RESEARCH ARTICLE

Self-Learning Control under Practical Actuation

Chengxi Zhang[1,*]

[1] School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China

Abstract

This paper develops a general and implementation friendly stability framework for self-learning control (SLC) laws of the form $u(t) = k_1(t)u(t-\tau) + k_2 v(t)$. Uniform ultimate boundedness is established for a class of general linear plants under two engineering actuator assumptions: (i) smoothness and (ii) saturation with maximum value. These assumptions are reasonable for practical systems and yield an explicit bound, which converts the delay learning mechanism into a nominal (delay-free) controller plus a bounded perturbation injection. The most notable feature of SLC is its simplicity of structure coupled with excellent performance. It is compatible with traditional algorithms and can enhance even PD-type controllers. A complete design procedure and a spacecraft attitude tracking simulation example are provided to demonstrate compatibility with aerospace applications while presenting a broadly applicable theoretical result.

Keywords: self-learning control, uniform ultimate boundedness, smooth saturated actuator, spacecraft attitude control.

1 Introduction

1.1 Motivation

Many aerospace control systems operate under strict computational and energy constraints, motivating control laws that are both resource-aware and implementable on embedded flight hardware [1, 12]. Moreover, practical actuation is rarely discontinuous: for actuators such as reaction wheels and control moment gyros, the commanded torque is bounded by saturation and evolves smoothly due to rate limits, bandwidth constraints, and internal motor/drive dynamics [2, 4]. This practical actuation property is not merely a modeling convenience; it represents a physically grounded prior shared by a broad class of engineered systems, which can be exploited in control design [3].

From a control-theoretic perspective, actuator limits and smoothness constraints can be viewed as hard input bounds and (possibly) input-derivative constraints, both of which directly affect stability margins, transient performance, and achievable tracking accuracy [2, 5]. Neglecting these constraints may lead to unrealistic torque profiles, degraded on-orbit performance, or even instability when implemented on real hardware. Consequently, a substantial body of literature incorporates saturation handling (e.g., anti-windup and constrained control) and actuator dynamics into the synthesis process to ensure feasibility and robustness under realistic actuation [6–8]. These considerations motivate control

Submitted: 21 November 2025
Accepted: 12 January 2026
Published: 29 January 2026

Vol. 1, **No.** 1, 2026.
10.62762/AEC.2025.320719

***Corresponding author:**
✉ Chengxi Zhang
dongfangxy@163.com

Citation
Zhang, C. (2026). Self-Learning Control under Practical Actuation. *Aerospace Engineering Communications*, 1(1), 36–46.

designs that explicitly respect bounded, smooth actuation while achieving high-precision attitude regulation and tracking in the presence of disturbances and modeling uncertainty [9].

In this context, self-learning control (SLC) is particularly appealing, formerly called online learning control in [10]. Instead of introducing elaborate adaptive/observer structures, SLC updates the current command by reusing a portion of past control information through a simple algebraic relation, leading to low implementation complexity and modest computational overhead [11]. While time-delay and iterative learning ideas have been extensively studied in aerospace attitude regulation and beyond, delay is most often treated as a harmful effect to be suppressed (e.g., as uncertainty or a disturbance). By contrast, SLC treats delayed/iterative information as a useful signal: when actuation is smooth and bounded, past input-output data become a reliable resource that can be leveraged to improve tracking and robustness in a systematic yet mechanistically simple manner. These features make SLC under practical actuation both broadly relevant and scientifically valuable.

1.2 Related Work

This paper provides a stability analysis framework for SLC applied to general linear systems

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and (A, B) is stabilizable. A typical SLC law takes the form [11]

$$u(t) = k_1(t)\, u(t-\tau) + k_2\, v(t), \tag{2}$$

where $\tau > 0$ is named the *Learning Interval*; in previous investigations, $k_1(t) \in (0, 1)$ is called the *Learning Intensity* and is set to constant (possibly time-varying, such as in [11, 13]); and $v(t)$ is a stabilizing *baseline* control. When the tracking error is large, (2) can lead to rapid growth in $u(t)$ and possible saturation. A variable learning intensity (VLI) mechanism alleviates this by reducing $k_1(t)$ when $u(t-\tau)$ is large [11]. A fully developed stability analysis framework of SLC for general linear systems is provided. To this end, the learning-difference transformation is introduced [11]:

$$\tilde{u}(t) := u(t) - u(t-\tau), \tag{3}$$

which is also a key idea in our proposed SLC method. This operator explicitly quantifies how much the commanded input changes over one learning interval window. Under *practical actuation*, the command signal delivered by real actuators is typically bounded and bandwidth-limited; hence $u(t)$ is expected to vary smoothly rather than exhibit abrupt jumps. Consequently, for a sufficiently small τ, the learning difference $\tilde{u}(t)$ is naturally small and can be regarded as a structured, physically interpretable *incremental input* rather than an arbitrary disturbance. This makes (3) a suitable interface between physics-driven actuation constraints and learning-based control: it allows past command information to be reused while keeping the induced learning term consistent with actuator smoothness, which is central to the proposed self-learning control design under practical actuation.

1.3 Contribution

The main contributions are threefold:

1) Physics- and engineering-grounded learning assumption. A rigorous connection is established between standard actuation properties-bounded input $\|u\| \le u_{\max}$ and smooth variation $\|\dot{u}\| \le L\|u\|$, and the key self-learning quantity, yielding the explicit learning difference bound $\|\tilde{u}\| \le L\tau u_{\max}$. This result turns the learning-difference condition from a heuristic requirement into a verifiable consequence of practical actuation.

2) Complete stability theory for SLC in linear systems. A systematic uniform ultimate boundedness (UUB) analysis is developed for the closed loop with a nominal stabilizer and a self-learning controller. The proof avoids delay-dependent LMI machinery and instead follows a transparent nominal-plus-perturbation argument that clarifies how the learning term affects stability and ultimate bounds.

3) Implementation simplicity and an end-to-end engineering pipeline. The resulting controller has a minimal structure: the learning update is an algebraic reuse of past commands, and the nominal feedback can be as simple as a PD-type stabilizer. The theoretical results are complemented by a complete design-and-simulation workflow, demonstrated through a spacecraft attitude tracking example that illustrates the effectiveness of the approach.

1.4 Organization

Section II formulates the problem and assumptions. Section III introduces the learning difference reformulation and establishes key bounds. Section IV presents the main UUB stability theorem and a practical tuning

guideline. Section V provides a spacecraft attitude simulation example. Section VI concludes this paper.

2 Self-Learning Control Fundamentals

2.1 Self-Learning Control Law

We employ an SLC/VLI-SLC structure that explicitly separates a *learning item* and an *updating item*:

$$u(t) = \underbrace{K_1(t)\, u(t-\tau)}_{\text{learning item}} + \underbrace{K_2\, v(t)}_{\text{updating item}} \tag{4}$$

where $\tau > 0$ is the learning interval. The matrix $K_1(t) \in \mathbb{R}^{m\times m}$ is a (possibly time-varying) *learning intensity* that reuses past actuation commands and thereby injects accumulated control experience into the current input. The matrix $K_2 \in \mathbb{R}^{m\times m}$ weights the *updating item*, which provides the instantaneous correction based on the current state.

For simplicity and in line with common SLC/VLI constructions, $K_1(t)$ is assumed to be diagonal (component-wise learning); the analysis can be extended to general bounded $K_1(t)$. The baseline control $v(t)$ is chosen as a simple feedback law,

$$v(t) = -Kx(t), \tag{5}$$

with $K \in \mathbb{R}^{m\times n}$. In this form, the learning item captures previous, experience-based reuse of prior commands, whereas the updating item can be implemented by a minimal stabilizer (e.g., a PD-type feedback), highlighting the modularity and implementation simplicity of the SLC architecture.

2.2 Engineering Actuator Assumptions

Assumption 1 (*Smooth and Saturated Actuation*). The actual control command $u(t)$ satisfies, for all $t \geq 0$,

$$\|\dot{u}(t)\| \leq L\,\|u(t)\|\,, \tag{6}$$
$$\|u(t)\| \leq u_{\max}, \tag{7}$$

for some constants $L > 0$ and $u_{\max} > 0$.

Remark 1. Assumption 2.2 formalizes two ubiquitous physical features of real actuation: *magnitude limits* and *rate/bandwidth limits*. The saturation bound (7) is routinely enforced in practice by command limiting or hardware constraints. The smoothness condition (6) captures the fact that many actuators cannot change their output arbitrarily fast; it is consistent with bandwidth-limited devices and naturally arises from internal actuator dynamics, drive electronics, and command filters.

Although motivated by aerospace systems (e.g., reaction wheels), the same premise applies far more broadly. Electric drives and motor-driven mechanisms exhibit finite acceleration and torque slew; compliant and mechanical subsystems (e.g., spring–mass–damper elements) produce forces that evolve continuously under continuous excitation; and, more generally, most engineered physical systems do not admit discontinuous actuation in real time. Therefore, Assumption 2.2 should be viewed as an engineering-grounded prior rather than an application-specific restriction, and it provides a broad foundation for the proposed self-learning control framework under practical actuation.

2.3 Control Objective

Our goal is to establish uniform ultimate boundedness of the closed-loop system under (4)–(5) and Assumption 2.2. In particular, we aim to show that the state remains bounded asymptotically, i.e.,

$$\limsup_{t\to\infty} \|x(t)\| \leq r, \tag{8}$$

for some explicit constant $r > 0$ determined by τ, L, $u_{\max}$, and the controller gains, and to provide an explicit bound $r = r(\tau, L, u_{\max}, K_1, K_2, K)$, i.e., we aim to construct a Lyapunov function $V(x)$ and constants $a > 0$, $b \geq 0$ s.t. along closed-loop solutions

$$\dot{V}(x(t)) \leq -a\, V(x(t)) + b, \tag{9}$$

with $b = b(\tau, L, u_{\max}, K_1, K_2, K)$, which directly yields an ultimate bound.

3 Learning-Difference Reformulation and Key Bounds for Self-Learning Control

3.1 Learning Difference Transformation

Define the *learning difference* (3). Then, we have

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

Then, we present the bound on learning difference from an engineering scenario.

Lemma 1 (*Learning-Difference Bound*). Under Assumption 2.2, the learning difference satisfies

$$\|\tilde{u}(t)\| \leq \alpha_\tau := L\tau u_{\max}, \quad \forall t \geq 0. \tag{11}$$

Proof: Seeing the learning difference definition, we have

$$\tilde{u}(t) = u(t) - u(t-\tau) = \int_{t-\tau}^{t} \dot{u}(s)\, ds. \tag{12}$$

Therefore, from Assumption 1, we have

$$\|\tilde{u}(t)\| \le \int_{t-\tau}^{t} \|\dot{u}(s)\|\, ds \le \int_{t-\tau}^{t} L\,\|u(s)\|\, ds \le \int_{t-\tau}^{t} L u_{\max}\, ds = L\tau u_{\max}. \tag{13}$$

3.2 Self-Learning Control Law

Substituting (10) into (4) gives

$$u(t) = K_1(t)\big(u(t) - \tilde{u}(t)\big) + K_2 v(t), \tag{14}$$

hence

$$\big(I - K_1(t)\big)u(t) = K_2 v(t) - K_1(t)\tilde{u}(t). \tag{15}$$

Assumption 2 (*Non-singularity of Learning Map*). There exist constants $0 < \underline{k} < \bar{k} < 1$ s.t. for all t,

$$\underline{k} I \preceq K_1(t) \preceq \bar{k} I, \tag{16}$$

so that $I - K_1(t)$ is invertible and

$$\big\|(I - K_1(t))^{-1}\big\| \le \frac{1}{1-\bar{k}}. \tag{17}$$

Define

$$\kappa_1(t) := (I - K_1(t))^{-1} K_2, \tag{18}$$

$$\kappa_2(t) := (I - K_1(t))^{-1} K_1(t). \tag{19}$$

Then from (15),

$$u(t) = \kappa_1(t) v(t) - \kappa_2(t)\tilde{u}(t). \tag{20}$$

Remark 2. Eq. (20) is the core simplification: the self-learning control law (with conventional *time-delay* item) becomes a delay-free algebraic equation with a bounded perturbation term $-\kappa_2(t)\tilde{u}(t)$. The size of this perturbation is controlled by α_τ and the learning intensity bounds.

4 Stability Analysis for Linear Systems

4.1 Closed-Loop Dynamics

Using (5) and (20), we obtain

$$u(t) = -\kappa_1(t) K x(t) - \kappa_2(t)\tilde{u}(t). \tag{21}$$

Substituting into (1) gives

$$\dot{x}(t) = \Big(A - B\kappa_1(t)K\Big)x(t) - B\kappa_2(t)\tilde{u}(t). \tag{22}$$

4.2 Nominal Stabilization Condition

We choose K (and possibly K_2) so that the nominal matrix

$$A_c := A - B\bar{\kappa}_1 K \tag{23}$$

is Hurwitz for a representative constant $\bar{\kappa}_1$ (e.g., worst-case gain upper bound), and we employ a quadratic Lyapunov function for robustness to the bounded injection.

To make this explicit, define bounds from Assumption 3.2. If $K_2 = k_2 I$ with $k_2 > 0$ and $K_1(t) \preceq \bar{k} I$, then

$$\|\kappa_1(t)\| \le \frac{k_2}{1-\bar{k}}, \qquad \|\kappa_2(t)\| \le \frac{\bar{k}}{1-\bar{k}}. \tag{24}$$

4.3 Uniform Ultimate Boundedness

Theorem 1(*UUB Stability*). Consider (1)–(5)–(4). Suppose Assumptions 2.2 and 3.2 hold. Assume there exists $P \succ 0$ and $Q \succ 0$ s.t. for all t,

$$\Big(A - B\kappa_1(t)K\Big)^{\mathrm{T}} P + P\Big(A - B\kappa_1(t)K\Big) \preceq -Q. \tag{25}$$

Then the closed-loop state $x(t)$ is uniformly ultimately bounded. Moreover, defining

$$c_Q := \lambda_{\min}(Q),\ c_P := \lambda_{\max}(P),\ c_{PB} := \|PB\|, \tag{26}$$

and using $\|\kappa_2(t)\| \le \bar{\kappa}_2$ and Lemma 3.1, the ultimate bound satisfies

$$\limsup_{t\to\infty} \|x(t)\| \le \frac{2c_{PB}\,\bar{\kappa}_2}{c_Q}\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}\,\alpha_\tau = \frac{2c_{PB}\,\bar{\kappa}_2}{c_Q}\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}\,L\tau u_{\max}. \tag{27}$$

Proof. Let $V(x) = x^{\mathrm{T}} P x$. Along (22),

$$\dot{V} = x^{\mathrm{T}}\Big(\big(A_\kappa\big)^{\mathrm{T}} P + P\big(A_\kappa\big)\Big)x - 2x^{\mathrm{T}} PB\kappa_2\tilde{u} \le -x^{\mathrm{T}} Q x + 2\,\|x\|\,\|PB\|\,\|\kappa_2\|\,\|\tilde{u}\|.$$

where $A_\kappa = A - B\kappa_1 K$. Using $x^{\mathrm{T}} Q x \ge c_Q\,\|x\|^2$, $\|\kappa_2\| \le \bar{\kappa}_2$, and $\|\tilde{u}\| \le \alpha_\tau$,

$$\dot{V} \le -c_Q\,\|x\|^2 + 2c_{PB}\bar{\kappa}_2\alpha_\tau\,\|x\|. \tag{28}$$

By Young's inequality, for any $\varepsilon > 0$,

$$2c_{PB}\bar{\kappa}_2\alpha_\tau\|x\| \le \frac{\varepsilon}{2}\|x\|^2 + \frac{1}{2\varepsilon}\big(2c_{PB}\bar{\kappa}_2\alpha_\tau\big)^2. \tag{29}$$

Choosing $\varepsilon = c_Q$ yields

$$\dot{V} \le -\frac{c_Q}{2}\|x\|^2 + \frac{2c_{PB}^2\bar{\kappa}_2^2}{c_Q}\alpha_\tau^2. \tag{30}$$

Using $\|x\|^2 \ge V/\lambda_{\max}(P)$, it follows that

$$\dot{V} \le -\frac{c_Q}{2\lambda_{\max}(P)}V + \frac{2c_{PB}^2\bar{\kappa}_2^2}{c_Q}\alpha_\tau^2, \tag{31}$$

which implies uniform ultimate boundedness. Let

$$a := \frac{c_Q}{2\lambda_{\max}(P)} \tag{32}$$

$$b := \frac{2c_{PB}^2\bar{\kappa}_2^2}{c_Q}\alpha_\tau^2. \tag{33}$$

Then

$$\dot{V} \le -aV + b. \tag{34}$$

By the comparison lemma, for any $t \ge t_0$,

$$V(t) \le e^{-a(t-t_0)}V(t_0) + \frac{b}{a}\big(1 - e^{-a(t-t_0)}\big), \tag{35}$$

which implies $\limsup_{t\to\infty} V(t) \le \frac{b}{a}$ and hence the closed-loop system is uniformly ultimately bounded. Moreover, since $\lambda_{\min}(P)\|x\|^2 \le V$, it follows that

$$\begin{aligned}\limsup_{t\to\infty}\|x(t)\| &\le \sqrt{\frac{1}{\lambda_{\min}(P)}\frac{b}{a}} \\ &= \frac{2c_{PB}\bar{\kappa}_2}{c_Q}\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}\,\alpha_\tau.\end{aligned} \tag{36}$$

4.4 Parameter Tuning Guidelines

A practical approach is to design a baseline gain K for a *fixed* nominal gain $\kappa_{1,0}$ (e.g., $\kappa_{1,0} = \frac{k_2}{1-\bar{k}}$) such that $A - B\kappa_{1,0}K$ is Hurwitz, then select P by solving the Lyapunov equation $(A - B\kappa_{1,0}K)^{\mathrm{T}}P + P(A - B\kappa_{1,0}K) = -Q$ for a chosen $Q \succ 0$. If $\kappa_1(t)$ varies in a small range, a common P may still satisfy (25) (quadratic stability over the gain range). This can be checked by evaluating the inequality at the extreme values of $\kappa_1(t)$ when $\kappa_1(t)$ is scalar/diagonal. From (27), the ultimate bound scales as

$$\limsup_{t\to\infty}\|x(t)\| = \mathcal{O}(L\tau u_{\max}). \tag{37}$$

Hence:

- Smaller learning interval τ improves the ultimate bound and reduces the injection magnitude.
- Smaller learning intensity upper bound $\bar{k}$ reduces $\bar{\kappa}_2 = \frac{\bar{k}}{1-\bar{k}}$, improving robustness.
- Larger c_Q (faster nominal convergence) reduces the ultimate bound, motivating more aggressive baseline feedback K within actuator limits.

Remark 3. The way the Analysis Avoids Delay LMIs. The key step is treating the learning term through the identity $u(t-\tau) = u(t) - \tilde{u}(t)$, turning the delay mechanism into a bounded injection $-\kappa_2(t)\tilde{u}(t)$. Under smoothness and saturation, $\tilde{u}$ admits an explicit bound proportional to τ. This enables a Lyapunov argument similar to input-to-state stability estimates without constructing Lyapunov-Krasovskii functionals.

5 Application Example: Spacecraft Attitude Tracking Simulation

This section illustrates how the general linear theory can guide an aerospace attitude control simulation. The simulation uses a nonlinear spacecraft model, while the analysis is presented in a general linear setting; this is typical when the main theoretical chapter targets a broad class and the application validates engineering feasibility.

5.1 Spacecraft Attitude Dynamics Model

Remark 4. While the theoretical analysis in Sections 3–4 focuses on linear systems for generality, the following simulation demonstrates the engineering feasibility and performance of SLC/VLI-SLC when applied to a nonlinear spacecraft attitude tracking problem. We consider rigid spacecraft attitude dynamics [11]

$$J\dot{\omega} = -\omega^{\times}J\omega + u + d, \tag{38}$$

$$\dot{q} = \frac{1}{2}\big(q^{\times} + q_0 I_3\big)\omega, \tag{39}$$

$$\dot{q}_0 = -\frac{1}{2}q^{\mathrm{T}}\omega, \tag{40}$$

where $\omega \in \mathbb{R}^3$ is angular velocity, $Q = [q^{\mathrm{T}}, q_0]^{\mathrm{T}}$ is the unit quaternion, and $u \in \mathbb{R}^3$ is the control torque.

5.2 Tracking Errors and Baseline Controller

Define attitude and angular velocity tracking errors (details omitted for brevity; standard quaternion error mapping is used). Let $s := \omega_e + \sigma q_e$ with $\sigma > 0$.

Choose a baseline stabilizer

$$v(t) = -k_3\,\phi(\omega)\,s(t), \tag{41}$$

$$\phi(\omega) = \|\omega\|^2 + \|\omega\| + 1, \tag{42}$$

and apply the SLC update component-wise:

$$u_i(t) = k_{1,i}(t)u_i(t-\tau) + k_2 v_i(t), \quad i = 1,2,3. \tag{43}$$

A common VLI mapping is

$$k_{1,i}(t) = \exp\left(-\gamma_1 \frac{|u_i(t-\tau)| + \varepsilon}{\gamma_2}\right) \in (0,1), \quad (44)$$

with $\gamma_1, \gamma_2, \varepsilon > 0$.

5.3 Simulation Setup

Inertia and disturbance:

- J is symmetric positive definite; use nominal plus uncertainty if desired.
- $d(t)$ is bounded (e.g., combination of sinusoidal terms).

Actuator non-idealities: saturation, dead-zone, bias torque, and time-delay may be included in the actuator model; command limiting enforces $\|u(t)\| \le u_{\max}$.

Parameter selection: Choose τ small (e.g., one sampling period), set ε small to avoid $k_{1,i}(t) \to 1$, and tune k_2, k_3, σ to obtain acceptable tracking and torque margins.

5.4 Metrics

Tracking performance is measured by $\|\omega_e(t)\|$ and attitude error angles. Control saturation weakening is quantified by the percentage of time at saturation and a cumulative energy index

$$E_i = \frac{\int_0^T |u_i(t)|dt}{\int_0^T |u_{i,\text{baseline}}(t)|dt}.$$

5.5 Discrete-Time Implementation

In practical digital controllers, the learning interval τ is an integer multiple of the sampling period T_s. Let k be the time index and $\tau = NT_s$

$$u[k] = K_1[k]\, u[k-N] + K_2\, v[k]. \quad (45)$$

The learning difference is

$$\tilde{u}[k] = u[k] - u[k-N]. \quad (46)$$

The VLI map can be implemented via lookup tables to reduce runtime computation. See Algorithm 1.

Algorithm 1: SLC with Learning-Difference View

1: Choose sampling period T_s and learning steps N ($\tau = NT_s$).
2: Select baseline controller $v[k]$ (42).
3: Choose K_2 and VLI parameters $(\gamma_1, \gamma_2, \varepsilon)$ to enforce $0 < k_{1,i}(k) < 1$.
4: **for** $k = 0, 1, 2, \ldots$ **do**
5: Measure/estimate $x[k]$ (or (q, ω)) and compute $v[k]$.
6: Compute learning intensity $K_1[k]$ (fixed or VLI).
7: Update $u[k] = K_1[k]u[k-N] + K_2 v[k]$.
8: Apply limits to enforce $\|u[k]\| \le u_{\max}$.
9: **end for**

5.6 Simulation Results

5.6.1 Simulation setup

A nonlinear spacecraft attitude tracking problem is considered to evaluate the proposed variable learning intensity scheme. The rigid-body attitude dynamics are described by

$$J\dot{\omega} = (J\omega) \times \omega + u + d, \quad (47)$$

where $\omega \in \mathbb{R}^3$ is the body angular velocity, $J \in \mathbb{R}^{3\times3}$ is the (possibly uncertain) inertia matrix, $u \in \mathbb{R}^3$ is the control torque, and $d \in \mathbb{R}^3$ denotes exogenous disturbances.

The attitude is represented by the unit quaternion $Q = \text{col}(q, q_0)$ and evolves according to the standard quaternion kinematics. The desired angular velocity is selected as

$$\omega_d(t) = 10^{-2}[\cos\psi, -\sin\psi, -\cos\psi]^\top, \ \psi = 0.1t. \quad (48)$$

The disturbance torque is chosen as

$$d(t) = 10^{-3}\begin{bmatrix} -3\cos(\varphi t) - 6\sin(0.3\varphi t) + 3 \\ 1.5\sin(\varphi t) - 3\cos(0.5\varphi t) - 2 \\ -3\sin(\varphi t) + 8\sin(0.4\varphi t) - 1 \end{bmatrix},$$
$$\varphi = 0.5 + \|\omega\|. \quad (49)$$

To test robustness against inertia uncertainty, the inertia matrix is modeled as

$$J(t) = \bar{J} + \Delta J(t), \quad (50)$$

where

$$\bar{J} = \begin{bmatrix} 20 & 2 & 0.9 \\ 2 & 17 & 0.5 \\ 0.9 & 0.5 & 15 \end{bmatrix}, \quad (51)$$

$$\Delta J(t) = \text{diag}\begin{pmatrix} (3+\sin(0.5t))e^{-0.1t} + 1 \\ (4+\cos(0.5t))e^{-0.1t} + 2 \\ (5+\sin(0.5t))e^{-0.1t} - 1 \end{pmatrix} \quad (52)$$

The initial conditions are set to $\omega(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$ and

$$Q(0) = \text{col}\left(\begin{bmatrix} -0.2 & 0.3 & -0.2 \end{bmatrix}^\top, \sqrt{0.83}\right). \quad (53)$$

The simulation runs for $T = 800$ s with sampling period $T_s = 0.01$ s. The learning interval is chosen as $\tau = 0.05$ s.

Table 1. Energy and tracking performance summary (last 50 s for RMS metrics).

Method	E_x	E_y	E_z	RMS($\|\omega_e\|$) (rad/s)	RMS(θ_e) (rad)	$\max_t \|u(t)\|_\infty/u_{\max}$
Non-SLC	1.0000	1.0000	1.0000	4.342569×10^{-3}	3.527421×10^{-2}	0.5471
SLC	0.9874	0.9787	0.9727	3.510444×10^{-4}	2.740603×10^{-3}	0.9500
VLI-SLC	0.9059	0.9110	0.8831	2.140248×10^{-4}	1.604341×10^{-3}	0.6119

5.6.2 *Controllers for comparison*

Three controllers are compared:

1) Non-SLC (baseline without self-learning). The learning term is removed and the control command is $u(t) = k_2 v(t)$.

2) SLC (fixed learning intensity). The fixed-intensity learning controller is implemented as

$$u(t) = k_1 u(t-\tau) + k_2 v(t), \quad k_1 = 0.9. \tag{54}$$

3) VLI-SLC (variable learning intensity). The variable intensity learning controller is given by

$$u_i(t) = k_{1,i}(t) u_i(t-\tau) + k_2 v_i(t), \; i = 1,2,3, \tag{55}$$

with

$$k_{1,i}(t) = \exp(-\gamma_1 \left(|u_i(t-\tau)| + \varepsilon\right)^{\gamma_2}). \tag{56}$$

The update term uses the sliding-like variable $s = \omega_e + \sigma q_e$ and

$$\begin{aligned} v(t) &= -k_3 \varsigma(t) s(t), \\ \varsigma(t) &= \|\omega(t)\|^2 + \|\omega(t)\| + 1. \end{aligned} \tag{57}$$

Unless otherwise stated, the parameters are selected as: $\gamma_1 = 4$, $\gamma_2 = 2$, $\varepsilon = 0.1$, $k_2 = 1$, $k_3 = 2$, $\sigma = 1$.

5.6.3 *Non-ideal actuator model and energy index*

To better reflect practical implementation conditions, a non-ideal actuator model is included, consisting of a small input delay, second-order actuator dynamics, bias torque, dead-zone nonlinearity, saturation, and an actuator health coefficient. The saturation bounds are set to ± 1 N $\cdot$ m. Please see *Actuator model for spacecraft attitude control simulation* for reference.

To quantify energy consumption, the energy index is computed as

$$E_i = \frac{P_i}{P_{i,\mathrm{Non}}}, \; P_i = \int_0^T |u_i(t)|\, dt, \; i \in \{x, y, z\}, \tag{58}$$

where $P_{i,\mathrm{Non}}$ is the corresponding baseline (Non-SLC) energy.

5.6.4 *Results and discussion*

Figures 1 and 2 illustrate the angular velocity tracking error and attitude error angle, respectively. All controllers achieve stable tracking under the considered inertia uncertainty and external disturbances. Compared with Non-SLC, both learning-based methods (SLC and VLI-SLC) provide significantly improved steady-state accuracy, demonstrating that the learning term effectively leverages past control information to enhance error compensation.

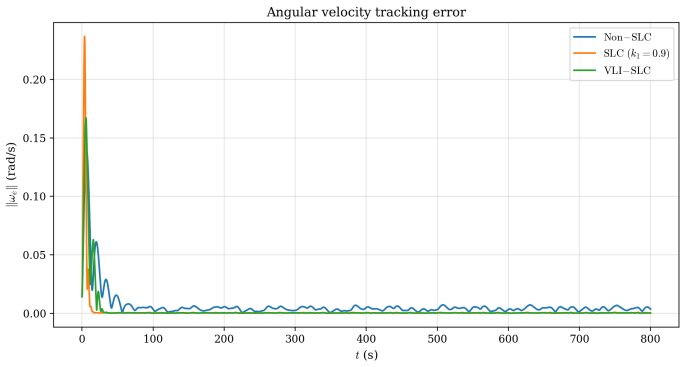

Figure 1. Angular velocity tracking error norm $\|\omega_e\|$.

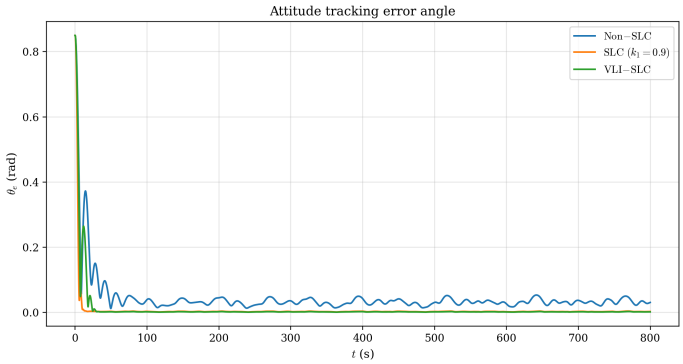

Figure 2. Attitude tracking error angle θ_e.

Figure 3 shows the actuator output torques. While SLC improves tracking accuracy, it exhibits a pronounced initial torque peak due to the fixed learning intensity, which increases the risk of hitting actuator saturation when the initial tracking errors are large.

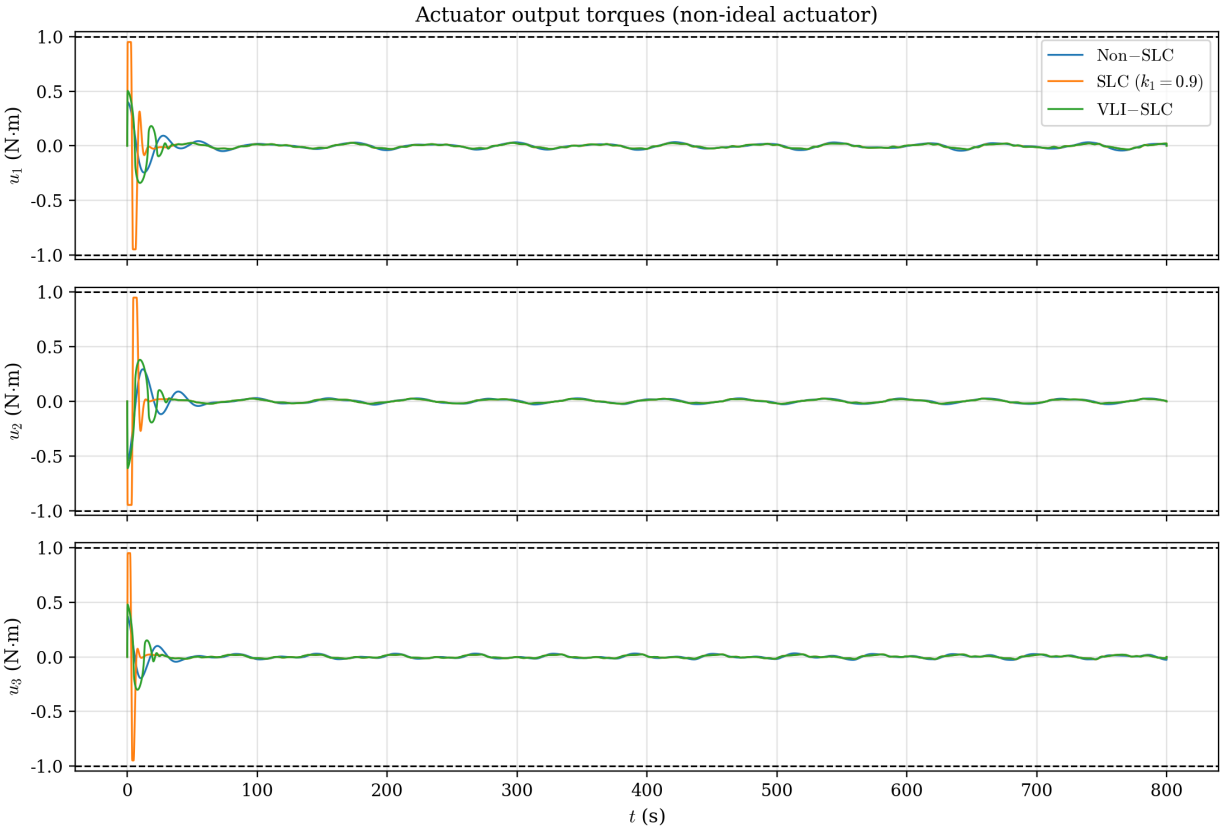

Figure 3. Actuator output torques under the non-ideal actuator model. Dashed lines indicate saturation bounds.

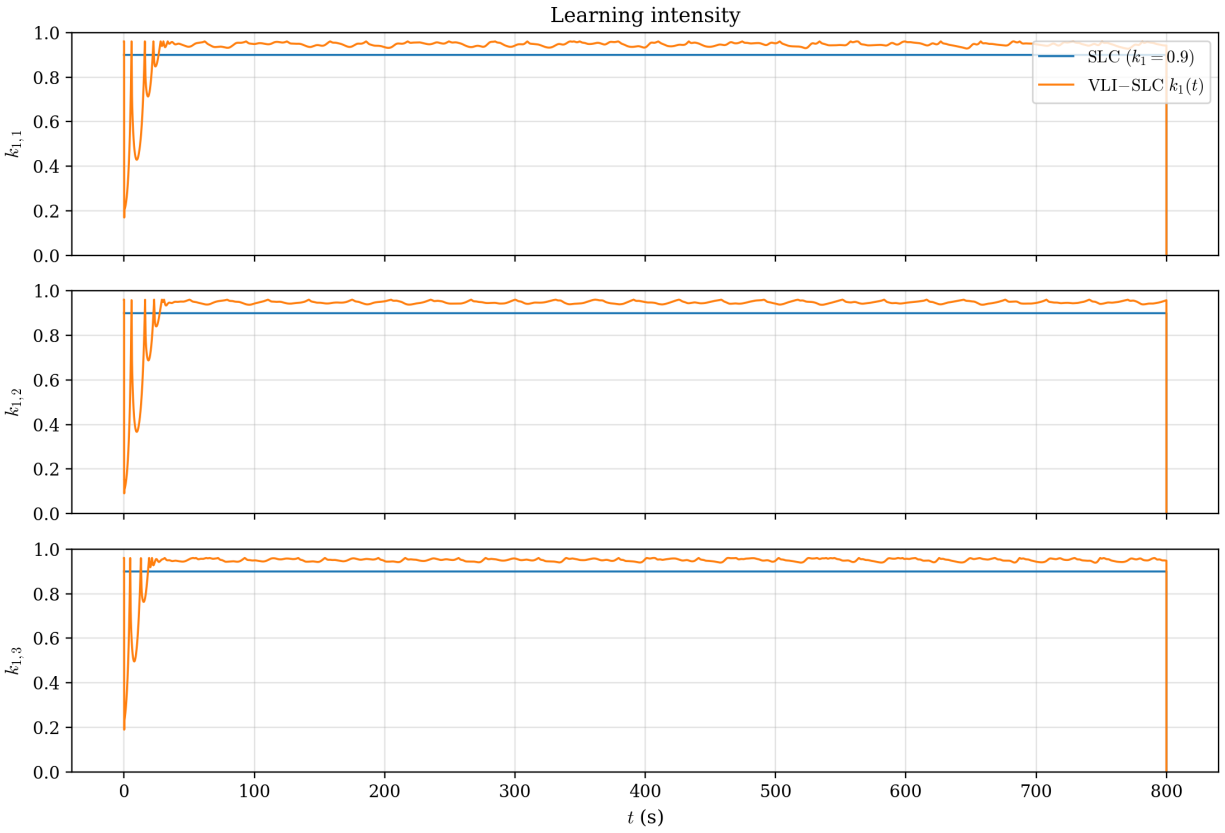

Figure 4. Learning intensity comparison: constant intensity in SLC versus adaptive intensities in VLI-SLC.

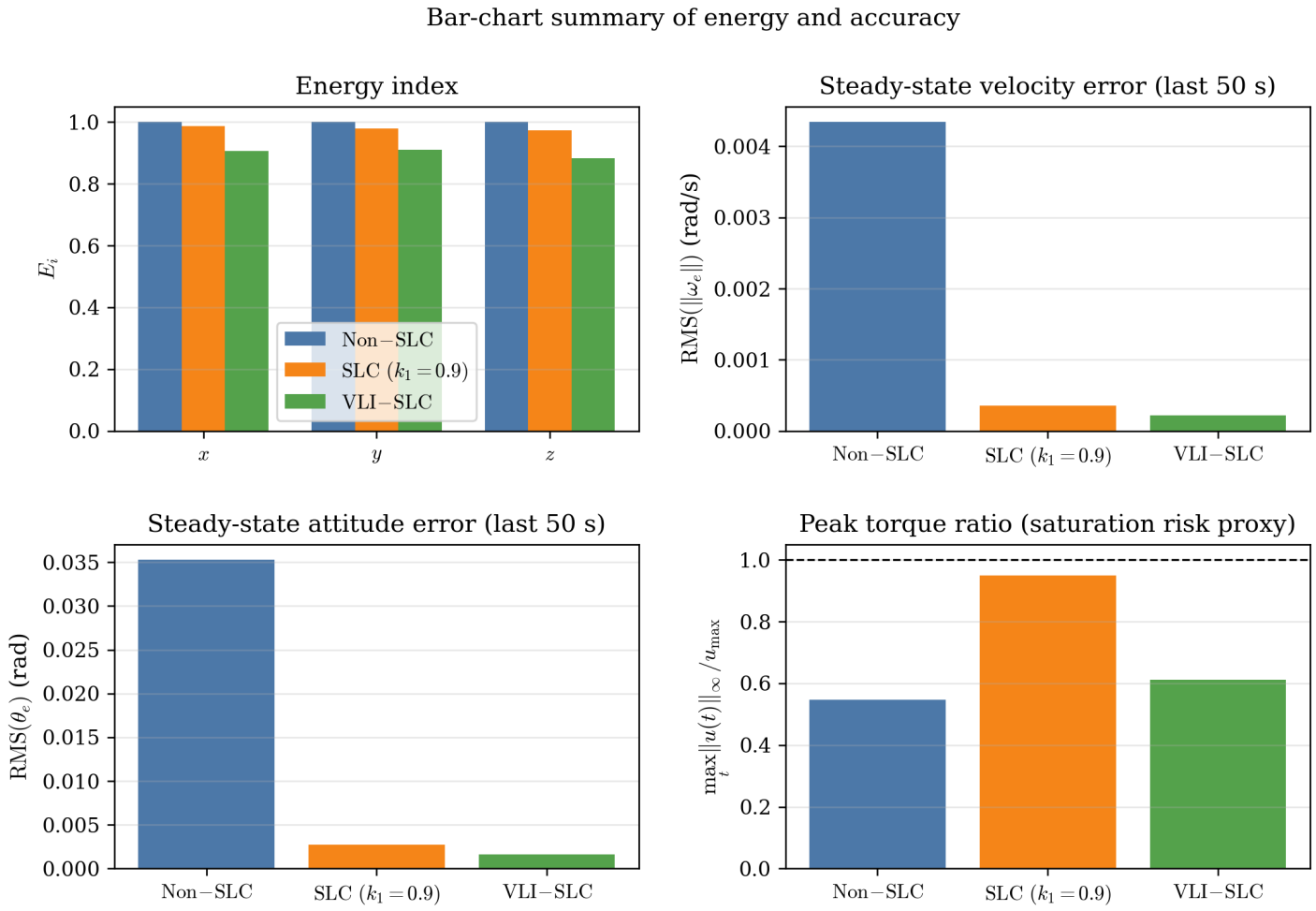

Figure 5. Bar-chart summary of energy index, steady-state accuracy (RMS), and peak torque ratio.

In contrast, VLI-SLC markedly reduces the peak torque demand in the initial transient, while maintaining comparable steady-state tracking accuracy. This improvement is attributed to the adaptive reduction of $k_{1,i}(t)$ when $|u_i(t-\tau)|$ becomes large, which weakens the aggressive learning behavior in high-torque regimes.

The evolution of learning intensities is shown in Figure 4. SLC keeps a constant k_1, whereas VLI-SLC automatically decreases $k_{1,i}(t)$ during large-torque transients and increases it when the control effort is small, thereby balancing transient saturation avoidance and steady-state learning benefits.

Finally, Figure 5 summarizes energy and accuracy metrics. VLI-SLC achieves a favorable trade-off: it preserves the steady-state accuracy of SLC, reduces the peak torque ratio (a proxy for saturation risk), and yields improved energy indices relative to fixed-intensity SLC in the considered scenario.

5.6.5 *Summary*

Table 1 reveals a clear trade-off between energy behavior and tracking accuracy across the three methods. Regarding the energy indices $E = [E_x, E_y, E_z]$, Non-SLC maintains $E_x = E_y = E_z = 1$, indicating the best energy preservation (i.e., minimal additional dissipation). SLC exhibits a mild reduction ($E \approx 0.97$–0.99), suggesting that improved tracking is achieved at the cost of only a small energy loss. VLI-SLC shows a more pronounced decrease ($E \approx 0.88$–0.91), consistent with stronger dissipation or more aggressive regulation introduced to enhance closed-loop error suppression.

In terms of RMS errors over the last 50 s, SLC reduces $\mathrm{RMS}(\|\omega_e\|)$ from 4.34×10^{-3} to 3.51×10^{-4} (about a $12\times$ reduction) and $\mathrm{RMS}(\theta_e)$ from 3.53×10^{-2} to 2.74×10^{-3} (about a $13\times$ reduction) compared with Non-SLC, i.e., roughly an order-of-magnitude improvement. VLI-SLC further decreases $\mathrm{RMS}(\|\omega_e\|)$

to 2.14×10^{-4} and $\mathrm{RMS}(\theta_e)$ to 1.60×10^{-3}, yielding an additional $\sim$1.6–1.7$\times$ improvement relative to SLC, which indicates stronger steady-state/late-stage error attenuation.

The peak control ratio (peak_ratio) is highest for SLC (0.9500), implying that its control input operates closest to the saturation limit, consistent with its substantial error reduction. Non-SLC is more conservative (0.5471), corresponding to noticeably larger RMS errors. VLI-SLC achieves the smallest RMS errors with a moderate peak ratio (0.6119), suggesting that its performance gains are not solely due to larger control magnitude but also to improved control effectiveness; however, this comes with the largest reduction in the energy indices.

Overall, Non-SLC favors energy preservation at the expense of tracking performance; SLC trades near-saturation peak actuation for an order-of-magnitude error reduction with only minor energy loss; and VLI-SLC provides the best RMS tracking with moderate peak usage but incurs the greatest energy decrease.

6 Conclusion

A general learning-difference reformulation was developed for self-learning control laws and applied to linear plants. Under practical actuator smoothness and saturation assumptions, an explicit bound $\|\tilde{u}\| \leq L\tau u_{\max}$ is obtained, which converts the delay learning mechanism into a nominal stabilizing controller plus a bounded perturbation injection. A Lyapunov-based analysis yields uniform ultimate boundedness and an explicit ultimate bound scaling with $L\tau u_{\max}$. A spacecraft attitude simulation setup was outlined to demonstrate the method in an aerospace context. In our future work, the differential learning reconstruction framework is extended from linear systems to nonlinear spacecraft attitude dynamics models, and errors are further reduced by optimizing the learning parameters.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported by the National Natural Science Foundation of China under Grant 62573211.

Conflicts of Interest

The author declares no conflicts of interest.

AI Use Statement

The author declares that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Wu, B., & Cao, X. (2018). Robust attitude tracking control for spacecraft with quantized torques. *IEEE Transactions on Aerospace and Electronic Systems, 54*(2), 1020–1028. [CrossRef]

[2] Kristiansen, R., & Hagen, D. (2009). Modelling of actuator dynamics for spacecraft attitude control. *Journal of Guidance, Control, and Dynamics, 32*(3), 1022–1025. [CrossRef]

[3] Carrara, V., & Kuga, H. K. (2013). Torque and speed control loops of a reaction wheel. *In Proceedings of the International Conference on Vibration Problems*.

[4] Malekzadeh, M., & Sadeghian, H. (2017). Attitude control of spacecraft simulator with reaction wheels regulation. *In 2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM)* (pp. 101–106). [CrossRef]

[5] Wang, Z., Guo, Z., & Li, S. (2024). Finite-time disturbance rejection control for rigid spacecraft attitude set stabilization with actuator saturation. *IEEE Transactions on Aerospace and Electronic Systems, 60*(4), 4910–4922. [CrossRef]

[6] Tarbouriech, S., & Turner, M. (2009). Anti-windup design: an overview of some recent advances and open problems. *IET Control Theory & Applications, 3*(1), 1–19. [CrossRef]

[7] Chen, B., Shao, X., Yang, H., Li, D., & Hu, Q. (2025). Spacecraft anti-unwinding attitude tracking with guaranteed performance: A DREM-based adaptive control approach. *Advances in Space Research*. [CrossRef]

[8] Arimoto, S., Kawamura, S., & Miyazaki, F. (1984). Bettering operation of robots by learning. *Journal of Robotic systems, 1*(2), 123-140. [CrossRef]

[9] Javaid, U., Basin, M. V., & Ijaz, S. (2024, February). Spacecraft Attitude Stabilization Control Under Actuator Faults and Input Saturation. In *2024 IEEE 18th International Conference on Advanced Motion Control (AMC)* (pp. 1-6). IEEE. [CrossRef]

[10] Zhang, C., Xiao, B., Wu, J., & Li, B. (2021). On low-complexity control design to spacecraft attitude stabilization: an online-learning approach. *Aerospace Science and Technology, 110*, 106441. [CrossRef]

[11] Zhang, C., Ahn, C. K., Wu, J., & He, W. (2021). Online-learning control with weakened saturation response to attitude tracking: A variable learning intensity approach. *Aerospace Science and Technology, 117*, 106981. [CrossRef]

[12] Zheng, W. H., & Armstrong, J. T. (2010). Wireless intra-spacecraft communication: The benefits and the challenges. *In 2010 NASA/ESA Conference on Adaptive Hardware and Systems* (pp. 75–78). [CrossRef]

[13] Zhang, C., Lu, W., Zhao, S., Wu, J., Zhu, X., Liu, Z., & He, W. (2025). Enhancing attitude tracking with self-learning control using tanh-type learning intensity. *IEEE Transactions on Automation Science and Engineering, 22*, 16976–16986. [CrossRef]

Chengxi Zhang received B.S. and M.S. degrees from Harbin Institute of Technology, China, in 2012 and 2015; and Ph.D. degree from Shanghai Jiao Tong University, China, in 2019. He is now an Associate Professor at Jiangnan University, China. His research interests include space engineering, robotic systems, and control.

He is the Editor-in-Chief of *Aerospace Engineering Communications*. He is an Editor of *Frontiers in Aerospace Engineering*, *Frontiers in Mechanical Engineering*, and *Scientific Reports*, on the Editorial Board of *Symmetry* and *Aerospace Systems*. He is a Committee Member of the Youth Working Committee, Chinese Association of Automation since 2025. (Email: dongfangxy@163.com)