/10.1016/j.ast.2015.09.018.
[20] X. Zhang, K.V. Ling, Z. Lu, X. Zhang, W. Liao, W.S. Lim, Piece-wise affine MPC-based attitude control for a CubeSat during orbital manoeuvres, Aerosp. Sci. Technol. 118 (2021) 106997, https://doi.org/10.1016/j.ast.2021.106997.
[21] U. Lee, M. Mesbahi, Constrained autonomous precision landing via dual quaternions and model predictive control, J. Guid. Control Dyn. 40 (2) (2017) 292–308, https://doi.org/10.2514/1.G001879.
[22] Y. Yang, Coupled orbital and attitude control in spacecraft rendezvous and soft docking, Proc. Inst. Mech. Eng., G J. Aerosp. Eng. 233 (9) (2019) 3109–3119, https://doi.org/10.1177/0954410018792991.
[23] J. Zhang, T. Shen, Real-time fuel economy optimization with nonlinear MPC for PHEVs, IEEE Trans. Control Syst. Technol. 24 (6) (2016) 2167–2175, https://doi.org/10.1109/TCST.2016.2517130.
[24] E. Hill, S.A. Gadsden, M. Biglarbegian, Explicit nonlinear MPC for fault tolerance using interacting multiple models, IEEE Trans. Aerosp. Electron. Syst. 57 (5) (2021) 2784–2794, https://doi.org/10.1109/TAES.2021.3065089.
[25] S.D. Cairano, M. Brand, S.A. Bortoff, Projection-free parallel quadratic programming for linear model predictive control, Int. J. Control 86 (8) (2013) 1367–1385, https://doi.org/10.1080/00207179.2013.801080.
[26] E. Kayacan, W. Saeys, H. Ramon, C. Belta, J.M. Peschel, Experimental validation of linear and nonlinear MPC on an articulated unmanned ground vehicle, IEEE/ASME Trans. Mechatron. 23 (5) (2018) 2023–2030, https://doi.org/10.1109/TMECH.2018.2854877.
[27] W. EI Aouni, L.A. Dessaint, Real-time implementation of input-state linearization and model predictive control for robust voltage regulation of a DC-DC boost converter, IEEE Access 8 (2020) 192101–192108, https://doi.org/10.1109/ACCESS.2020.3032327.
[28] A. Golzari, H. Nejat Pishkenari, H. Salarieh, T. Abdollahi, Quaternion based linear time-varying model predictive attitude control for satellites with two reaction wheels, Aerosp. Sci. Technol. 98 (2020) 105677, https://doi.org/10.1016/j.ast.2019.105677.
[29] J. Imura, Optimal control of sampled-data piecewise affine systems, Automatica 40 (4) (2004) 661–669, https://doi.org/10.1016/j.automatica.2003.11.012.
[30] J. Imura, A. van der Schaft, Characterization of well-posedness of piecewise-linear systems, IEEE Trans. Autom. Control 45 (9) (2000) 1600–1619, https://doi.org/10.1109/9.880612.
[31] S. Pang, S. Jafari, T. Nilolaidis, Q. Li, Reduced-dimensional MPC controller for direct thrust control, Chin. J. Aeronaut. 35 (4) (2022) 66–81, https://doi.org/10.1016/j.cja.2021.08.024.
[32] K.V. Ling, W.K. Ho, Y. Feng, B. Wu, Integral-square-error performance of multiplexed model predictive control, IEEE Trans. Ind. Inform. 7 (2) (2011) 196–203, https://doi.org/10.1109/TII.2011.2106451.
[33] K.V. Ling, J. Maciejowski, A. Richards, B. Wu, Multiplexed model predictive control, Automatica 48 (2) (2012) 396–401, https://doi.org/10.1016/j.automatica.2011.11.001.
[34] H. Richter, A.V. Singaraju, J.S. Litt, Multiplexed predictive control of a large commercial turbofan engine, J. Guid. Control Dyn. 31 (2) (2008) 273–281, https://doi.org/10.2514/1.30591.
[35] X.F. Yang, L.W. Zhang, W. Xie, J. Zhang, Sequential and iterative distributed model predictive control of multi-motor driving cutterhead system for TBM, IEEE Access 7 (2019) 46977–46989, https://doi.org/10.1109/ACCESS.2019.2908388.
[36] Y. Hu, Z. Lu, W. Liao, Speed-adaptive dynamic surface attitude control for a satellite with moving masses under input constraints, Trans. Inst. Meas. Control 42 (16) (2020) 3091–3109, https://doi.org/10.1177/0142331220940427.
[37] L. Lu, Y. Hu, W. Liao, X. Zhang, Modeling and attitude control of CubeSat utilizing moving mass actuators, Adv. Space Res. 67 (1) (2021) 521–530, https://doi.org/10.1016/j.asr.2020.09.027.
[38] J.M. Picone, A.E. Hedin, D.P. Drob, NRLMSISE-00 empirical model of the atmosphere: statistical comparisons and scientific issues, J. Geophys. Res. 107 (A12) (2002) 1–16, https://doi.org/10.1029/2002JA009430.
[39] Q. Tang, Y. Zhou, Z. Du, C. Zhou, J. Qiao, G. Chen, A comparison of meteor radar observation over China region with Horizontal Wind Model (HWM14), Atmosphere 12 (98) (2021) 1–12, https://doi.org/10.3390/atmos12010098.
[40] M. Mandea, A. Chambodut, Geomagnetic field processes and their implications for space weather, Surv. Geophys. 41 (6) (2020) 1611–1627, https://doi.org/10.1007/s10712-020-09598-1.
[41] N. Lasemi, H.R. Shaker, Spacecraft attitude control: application of fine trajectory linearization control, Adv. Space Res. 68 (9) (2021) 3663–3676, https://doi.org/10.1016/j.asr.2021.08.018.
[42] R.E. Kalman, J.E. Bertram, Control system analysis and design via the 'Second Method' of Lyapunov: II—Discrete-time systems, J. Fluids Eng. 82 (1960) 394–400, https://doi.org/10.1115/1.3662605.
[43] D.Y. Lee, R. Gupta, U.V. Kalabić, S. Di Cairano, A.M. Bloch, J.W. Cutler, I.V. Kolmanovsky, Geometric mechanics based nonlinear model predictive spacecraft attitude control with reaction wheels, J. Guid. Control Dyn. 40 (2) (2017) 309–319, https://doi.org/10.2514/1.G001923.