Aerospace Science and Technology 110 (2021) 106441

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Short communication

On low-complexity control design to spacecraft attitude stabilization: An online-learning approach [☆]

Chengxi Zhang [a], Bing Xiao [b,*], Jin Wu [c], Bo Li [d]

[a] School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen 518055, China
[b] School of Automation, Northwestern Polytechnical University, Xi'an 710072, China
[c] Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China
[d] Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China

ARTICLE INFO

Article history:
Received 20 September 2020
Received in revised form 20 November 2020
Accepted 13 December 2020
Available online 30 December 2020
Communicated by Chaoyong Li

Keywords:
Spacecraft control
Attitude stabilization
Online-learning control (OLC)

ABSTRACT

This paper studies the spacecraft attitude stabilization problem with external disturbances. A new control scheme entitled *online-learning control* is proposed to achieve a robust, accurate, and simple-structure control algorithm. Compared with the conventional control design, an obvious distinction of the online-learning control algorithm is that it together utilizes the previous control input information and the system's current state information, as if learning experience from previous control input. In contrast, the conventional control scheme does not fully use the existing information and chooses to discard the previous control input information when generating control instructions. Due to the learning strategy, the utility of adaptive- or observer-based tools can be avoided when designing a robust control law, making a simple, effective algorithm, moreover saving system resources. The proposed control law can stabilize the attitude system by achieving the uniformly ultimately bounded convergence.

1. Introduction

1.1. Motivation

The spacecraft attitude control during operation is significant. The demands for the control schemes are threefold.

1. Achieve the control purpose quickly and accurately.
2. Respond robustly to non-ideal internal and external disturbances.
3. Achieved with inexpensive online computations.

The listed demands are always in trade-offs. Various efforts have been made in designing spacecraft attitude control algorithms, such as [1–14]. In the literature, most existing results consider the first twofold and make balance. For example, to improve the control robustness, passive adaptive approaches [6,9,15] and active observers [3,4,7,11,16] are utilized; to achieve high accuracy control performance, sliding mode control [8,9,16], prescribed performance control [1,10], and neural networks-based approach [12,17] are utilized. Most previous investigations have been achieved to cope with unpredictable variations in the ideal case, *i.e.*, without considering computation consumption and may lead to sophisticated control algorithms and engineering systems.

However, on the premise of ensuring satisfying control performance and robustness, we also need to pay attention to the often-overlooked problem that the algorithm structure should simplify engineering implementation and save system resources. Designing a low-complexity but effective control algorithm is a remaining challenge, which is also the existing studies' weakness. The focus of this paper is to design a structure-simple and effective control algorithm.

1.2. Related works

This paper proposes a new online-learning control algorithm with a simple structure for spacecraft attitude control systems. The innovation owes to a distinct perspective in coping with disturbances and actuator faults and avoiding employing sophisticated adaptive and observer methods. Unlike conventional control algorithms, whose control input command is only composed of the current system state and discards the previous information, the OLC input is composed twofold, *i.e.*, the control input before a learning interval and the current system state. This investigation

☆ This research is partially supported by the National Natural Science Foundation of China (No. 62003112, 61873207), the Shenzhen Government Basic Research Grant (JCY20170412151226061, JCY20170808110410773, JCYJ20180507182241622).

* Corresponding author.
E-mail addresses: dongfangxy@163.com (C. Zhang), xiaobing@nwpu.edu.cn (B. Xiao), jin_wu_uestc@hotmail.com (J. Wu), libo@shmtu.edu.cn (B. Li).

https://doi.org/10.1016/j.ast.2020.106441
1270-9638/

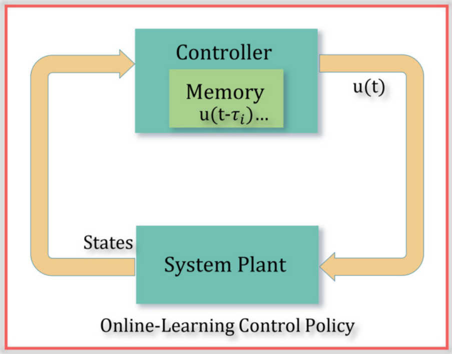

Fig. 1. Simple structure of online-learning control scheme.

is based on a simple idea that the current state of a physical system $x(t)$ is evolved from the previous state $x(t-\tau)$ with $\tau > 0$, therefore the control input $u(t)$ should also be generated based on $u(t-\tau)$.

To see this point more clearly, consider the following system example

$$\dot{x} = f(x,t) + u. \tag{1}$$

A conventional controller can be characterized as

$$u = -e(t,x) \tag{2}$$

whose features are apparent.

1. The control commands are only associated with the system's current states, and the previous control input information is regarded as the time-delay item with negative impact and hence discarded regardless of whether it contains valuable experience.
2. It is a predicament that if a straightforward control design u is used to reduce computational complexity, the control performance and robustness may be adversely affected; or, advanced algorithms enable high-performance control but adversely increase computational complexity.

Since the previous control input information is discarded, the controller cannot learn from the past control input and obtain past experiences dealing with disturbances. In general, the second feature of conventional controllers is unavoidable. For example, we can utilize a simple proportion-type controller to achieve the control objective for computation convenience, but static errors will occur. Though the integration item can improve this problem, it will extra increase the computational complexity. As for practical control systems, integration is a more complex operation than addition. We encounter a worth exploring problem that achieves sound control performance improvement with a moderate may increase in computational complexity. To solve this problem, we propose an online-learning control concept. The idea of OLC policy can be shown as

$$u = \underbrace{k_1 u(t-\tau)}_{\textit{learning item}} - \underbrace{e(t,x)}_{\textit{updating item}} \tag{3}$$

where $k_1 > 0$ is called the *learning intensity* from time $t-\tau$; $\tau > 0$ is called the *learning interval*. The control system structure is given in Fig. 1. The only difference between the conventional control scheme and OLC is that the OLC scheme requires to keep part of the previous information using a memory module. When the controller computes the control input commands, the previous information is weighted into the new control commands with a preset learning intensity. As in the previous analysis, we expect that the OLC scheme can benefit from learning the previous information when computing the control commands.

1. It fully uses the system's previous information by its online-learning mechanism, making the algorithm simple in structure, convenient in parameter tuning, and holds excellent engineering significance;
2. It owns strong robustness to disturbances and actuator faults while avoiding employing sophisticated adaptive and observer approaches, thereby saving computing resources.

It should be indicated that there is no explicit definition of the *learning* algorithm in control theory. The proposed OLC scheme is different from the learning algorithms using neural-networks [10,17], or iterative learning control [18], repetitive learning control [19], or reinforcement learning [20,21], not going in detail hither. Although the idea in (3) is simple, one obstacle lies in the stability analysis, which may not have been fully conducted.

1.3. Summary of contributions

1. We introduce the concept of *learning difference* to describe the difference between information over a specific time interval. Then, present a way to transform the stability proof problem of OLC into the conventional control stability proof problem.
2. We prove that the OLC-based spacecraft attitude control system can obtain high robustness, satisfying control performance, computation-saving, and achieve uniformly ultimately bounded convergence. A $\mathcal{L}^p$-function based performance enhancement technique is given to adjust the control performance further.
3. A general OLC is given. We draw a corollary from the analysis that OLC can be employed simultaneously using learning multiple previous information, and when the corresponding learning intensity suffices certain conditions, the system still maintains stability.

1.4. Outline

Section 2 is the Preliminaries. Section 3 presents the main results which are the control design, the stability analysis and further discussions on the generalized form of the OLC. A comparative simulation is given in Section 4. Finally the concluding remarks are given in Section 5.

2. Preliminaries

In attitude control, we are accustomed to denoting an inertial system as $\mathcal{I}$, and the frame fixedly connected to the spacecraft body as $\mathcal{B}$. The attitude or orientation of spacecraft, is defined by the state of $\mathcal{B}$ relative to $\mathcal{I}$. The spacecraft system model are given by [6, Section II], with attitude dynamics

$$J\dot{\omega} = -\omega^{\times} J\omega + u(t) + d, \tag{4}$$

and kinematics

$$\dot{q} = \frac{1}{2}\left(q^{\times} + q_0 I_3\right)\omega \tag{5}$$

$$\dot{q}_0 = -\frac{1}{2} q^{\top}\omega \tag{6}$$

where $J \in \mathbb{R}^{3\times 3}$ denotes the inertia matrix; $\omega \in \mathbb{R}^{3\times 1}$ denotes the angular velocity defined by $\mathcal{B}$ with respect to $\mathcal{I}$ and expressed in $\mathcal{B}$; $\mathrm{col}(q \in \mathbb{R}^{3\times 1}, q_0 \in \mathbb{R})$ is the attitude unit-quaternion where q is the vector part and q_0 is the scalar part, besides $q^{\top}q + q_0^2 = 1$.

$d \in \mathbb{R}^{3\times1}$ denotes the disturbances which is typically assumed to be bounded *i.e.*, $\|d\| \le \bar{d}$; $u = u(t) \in \mathbb{R}^{3\times1}$ denotes the control torques produced by actuators. The operator $(x)^\times$ for a vector $x = [x_1, x_2, x_3]^\top$ denotes a skew-symmetric matrix

$$x^\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}.$$

The control objective is to develop a robust attitude control scheme such that the uniformly ultimately bounded [22, Definition 1] stabilization can be achieved in the presence of external disturbances/faults. The goals can be briefly described as follows.

1) The closed-loop system is globally stable.
2) The attitude and angular velocity converge to a small region.

3. Online-learning control design

In this section, we first present the OLC scheme for spacecraft attitude stabilization with stability analysis, then the general-OLC scheme; finally, we present a simple enhancement technique that can further improve the control performance.

3.1. Control law design and stability analysis

Denote an aided variable

$$s(t) = \omega + \sigma q \tag{7}$$

where $\sigma > 0$ is a constant. Besides, we choose

$$e(t) = s(t) - \frac{1}{\kappa_1} P(\cdot) \tag{8}$$

where $P(\cdot) = -\omega^\times J\omega + \frac{\sigma}{2} J\left(q^\times + q_0 I_3\right)\omega$ and $\kappa_1 > 0$ is a constant to be designed. Then, the OLC law is proposed as follows

$$u(t) = k_1 u(t-\tau) - k_2 e(t) \tag{9}$$

where $1 > k_1 > 0$, $k_2 > 0$ are constants; $\tau > 0$ is called *learning interval*. $e(t) \in \mathbb{R}^{3\times1}$ is the *updating variable* which is under design. k_1 and k_2 respectively represent the weight of $u(t-\tau)$ and the current system state $e(t)$ on $u(t)$, and k_1 can be called *learning intensity*.

Definition 1. We denote a variable

$$\tilde{u} = u(t) - u(t-\tau), \tag{10}$$

which is called the *learning difference*.

From Definition 1, we have $u(t-\tau) = u(t) - \tilde{u}$. Therefore the control law (9) can be expressed as

$$u(t) = k_1(u(t) - \tilde{u}) - k_2 e(t), \tag{11}$$

and thus can be simplified as

$$\begin{aligned} u(t) &= -\frac{k_2}{1-k_1} e(t) - \frac{k_1}{1-k_1}\tilde{u} \\ &= -\kappa_1 e(t) - \kappa_2 \tilde{u} \end{aligned} \tag{12}$$

where $\kappa_1 = \frac{k_2}{1-k_1}$, $\kappa_2 = \frac{k_1}{1-k_1}$.

Remark 1. $\|\tilde{u}\|$ is bounded by a constant $\bar{\tilde{u}}$. In actual physical systems, the state of the actuator changes gradually. Even for impulsive actuators, they exhibit the properties of gradual change because of the build-up time and decay time. Since actual actuators have the *maximum output limitation*, the learning difference is naturally bounded.

Proposition 1. *Considering spacecraft attitude control system given by (4) - (6), then by using the control law given by (9) and (8), the attitude stabilization can be achieved, provided*

$$0 < k_1 < 1, \ \frac{1}{2} < k_2. \tag{13}$$

Proof of Proposition 1. Consider the following Lyapunov function candidate

$$V(t) = \frac{1}{2} s^\top(t) J s(t). \tag{14}$$

Then, the derivative of $V(t)$ is $\dot{V}(t) = s^\top(t) J \dot{s}(t)$. Note that

$$\begin{aligned} J\dot{s}(t) &= -\omega^\times J\omega + u(t) + d + \frac{\sigma}{2} J\left(q^\times + q_0 I_3\right)\omega \\ &= P(\cdot) + u(t) + d. \end{aligned} \tag{15}$$

Thus, we have

$$\dot{V}(t) = s^\top(t)(P(\cdot) + u(t) + d). \tag{16}$$

Substituting (11) into (16), it yields

$$\begin{aligned} \dot{V}(t) &= s^\top(t)\begin{pmatrix} P(\cdot) - \kappa_1(s(t) - \frac{1}{\kappa_1}P(\cdot)) \\ -\kappa_2\tilde{u} + d \end{pmatrix} \\ &= s^\top(t)\left(-\kappa_1(s(t) - \kappa_2\tilde{u} + d\right) \\ &= -\kappa_1 s^\top(t)s(t) + s(t)^\top(d - \kappa_2\tilde{u}) \\ &\le -\kappa_1\|s(t)\|^2 + \|s(t)\|\|d\| + \kappa_2\|s(t)\|\|\tilde{u}\| \\ &\le -\left(\kappa_1 - \frac{1}{2} - \frac{1}{2}\kappa_2\right)\|s(t)\|^2 + \frac{1}{2}\|\bar{d}\| + \frac{1}{2}\kappa_2^2\bar{\tilde{u}} \\ &\le -\eta\|s(t)\|^2 + \delta \end{aligned} \tag{17}$$

where $\eta = \kappa_1 - \frac{1}{2} - \frac{1}{2}\kappa_2$, $\delta = \frac{1}{2}\bar{d} + \frac{1}{2}\kappa_2^2\bar{\tilde{u}}$. Since $\|\tilde{u}\| \le \bar{\tilde{u}}$, δ is bounded.

Note that $\|s(t)\|^2 \ge \frac{1}{J_{\max}} s^\top(t) J s(t)$. Thus, as long as the condition (13) is satisfied, we can easily verify $\eta > 0$, (17) can be re-expressed by

$$\begin{aligned} \dot{V}(t) &\le -\frac{2\eta}{J_{max}} V(t) + \delta \\ &\le -\pi V(t) + \delta \end{aligned} \tag{18}$$

where $\pi = \frac{2\eta}{J_{max}}$. According to [22, Theorem 1], we have

$$s^\top(t) J s(t) \le \exp(-\pi t) s^\top(0) J s(0) + \frac{\delta}{\pi}, \tag{19}$$

namely, $\lim_{t\to\infty} \|s(t)\| \le \sqrt{\frac{\delta}{\pi\lambda_{\min}(J)}}$.

According to [23, Theorem 2], when $s(t)$ converges to a small region Θ, then $|\omega_i(t)| \le 2\Theta$, $|q_i(t)| \le \Theta/\sigma$, $i = 1, 2, 3$.

This completes the proof. □

Remark 2. $k_1 u(t-\tau)$ cannot merely be viewed as a time-delay item with adversarial impact [14]. It also contains much information about the controller's acts on the system, representing the inheritance and learning from previous information. If $k_1 u(t-\tau)$ is removed, then OLC will degenerate into a conventional controller. The previous information was utterly discarded in conventional schemes when designing a controller, which also discarded the controller's efforts to deal with disturbances.

Remark 3. A better choice is to change the controller to make further adjustments based on the previous control input, learning the controller's experience in suppressing disturbances. This shifts the baseline for control commands computing. The conventional controller computes the control commands from 0, while OLC's baseline is $u(t-\tau)$.

Remark 4. It should be noted that $e(t)$ part in (9) does not require any superior design rather than conventional schemes. Namely, if $u=-e(t)$ can stabilize the system, then the OLC (9) can achieve stability by selecting appropriate learning intensity and time intervals. The nature of OLC is that $e(t)$ can be selected as simple as possible. Although it does not use advanced algorithms such as adaptive tools and observer tools, it can still ensure good control performance. Moreover, we can also see that compared to the complex structure of adaptive algorithms and observer algorithms, OLC in (9) only appends one addition-operation, which unquestionably saves computing resources.

Remark 5. It is worth discussing that the control command is generated by combining the previous control information (the previous states included) and states. We chose to use $u(t-\tau)$, avoiding extra intricate designs, which implies that many existing algorithms can be improved via the proposed results. Provided that $u=-e(t)$ is a controller that can stabilize the system, OLC can significantly upgrade the controller's accuracy and robustness. More importantly, OLC is merely a simple design without complicated tools.

Remark 6. It should be noted that OLC and reinforcement learning have certain similarities in terms of *learning existing experience*. The difference is that reinforcement learning is a data-driven algorithm that requires many optimization arrangements, such as loss and gain. It needs to use numerous computations from the current time to infer a decision after a step, such as [20]. OLC is a non-data-driven algorithm, which does not consider environmental factors, gain, and loss, but only needs to learn part of the control value from the previous moment, closer to an adaptive algorithm. Since it employs barely one algebraic equation, it is simple in structure and computationally efficient.

Remark 7. It is worth mentioning that OLC is different from similar investigation [16]. Firstly, this study's control input command is given by a differential equation, which is more computationally complex than the OLC scheme, and the derivation is sensitive to disturbances. Secondly, the control performance in [16] is based on advanced high-order observers, high-order sliding modes, with several parameters, making parameter tuning challenges. The proposed OLC is simple in structure and has low computational complexity properties. Since OLC uses fewer parameters than complex algorithms, parameter tuning is thus convenient. Therefore, OLC has strong usability.

Remark 8. The learning interval τ is a constant, which represents the position on the time axis of the learned controller's information. If the disturbance changes rapidly, it should be chosen as a small value. Oppositely, large values can also be considered. Since past information is also time-sensitive, under typical cases, it should not be chosen over large. It can be selected as one or multiple sampling intervals, especially for modern digital control systems or sampled-data control systems.

Remark 9. The learning intensity describes the intensity of the current controller inheriting the previous state. Note that in (9), OLC's learning item is more like an augmentation to a controller $u=-e(t)$. If the learning intensity is overlarge, the control input will increase rapidly when the system error is large, which may cause undesired saturation. Therefore, it is necessary to restrict the maximum control input according to the practical systems when designing the controller, which will ensure that $\|\tilde{u}\| \le \bar{\tilde{u}}$ is always satisfied. Under the previous cases, the controller will consume more energy in the initial stage of control, which is a shortcoming behind the advantage, and the coping approach can be further explored based on this investigation in the future. We also need to indicate that the OLC can work in a good state by selecting proper learning intensity parameters. Although the energy consumption is comparatively high in the initial stage of control, OLC will exhibit energy-saving properties in the longer-term control process.

Remark 10. Another idea of thinking about effectively using existing information to design a controller is time-delay control. The similarity between time-delay control and OLC is that the information previously considered harmful and should be discarded now turned into useful information when designing the controller. The core difference between them is that the time-delay control scheme's core element is the *time-delay estimation* technique. Such as the investigation in [24] and [25], much analysis is consumed to conduct time-delay estimation and then design complex controllers based on the estimation results. The proposed OLC does not demand time delay estimation, allowing designers to directly incorporate conventional control techniques into OLC, as demonstrated in the stability analysis section. This makes OLC a reasonable extension of conventional control approaches, which is a good character.

3.2. Generalized online-learning control

According to the idea of Section 3.1, a general OLC can be given by

$$u(t)=\sum_{i=1}^{M} k_i u(t-\tau_i)-k_e e(t) \tag{20}$$

where $M \in \mathbb{Z}_+$ is an positive integer; $k_e>0$, $\tau_i>0$, $i=\{1,\ldots,M\}$, are constants. Alike Definition 1, we define learning differences for different learning items in (20) as follows $\tilde{u}_i=u(t)-u(t-\tau_i)$. Therefore, from (11), (12), we have the re-expressed form of (20) as follows

$$u(t)=-\frac{k_e}{1-\sum\limits_{i=1}^{M}k_i}e(t)-\frac{\sum\limits_{i=1}^{M}k_i\tilde{u}_i}{1-\sum\limits_{i=1}^{M}k_i}. \tag{21}$$

Corollary 1. *If a system can be stabilized under controller (9), it can stabilize the system using controller (20) with $e(t)$ in (9), as long as the parameters satisfy*

$$0<\sum_{i=1}^{M}k_i<1,\ \frac{1}{2}<k_e. \tag{22}$$

Proof of Corollary 1. The proof is similar to that of Proposition 1. Thus, the detailed process is omitted. □

Remark 11. Corollary 1 is a more generalized framework for OLC, whose meaning is that we can generate current control instructions from previous system information through different intensities of learning. Simply put, multiple learning is similar to the weighted average at various times, which can achieve a *smoothing* effect on the signal. This strategy's design idea can be analogous to the weighted moving average approach in the filtering algorithm, and if k_i keeps equal weights, it degenerates into the moving average approach. Since the user chooses τ_i in (20) arbitrary, there are still differences between (20) and the filtering algorithm.

3.3. $\mathcal{L}^p$-function based control performance enhancement

Typical performance indicators usually have demands on the final convergence objective but have fewer demands on the control process. To make the system trajectory always converge in a certain way, we give a $\mathcal{L}^p$-function based prescribed performance tuning approach to our controller so that the controller can adjust the system's transient response.

Denote $s(t) = [s_1(t), s_2(t), s_3(t)]^\top$. Define a transformation function $\mathcal{T}_i : (-1, 1) \to \mathbb{R}$ to replace the $s_i(t)$ in (8). Consider $\mathcal{T}(\cdot) = [\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3]^\top$ where $\mathcal{T}_i$ is defined for each dimension of $s(t)$

$$\mathcal{T}_i(\varepsilon_i(t), \rho_i(t)) = \alpha \ln\left(\frac{1+\varepsilon_i(t)}{1-\varepsilon_i(t)}\right), \quad i = 1, 2, 3 \tag{23}$$

where $\varepsilon_i(t) = \frac{s_i(t)}{\rho_i(t)}$. The modified control law (9) can be expressed as

$$u(t) = k_1 u(t-\tau) - k_2 \left(\mathcal{T}(s(t), \rho(t)) - \frac{1-k_1}{k_2} P(\cdot)\right). \tag{24}$$

Note that $\mathcal{T}_i(\varepsilon_i(t)) > 2\alpha\|\varepsilon_i(t)\| + \frac{2}{3}\alpha\|\varepsilon_i(t)\|^3$ on $\varepsilon_i(t) \in (-1, 1)$. Hence, when we choose $2\alpha > \|\rho_i(t)\|$, it can be obtained that

$$\|\mathcal{T}_i(\varepsilon_i(t), \rho_i(t))\| \geq \|s_i(t)\|, \tag{25}$$

which means the stability will not be changed. To satisfy the definition domain $\varepsilon_i(t) \in (-1, 1)$, we provide a series of $\rho_i(t)$ functions with so-called *$\mathcal{L}^p$-functions*. For $p \in [1, +\infty)$, if a function $\xi(t)_{:\mathbb{R}_{t\geq 0}\to\mathbb{R}} \in \mathcal{L}^p$-function spaces, IFF $\xi(t)$ is p-integrable, satisfying $\int_0^\infty |\xi(t)|^p dt < \infty$. Then, $\rho_i(t)$ is given by

$$\rho_i(t) = \|s_i(t)\| + \sum_{j=1}^{N} \xi_j(t) + \rho_{i,\infty}, \quad j = 1, .., N \tag{26}$$

where N is a positive integer, constant $\rho_{i,\infty} > 0$. Note that $\|s(t)\|$ is ultimately uniformly bounded and converges to a small region, thus there exists a maximum value $\|s_{i,\max}\|$. As long as we choose $\alpha > \frac{1}{2}\rho_i(0)$, therefore, it can be concluded that $\varepsilon_i(t) \in (-1, 1)$ always.

Remark 12. Simply put, the $\mathcal{L}^p$-function is a class of functions with final convergence but is not strictly required to be in a downward trend under any circumstances. The exponential function commonly used in prescribed performance control is a particular case of $\mathcal{L}^p$-functions [26, Section III]. Common ones, such as $\frac{1}{1+t^2}$, $\frac{1}{1+t^3}$, ..., all belong to $\mathcal{L}^p$-type functions. They can tune the form of $\rho_i(t)$ through combination so that the system can reach convergence in the form of an almost prescribed way. Since $\mathcal{L}^p$-functions satisfies linear additivity, (26) holds.

Remark 13. Prescribed performance control is a powerful tool to tune the system to guarantee the system convergence within a preassigned trajectory. Performance tuning idea is motivated by [27, Section 2]. The difference from [27] is that there is no need to know the error's initial sign in (23), and the prescribed performance function (26) appends $\|s_i(t)\|$. The convergence speed of $s_i(t)$ is greater than $\rho_i(t)$, thus avoids the singularity problem. For the existence of disturbances/faults, the performance function in [27] may cause the system unstable at certain moments when encountering unpredictable, large disturbances. Therefore, appending $\|s_i(t)\|$ is equivalent to mitigating the prescribed performance condition. The expected system trajectory converges in the time-dependent trend but allows the trajectory to be exceeded at certain times, thus increasing the system's robustness.

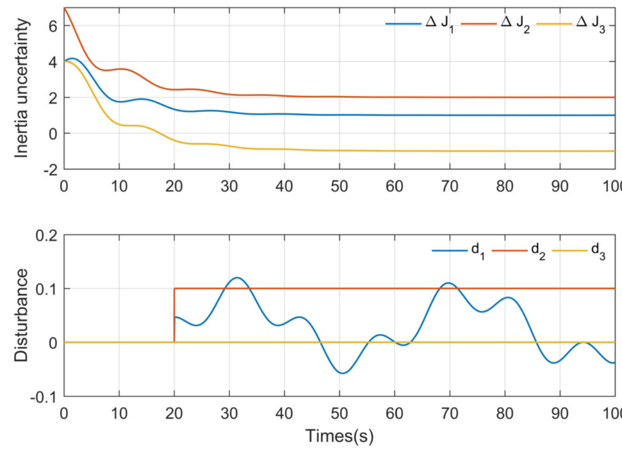

Fig. 2. Evolution of inertia uncertainty/disturbance.

4. Comparative example

We conduct a comparative simulation to show the performance.

4.1. Parameters

The controller is given by (9) and (8). The controller utilized for comparison removes the learning part in (9), *i.e.*, the Non-OLC $u = -k_2 e(t)$. In this simulation, we also consider the uncertainty of inertia to better verify the OLC scheme's performance against non-ideal internal and external disturbances. Denote $J = J_0 + \Delta J$ where J_0 is the nominal part and ΔJ is the unknown and time-varying part (due to, for example, fuel burning and payload release [28, Section IV]). The evolution of ΔJ is shown in upper part of Fig. 2. The selected parameters are as follows.

- Nominal inertia matrix (kg·m^2) [6] $J = \begin{bmatrix} 20 & 2 & 0.9 \\ 2 & 18 & 0.5 \\ 0.9 & 0.5 & 15 \end{bmatrix}$,
  Inertia uncertainty $\Delta J = \mathrm{diag}\begin{pmatrix} (3+\sin(0.5t))\,e^{-0.1t}+1 \\ (4+\cos(0.5t))\,e^{-0.1t}+2 \\ (5+\sin(0.5t))\,e^{-0.1t}-1 \end{pmatrix}$;
- Initial angular velocity $\omega(0) = [0, 0, 0]^\top$ rad/s;
- Initial attitude quaternion $\mathrm{col}(q(0), q_0(0)) = [-0.1, 0.3, -0.2, \sqrt{0.86}]^\top$;
- Control parameters $k_1 = 0.8$, $k_2 = 2$, $\tau = 0.01$ s in (9); $\alpha = 1$ in (23); $\xi(t) = \frac{1}{1+t^2}$ in $\rho_i(t)$ of (26), $\rho_{i,\infty} = 0.2$;
- Maximum actuator output 1 N·m;
- We arrange the system to run under no disturbance and fault conditions before 20 s as a comparison *i.e.*, $d = [0, 0, 0]^\top$ N·m. After 20 s, there are time-varying and constant faults in u_1 and u_2, respectively, and u_3 remains healthy $d = 0.01\,[-3\cos(0.5t) - 6\sin(0.15t) + 3, 10, 0]^\top$ N·m. The evolution of $d(t)$ is shown in the lower part in Fig. 2.

4.2. Simulation results

Figs. 3 - 6 show the simulation results, which are the evolution of inertia uncertainty/disturbance, angular velocity, attitude angle,

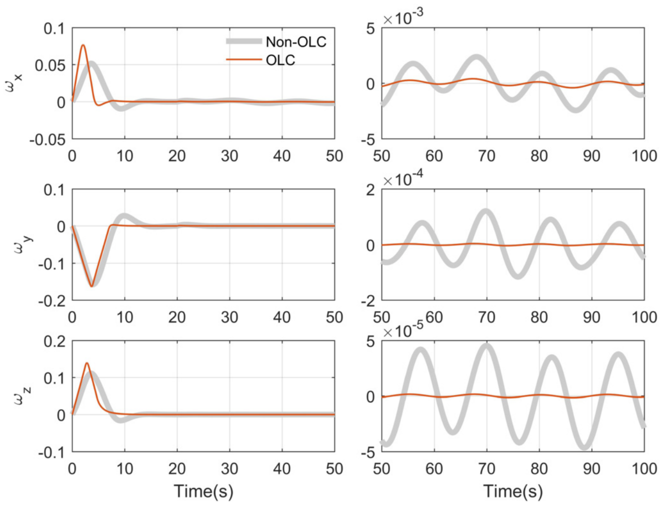

Fig. 3. Time response of angular velocity. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

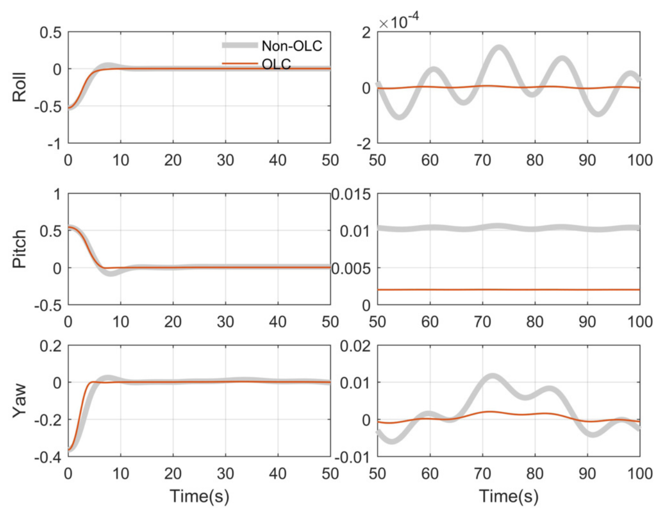

Fig. 4. Time response of attitude (quaternions to rotation angles with rotation order: ZYX).

control input of the spacecraft and energy index comparison, respectively.

To display the difference between the curves more clearly, in the simulation results of angular velocity and attitude, we use the brown-yellow line to represent the OLC simulation result and the thick gray line representing the Non-OLC simulation result. As shown in Figs. 3 and 4, within the first 20 s of the simulation, the angular velocity and attitude of the spacecraft seem to be stabilized under both controllers. However, after the 20 s, the spacecraft is subject to strong time-varying and constant fault disturbances. After the figures are zoomed in, we can see more details for the trajectory curves in Fig. 3 and Fig. 4: Starting from 20 s, it can be seen that the system state under the controlling of OLC is more stable than that of Non-OLC, almost a straight line, while the Non-OLC's results exhibit noticeable fluctuations. In terms of control performance, OLC has shown better results in the presence of inertia uncertainty and disturbance.

Fig. 5 gives the evolution of control torques. In the current parameter settings, we see that when the system error is relatively large, the OLC's control input is relatively large in the short term compared to the Non-OLC one. It can be seen from Fig. 5 that before 10 s, to make the system converge promptly, OLC demands the actuator to provide larger control torques, which is apparent.

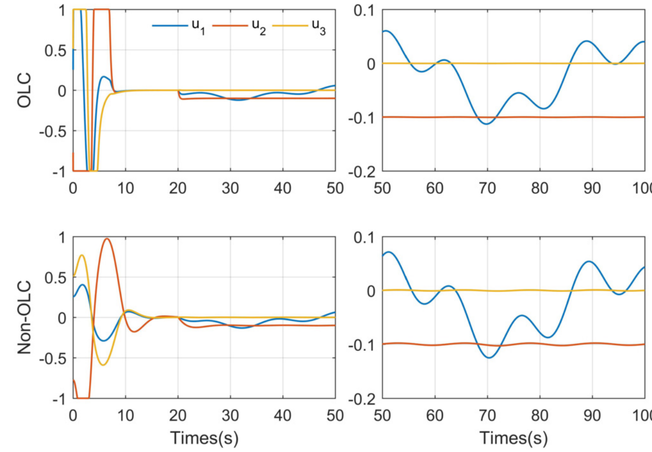

Fig. 5. Evolution of control torques.

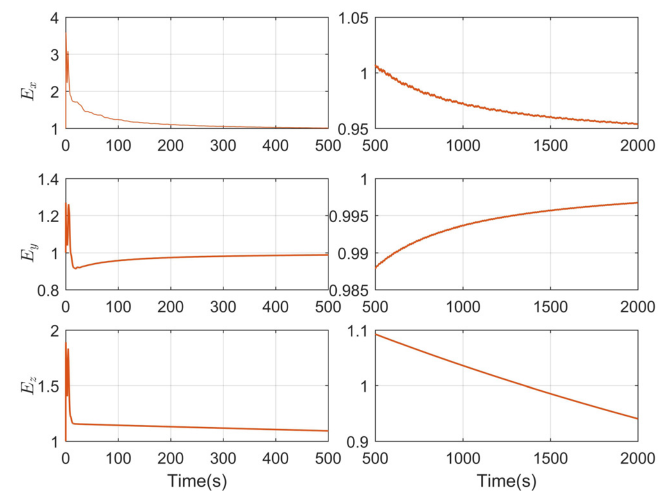

Fig. 6. Energy index comparison.

After 10 s, the control torques decrease rapidly, which can also be seen in Fig. 3 and Fig. 4. This is because the system error becomes smaller than the Non-OLC results such that OLC can also quickly learn the system's new situation. It seems that in the short term, OLC will consume more energy, which is also a shortcoming, as we have analyzed in the previous section.

In order to discuss this issue, a comparison of energy consumption is performed. Denote $E = [E_x, E_y, E_z]^\top$ as the energy index where

$$E_i = \frac{P_{i,olc}}{P_{i,non-olc}}, \; i = x, y, z$$

and $P_i = \int |u_i| dt$ is defined in [29]. If the energy consumption of the two controllers is the same, then the ratio should be 1. If the ratio is less than 1, then OLC is more energy efficient. To discuss this issue comprehensively, we show the simulation results of the energy index in 2000 seconds. It can be seen from Fig. 2 that in the initial stage of the simulation, OLC does consume more energy, and the energy index shortly arrive at a peak. Consistent with the previous simulation results, it appeared before 10 s. After 10 s, the energy index gradually decreases. After about 1200 s, all lines in three axes, x, y, z, are less than 1. These results show that OLC is more energy-efficient in the long-term, suitable for spacecraft engineering applications. Moreover, based on previous simulation results, we can see that it is profitable to further study to avoid saturation issues regarding large system errors and further enhance OLC's application value in the future.

In conclusion, this comparative simulation verifies the superior and efficient control performance of the proposed OLC scheme.

5. Concluding remarks

This paper proposes a new online-learning control algorithm for spacecraft attitude stabilization. This control design is composed of the system's previous control input information and current error, *i.e.*, a learning item is attached to the conventional controller, which inherits the previous control input experience. Unlike adaptive methods and observer methods that require intricate schemes, OLC has a simple structure and low computational complexity. It shows decent robustness and sound control performance in the simulation when coping with time-varying and constant disturbances/faults. Due to its simple structure, it consumes fewer computations and saves considerable energy power resources in the long-term run. Moreover, we propose a $\mathcal{L}^p$-function based performance tuning approach to achieve control performance enhancement and give a generalized OLC scheme for practical, convenient utilization. In the future, avoiding the saturation behavior when the system error is large will be a forthcoming study.

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] M. Liu, X. Shao, G. Ma, Appointed-time fault-tolerant attitude tracking control of spacecraft with double-level guaranteed performance bounds, Aerosp. Sci. Technol. 92 (2019) 337–346, https://doi.org/10.1016/j.ast.2019.06.017.
[2] S. Xu, H. Wen, Z. Huang, Robust fuzzy sampled-data attitude control of spacecraft with actuator saturation and persistent disturbance, Aerosp. Sci. Technol. 101 (2020) 105850, https://doi.org/10.1016/j.ast.2020.105850.
[3] B. Li, Q. Hu, Y. Yang, Continuous finite-time extended state observer based fault tolerant control for attitude stabilization, Aerosp. Sci. Technol. 84 (2019) 204–213, https://doi.org/10.1016/j.ast.2018.10.006.
[4] Q. Hu, X. Zhang, G. Niu, Observer-based fault tolerant control and experimental verification for rigid spacecraft, Aerosp. Sci. Technol. 92 (2019) 373–386.
[5] Q. Hu, G. Niu, C. Wang, Spacecraft attitude fault-tolerant control based on iterative learning observer and control allocation, Aerosp. Sci. Technol. 75 (2018) 245–253, https://doi.org/10.1016/j.ast.2017.12.031.
[6] C. Zhang, M.-Z. Dai, P. Dong, H. Leung, J. Wang, Fault-tolerant attitude stabilization for spacecraft with low-frequency actuator updates: an integral-type event-triggered approach, IEEE Trans. Aerosp. Electron. Syst., https://doi.org/10.1109/TAES.2020.3009542.
[7] C. Zhang, J. Wang, D. Zhang, X. Shao, Learning observer based and event-triggered control to spacecraft against actuator faults, Aerosp. Sci. Technol. 78 (2018) 522–530, https://doi.org/10.1016/j.ast.2018.05.007.
[8] Q. Hu, X. Shao, Smooth finite-time fault-tolerant attitude tracking control for rigid spacecraft, Aerosp. Sci. Technol. 55 (2016) 144–157, https://doi.org/10.1016/j.ast.2016.05.019.
[9] C. Zhang, J. Wang, D. Zhang, X. Shao, Fault-tolerant adaptive finite-time attitude synchronization and tracking control for multi-spacecraft formation, Aerosp. Sci. Technol. 73 (2018) 197–209, https://doi.org/10.1016/j.ast.2017.12.004.
[10] C. Wei, J. Luo, H. Dai, G. Duan, Learning-based adaptive attitude control of spacecraft formation with guaranteed prescribed performance, IEEE Trans. Cybern. 49 (11) (2018) 4004–4016, https://doi.org/10.1109/TCYB.2018.2857400.
[11] R. Sun, A. Shan, C. Zhang, J. Wu, Q. Jia, Quantized fault-tolerant control for attitude stabilization with fixed-time disturbance observer, J. Guid. Control Dyn., https://doi.org/10.2514/1.G005465.
[12] R. Sun, J. Wang, D. Zhang, X. Shao, Neural-network-based sliding-mode adaptive control for spacecraft formation using aerodynamic forces, J. Guid. Control Dyn. 41 (3) (2018) 754–760, https://doi.org/10.2514/1.G003063.
[13] W. He, T. Wang, X. He, L.-J. Yang, O. Kaynak, Dynamical modeling and boundary vibration control of a rigid-flexible wing system, IEEE/ASME Trans. Mechatron. 4435, https://doi.org/10.1109/tmech.2020.2987963.
[14] S. Xu, Z. Wei, Z. Huang, H. Wen, D. Jin, Fuzzy-logic-based robust attitude control of networked spacecraft via event-triggered mechanism, IEEE Trans. Aerosp. Electron. Syst., https://doi.org/10.1109/TAES.2020.3012000.
[15] W. He, X. Mu, L. Zhang, Y. Zou, Modeling and trajectory tracking control for flapping-wing micro aerial vehicles, IEEE/CAA J. Autom. Sin., https://doi.org/10.1109/JAS.2020.1003417.
[16] M.A. Golkani, L. Fridman, S. Koch, M. Reichhartinger, M. Horn, Observer-based saturated output feedback control using twisting algorithm, in: Proc. IEEE Int. Work. Var. Struct. Syst., July 4, 2016, 2016, pp. 246–250.
[17] W. He, C. Xue, X. Yu, Z. Li, C. Yang, Admittance-based controller design for physical human-robot interaction in the constrained task space, IEEE Trans. Autom. Sci. Eng. 17 (4) (2020) 1937–1949, https://doi.org/10.1109/TASE.2020.2983225.
[18] W. He, T. Meng, X. He, S.S. Ge, Unified iterative learning control for flexible structures with input constraints, Automatica 96 (2018) 326–336, https://doi.org/10.1016/j.automatica.2018.06.051.
[19] Y. Qian, Y. Fang, B. Lu, Adaptive repetitive learning control for an offshore boom crane, Automatica 82 (2017) 21–28, https://doi.org/10.1016/j.automatica.2017.04.003.
[20] J. Jiang, X. Zeng, D. Guzzetti, Y. You, Path planning for asteroid hopping rovers with pre-trained deep reinforcement learning architectures, Acta Astronaut. 171 (2020) 265–279, https://doi.org/10.1016/j.actaastro.2020.03.007.
[21] Y. Jiang, B. Kiumarsi, J. Fan, T. Chai, J. Li, F.L. Lewis, Optimal output regulation of linear discrete-time systems with unknown dynamics using reinforcement learning, IEEE Trans. Cybern. 50 (7) (2020) 3147–3156, https://doi.org/10.1109/TCYB.2018.2890046.
[22] Y. Lee, S.H. Zak, Uniformly ultimately bounded fuzzy adaptive tracking controllers for uncertain systems, IEEE Trans. Fuzzy Syst. 12 (6) (2004) 797–811, https://doi.org/10.1109/TFUZZ.2004.836087.
[23] B. Wu, Spacecraft attitude control with input quantization, J. Guid. Control Dyn. 39 (1) (2015) 1–5, https://doi.org/10.2514/1.G001427.
[24] M. Jin, S.H. Kang, P.H. Chang, J. Lee, Robust control of robot manipulators using inclusive and enhanced time delay control, IEEE/ASME Trans. Mechatron. 22 (5) (2017) 2141–2152, https://doi.org/10.1109/TMECH.2017.2718108.
[25] Y. Wang, F. Yan, J. Chen, F. Ju, B. Chen, A new adaptive time-delay control scheme for cable-driven manipulators, IEEE Trans. Ind. Inform. 15 (6) (2019) 3469–3481, https://doi.org/10.1109/TII.2018.2876605.
[26] L.N. Bikas, G.A. Rovithakis, Combining prescribed tracking performance and controller simplicity for a class of uncertain MIMO nonlinear systems with input quantization, IEEE Trans. Autom. Control 64 (3) (2019) 1228–1235, https://doi.org/10.1109/TAC.2018.2847458.
[27] A.K. Kostarigka, Z. Doulgeri, G.A. Rovithakis, Prescribed performance tracking for flexible joint robots with unknown dynamics and variable elasticity, Automatica 49 (5) (2013) 1137–1147, https://doi.org/10.1016/j.automatica.2013.01.042.
[28] W. Cai, X. Liao, D.Y. Song, Indirect robust adaptive fault-tolerant control for attitude tracking of spacecraft, J. Guid. Control Dyn. 31 (5) (2008) 1456–1463, https://doi.org/10.2514/1.31158.
[29] Q. Hu, L. Li, M.I. Friswell, Spacecraft anti-unwinding attitude control with actuator nonlinearities and velocity limit, J. Guid. Control Dyn. 38 (10) (2015) 1–8, https://doi.org/10.2514/1.G000980.