AppliedMath

MDPI

Article

Self-Learning Control for Multi-Agent Consensus

Chengxi Zhang

School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China; dongfangxy@163.com

Abstract

This paper addresses the consensus problem in multi-agent systems via a *self-learning control* scheme that directly reuses prior control information to accelerate transient coordination while maintaining robustness. I study agents with linear dynamics and external disturbances, and design a lightweight self-learning consensus control law for the distributed consensus domain, formulated as $u_i(t) = k_1\, u_i(t-\tau) + k_2\, s_i(t)$ with learning intensity k_1 and learning interval τ. I provide a Lyapunov-based stability proof showing uniform ultimate boundedness of the consensus error under bounded disturbances. Compared to non-learning consensus laws, the proposed strategy achieves faster agreement with reduced long-term effort and retains simplicity suitable for resource-constrained multi-agent platforms, while also achieving decent performance against external disturbances. Simulations validate the improved transient speed and steady accuracy. The full-version-source code is open-sourced.

Keywords: multi-agent systems; consensus control; self-learning control; robustness

1. Introduction

Consensus control of networked multiagent systems (MAS) has broad applications in robotic swarms, spacecraft formations, and distributed sensing [1–4]. Practical deployments require controllers that are simple in structure, robust to disturbances, and fast in transients. Classical consensus controllers use static or dynamic feedback gains; while effective, they often trade off convergence speed and robustness, and may demand significant computational or communication resources when augmented by adaptive/observer structures [5–7]. See reviews and survey for more [8,9].

Recent research on multi-agent systems has largely advanced by adding algorithmic complexity to meet coordination objectives. This trajectory often overlooks the value of simpler structures that reuse information already generated within the loop. Time delay illustrates this point. Classical designs regard delay as detrimental and seek to cancel or compensate it [10–12], which removes the informative trace of recent control effort. That trace encodes how the controller acted just before the present and is typically discarded. I explicitly retain and exploit this history. A conventional consensus law takes the form $u(t) = ks(t)$, akin to a minimal PID-like action. The proposed self-learning law $u(t) = k_1 u(t-\tau) + k_2 s(t)$ utilizes the delayed term as a compact memory [13] to stabilize transients and improve robustness without increasing structural complexity. The idea is intuitive and interesting. Consider backing a car into a garage. A helper rarely specifies exact distances to the driver in that car at each moment. The instruction is to "keep reversing", and only when the car reaches the desired position does the helper say "stop". The driver's recent action is continued until a condition triggers a change. Our controller follows the same principle: it carries forward the recent control action as learned

Academic Editor: Libor Pekař

Received: 8 October 2025
Revised: 5 February 2026
Accepted: 24 February 2026
Published: 3 March 2026

experience and adjusts only when the disagreement signal calls for it. This converts past actions into a lightweight prior that accelerates convergence while preserving simplicity. Therefore, the self-learning control (SLC) mechanism is very familiar with the former scenario: guidance proceeds by maintaining an effective action until a condition is met, rather than by prescribing precise increments at every instant (Due to disturbances, it is akin to aeroplanes: even when following identical flight paths, their control processes will surely differ). In this sense, the controller converts past actions into a lightweight prior that accelerates convergence while preserving simplicity.

Self-learning control, which leverages prior control information as a memory, provides a compelling alternative for accelerating transients without complex adaptation. The self-learning control paradigm showed that directly reusing previous control inputs via an algebraic recursion can enhance robustness and energy efficiency while preserving simplicity [13,14]. Its engineering value has been demonstrated in both unmanned rotorcraft [15] and gyroscope systems [16], achieving decent performances. In a related line, I developed learning observers and performance tuning policies for robust multiagent consensus with guaranteed boundedness under uncertainties [17]. Motivated by these findings, this paper develops a *self-learning control* consensus controller that integrates previous control information with current consensus errors. The design retains the algebraic simplicity of the learning concept, avoids the need for persistent excitation or heavy observers, and provably ensures the uniform ultimate boundedness of the consensus error using Lyapunov methods similar to those in our prior spacecraft control study. I also discuss parameter tuning to achieve fast convergence with weakened saturation tendencies.

The main contributions of this paper are:

(1) I propose a self-learning control consensus law for MAS that combines a learning term and an updating term, implemented via a single-step algebraic update with low computational cost.
(2) Using a Lyapunov function on the stacked consensus error, I derive uniform ultimate boundedness (UUB) stability under bounded disturbances, and provide transparent gain conditions linking learning intensity, interval, and consensus gains.
(3) Simulations show faster convergence and accuracy compared to non-learning baselines, resulting in an order-of-magnitude improvement over traditional algorithms. I present a practical tuning guideline to help readers implement the scheme.

2. Preliminary and Problem Formulation

2.1. Agent Dynamics with Disturbance

Consider N agents with dynamics for each $i \in 1, ..., N$

$$\dot{x}_i = Ax_i + Bu_i + d, \tag{1}$$

$$y_i = Cx_i, \tag{2}$$

where $x_i \in \mathbb{R}^n$, $u_i \in \mathbb{R}^m$, $y_i \in \mathbb{R}^p$. The disturbance $d \in \mathbb{R}^n$ is shared (or equivalently, an agent-wise bounded disturbance can be handled with minor notation changes) and satisfies

$$\|d(t)\| \leq \bar{d}, \quad \forall t \geq 0. \tag{3}$$

Assumption 1. *The pair (A, B) is stabilizable, and (A, C) is detectable.*

2.2. Graph and Consensus Error

I use graph theory to express the topology of the multi-agent system. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected undirected graph with $|\mathcal{V}| = N$. The weighted adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{N\times N}$ is defined by

$$a_{ij} = \begin{cases} > 0, & \text{if } (i,j) \in \mathcal{E},\ i \neq j, \\ 0, & \text{else,} \end{cases} \qquad \text{with } A = A^{\top},$$

in block form, i.e.,

$$A = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1N} \\ a_{21} & 0 & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & 0 \end{bmatrix}, \quad a_{ij} = a_{ji} \geq 0.$$

Let $D = \operatorname{diag}(\underline{d}_1, \ldots, \underline{d}_N)$ with

$$\underline{d}_i = \sum_{j=1}^{N} a_{ij}, \tag{4}$$

and define the Laplacian $L = D - A$.

Define the local consensus error (disagreement signal)

$$s_i(t) = \sum_{j=1}^{N} a_{ij}\big(x_i(t) - x_j(t)\big), \tag{5}$$

and the stacked vectors $x = \operatorname{col}(x_1, \ldots, x_N) \in \mathbb{R}^{Nn}$ and $s = \operatorname{col}(s_1, \ldots, s_N) \in \mathbb{R}^{Nn}$. We have the compact form

$$s(t) = \big(L \otimes I_n\big)\, x(t). \tag{6}$$

2.3. Control Objective

Here, I design a distributed control law $u_i(t)$ for each agent using only local information such that the consensus error $s(t)$ is UUB in the presence of bounded disturbances, and achieves fast transient convergence.

(1) Consensus in the disturbance-free case ($d_i \equiv 0$), i.e., consensus corresponds to $s(t) \to 0$ (equivalently, $x_i - x_j \to 0$ for all pairs): $\lim_{t\to\infty} \|s(t)\| = 0$.

(2) UUB under bounded disturbances: there are constants $b > 0$ and a class-$\mathcal{KL}$ function $\beta(\cdot,\cdot)$ such that for all initial conditions and all disturbances satisfying $\|d_i(t)\| \leq D$,

$$\|s(t)\| \leq \beta\big(\|s(0)\|, t\big) + b, \quad \forall\, t \geq 0,$$

hence there is a $T \geq 0$ (possibly dependent on $\|s(0)\|$) with

$$\|s(t)\| \leq b, \quad \forall\, t \geq T.$$

For connected $\mathcal{G}$, $\ker(L \otimes I_n) = \operatorname{span}\{\, \mathbf{1}_N \otimes v : v \in \mathbb{R}^n \,\}$, so $s(t) \to 0$ is equivalent to $x_i(t) - x_j(t) \to 0$ for all i, j.

3. Self-Learning Control for Multiagent Systems

3.1. Self-Learning Consensus Control Law

I propose the self-learning control scheme:

$$u_i(t) = \underbrace{k_1\, u_i(t-\tau)}_{Self-Learning} + \underbrace{k_2\, s_i(t)}_{Consensus}, \quad i = 1, \ldots, N, \tag{7}$$

where $k_1 \in (0,1)$ is the *learning intensity*, $k_2 < 0$ is the consensus gain, and $\tau > 0$ is the *learning interval*. The control law reuses the past input information $u_i(t-\tau)$ (memory term) and adds a proportional consensus update $k_2 s_i(t)$ (fresh feedback). In fact, the specific form of the consensus update is not crucial—any minimal controller that guarantees consensus suffices. Our self-learning control plays the role of the *keep-reversing* cue: as long as consensus has not been achieved, it persistently carries forward and amplifies the recent control effort, and only relaxes once the task is effectively done.

Remark 1. *Here, whether $k_2 > 0$ or $k_2 < 0$ depends on the implementation within the controller; it is not mandatory. If a plus sign (+) is used between two elements in (7), then $k_2 < 0$. If a minus sign (−) is employed, indicating that it represents inherently negative feedback, then we say $k_2 > 0$ is required. To avoid ambiguity, we strictly define k_2 as a negative scalar ($k_2 < 0$).*

Define the stacked format of the control variable $u = \mathrm{col}(u_1, \ldots, u_N) \in \mathbb{R}^{Nm}$,

$$u(t) = k_1\, u(t-\tau) + k_2\, s(t). \tag{8}$$

3.2. Closed-Loop Disagreement Dynamics

Stacking (1) across agents and using (6)–(8),

$$\begin{aligned} \dot{x}(t) &= \big(I_N \otimes A\big)x(t) + \big(I_N \otimes B\big)u(t) + \mathbf{1} \otimes d, \\ s(t) &= \big(L \otimes I_n\big)x(t), \\ u(t) &= k_1\, u(t-\tau) + k_2\, s(t). \end{aligned} \tag{9}$$

Differentiating s:

$$\begin{aligned} \dot{s}(t) &= (L \otimes I_n)\dot{x}(t) \\ &= (L \otimes A)x(t) + (L \otimes B)u(t) + (L \otimes I_n)(\mathbf{1} \otimes d). \end{aligned} \tag{10}$$

Since $L\mathbf{1} = 0$, the disturbance injects only through agent channels yet cancels in the agreement subspace; in the disagreement subspace, it manifests via heterogeneity. Specifically, the term $w(t)$ is introduced to represent these heterogeneous components, whose boundedness is directly guaranteed by the limits of individual agent disturbances.

$$\dot{s}(t) = (L \otimes A)x(t) + (L \otimes B)u(t) + w(t) \tag{11}$$

where $\|w(t)\| \le \bar{w}$. Using the orthogonal decomposition [18]

$$x = (L^{\dagger} \otimes I_n)s + \big(\frac{1}{N}\mathbf{1}\mathbf{1}^{\top} \otimes I_n\big)x, \tag{12}$$

the consensus component vanishes in s-dynamics. Standard in consensus analysis, we can bound $\|x\|$ by $\varrho\|s\|$ in the disagreement subspace (via spectral properties of L).

Remark 2. *$L^\dagger$ denotes the pseudoinverse of the graph Laplacian L. Their relationship is as follows: L is symmetric positive semidefinite with $L\mathbf{1} = \mathbf{0}$, hence singular (has a zero eigenvalue). $L^\dagger$ satisfies $LL^\dagger L = L$, $L^\dagger LL^\dagger = L^\dagger$, and $(LL^\dagger)^\top = LL^\dagger$, $(L^\dagger L)^\top = L^\dagger L$. For a connected graph, $LL^\dagger = L^\dagger L = I_N - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$, the orthogonal projector onto the disagreement subspace $\mathbf{1}^\perp$. Consequently, $(L^\dagger \otimes I_n)s$ is the minimum-norm solution of $(L \otimes I_n)x = s$ in the orthogonal complement of the consensus subspace.*

Remark 3. *In the following analysis, I utilize the standard spectral graph theory result $\|x\| \leq \frac{1}{\lambda_2(L)}\|s\|$ to explicitly bound the state deviation x by the disagreement vector s, where $\lambda_2(L)$ denotes the algebraic connectivity of the graph.*

Hence there is an $\alpha_A > 0, \alpha_B > 0$ such that

$$\|(L \otimes A)x\| \leq \alpha_A \|s\|, \tag{13}$$

$$\|(L \otimes B)u\| \leq \alpha_B \|u\|. \tag{14}$$

Substituting (8) into (11), we have

$$\dot{s}(t) = \Phi s(t) + \Psi u(t-\tau) + w(t), \tag{15}$$

where

$$\Phi := (L \otimes A) + k_2(L \otimes B), \tag{16}$$

$$\Psi := k_1(L \otimes B). \tag{17}$$

3.3. Theorem and Stability Analysis

Theorem 1. *Consider (1) over a connected undirected graph, with disturbance bounded by (3). Under the self-learning control scheme (7) with $k_1 \in (0,1)$, $k_2 < 0$, and a small learning interval τ such that learning difference upper bound α_τ is small, if there is a $P = P^\top > 0$, $Q = Q^\top > 0$ satisfying $\Xi \prec 0$ in (28), then the consensus error $s(t)$ is uniformly ultimately bounded and obeys (31), i.e., $s(t)$ enters a small region exponentially.*

Proof. I employ a Lyapunov-type functional similar to self-learning analysis methods to analyze the UUB property

$$V(t) = \frac{1}{2}s^\top(t)P\,s(t) + \int_{t-\tau}^{t} s^\top(\zeta)Q\,s(\zeta)\,d\zeta, \tag{18}$$

where $P = P^\top > 0$, $Q = Q^\top > 0$. Differentiating, we have

$$\begin{aligned}\dot{V}(t) &= s^\top P\dot{s} + s^\top Qs - s^\top(t-\tau)Qs(t-\tau)\\ &= s^\top P\big(\Phi s + \Psi u(t-\tau) + w\big) + s^\top Qs - s_\tau^\top Qs_\tau\\ &= s^\top(P\Phi + Q)s + s^\top P\Psi\, u(t-\tau) + s^\top Pw - s_\tau^\top Qs_\tau,\end{aligned} \tag{19}$$

where $s_\tau := s(t-\tau)$. From (8), $u(t-\tau) = k_1 u(t-2\tau) + k_2 s(t-\tau)$. To avoid infinite recursion, we upper-bound the learning term via the actuator smoothness assumption as in our self-learning work: we define the learning difference

$$\tilde{u}(t) := u(t) - u(t-\tau) \tag{20}$$

and assume $\|\tilde{u}(t)\| \le \alpha_\tau$, with $\alpha_\tau \to 0$ as $\tau \to 0$ (small sampling). Then

$$u(t-\tau) = u(t) - \tilde{u}(t) \tag{21}$$

and using $u(t) = k_1 u(t-\tau) + k_2 s(t)$, we obtain the equivalent algebraic form

$$u(t) = \kappa_1 s(t) - \kappa_2 \tilde{u}(t), \tag{22}$$

with

$$\kappa_1 := \frac{k_2}{1-k_1}, \quad \kappa_2 := \frac{k_1}{1-k_1}. \tag{23}$$

Thus,

$$u(t-\tau) = u(t) - \tilde{u}(t) = \kappa_1 s(t) - (\kappa_2 + 1)\tilde{u}(t). \tag{24}$$

Substitute (24) into (19), and we have

$$\begin{aligned} \dot{V}(t) &= s^\top (P\Phi + Q)s + s^\top P\Psi\big(\kappa_1 s - (\kappa_2+1)\tilde{u}\big) \\ &\quad + s^\top P w - s_\tau^\top Q s_\tau \\ &= s^\top \big(P\Phi + Q + \kappa_1 P\Psi\big)s - s^\top P\Psi(\kappa_2+1)\tilde{u} \\ &\quad + s^\top P w - s_\tau^\top Q s_\tau. \end{aligned} \tag{25}$$

By Young's inequality, for any $\eta_1 > 0, \eta_2 > 0$,

$$\begin{aligned} -s^\top P\Psi(\kappa_2+1)\tilde{u} &\le \eta_1 \|s\|^2 + \frac{(\kappa_2+1)^2}{4\eta_1}\|P\Psi\|^2\|\tilde{u}\|^2, \\ s^\top P w &\le \eta_2 \|s\|^2 + \frac{1}{4\eta_2}\|P\|^2 \bar{w}^2. \end{aligned} \tag{26}$$

Using $\|\tilde{u}\| \le \alpha_\tau$ and $-s_\tau^\top Q s_\tau \le 0$, we obtain

$$\begin{aligned} \dot{V}(t) &\le s^\top \big(P\Phi + Q + \kappa_1 P\Psi\big)s + (\eta_1+\eta_2)\|s\|^2 \\ &\quad + \frac{(\kappa_2+1)^2}{4\eta_1}\|P\Psi\|^2\alpha_\tau^2 + \frac{1}{4\eta_2}\|P\|^2\bar{w}^2. \end{aligned} \tag{27}$$

Choose P, Q such that the symmetric part

$$\Xi := \frac{1}{2}\big(P\Phi + \Phi^\top P\big) + Q + \frac{\kappa_1}{2}\big(P\Psi + \Psi^\top P\big) \tag{28}$$

is negative definite. Then there is a $\lambda > 0$ such that $s^\top \big(P\Phi + Q + \kappa_1 P\Psi\big)s \le -\lambda\|s\|^2$. With this,

$$\dot{V}(t) \le -(\lambda - \eta_1 - \eta_2)\|s\|^2 + \delta, \tag{29}$$

where

$$\delta := \frac{(\kappa_2+1)^2}{4\eta_1}\|P\Psi\|^2\alpha_\tau^2 + \frac{1}{4\eta_2}\|P\|^2\bar{w}^2. \tag{30}$$

Select η_1, η_2 small enough so that $\lambda - \eta_1 - \eta_2 =: \lambda_\star > 0$. Since $V(t) \ge \frac{1}{2}\lambda_{\min}(P)\|s\|^2$, it follows from (29) that $s(t)$ is uniformly ultimately bounded, and

$$\limsup_{t\to\infty}\|s(t)\| \le \sqrt{\frac{2\,\delta}{\lambda_\star\,\lambda_{\min}(P)}}. \tag{31}$$

This result explicitly establishes the mapping to the standard class-$\mathcal{KL}$ definition of UUB, confirming that the system state converges exponentially to a residual set characterized by the ultimate bound radius $b = \sqrt{2\delta/(\lambda_\star \lambda_{\min}(P))}$. This completes the proof. □

Remark 4. *Sketch of gain feasibility. Since* Φ *and* Ψ *scale with* k_2 *and* k_1*, respectively, per (17), the LMI* $\Xi \prec 0$ *can be satisfied by (i) choosing* k_2 *large enough to dominate* $L \otimes A$ *in the disagreement subspace, and (ii) choosing* $k_1 \in (0,1)$ *such that* κ_1, κ_2 *are finite. A small* τ *yields a small* α_τ*, tightening the ultimate bound.*

3.4. Tuning Guidelines and Practical Notes

- Sequential Tuning Workflow: fix τ according to hardware limits; select k_2 for baseline stability; adjust k_1 to accelerate convergence.
- Learning intensity k_1: A larger k_1 increases prior information usage and accelerates early transients but also increases κ_2 and sensitivity to α_τ. Values in $[0.7, 0.95)$ are effective; near 1 may risk peaking if actuators saturate.
- Consensus gain k_2: Boosts damping in the disagreement subspace. Pick k_2 to satisfy $\Xi \prec 0$ with a margin; a larger k_2 yields a faster decay but may excite saturation.
- Learning interval τ: choose as one or a few sampling periods so that the learning difference α_τ is small; this directly reduces δ and the ultimate bound (31).
- Saturation: As in our study, learning can cause initial peaks. Reducing k_1 at large $\|s\|$ (variable learning intensity) can weaken saturation; a static conservative choice of k_1 already achieves improved speed with low complexity.

4. Simulation

This section validates the proposed self-learning consensus controller with the finalized implementation. I emphasize: (i) a fully specified undirected connected topology that can be modified by edge pairs, (ii) negative-feedback consensus with self-learning intensity, and (iii) open sources.

4.1. Communication Graph

We consider $N = 6$ agents with an undirected connected topology specified directly by edge pairs $\mathcal{E} = \{(0,1), (1,2), (2,3), (3,4), (4,5), (5,0), (0,3)\}$ where nodes are zero-indexed in the simulation. See Figure 1. The Laplacian matrix $L \in \mathbb{R}^{6\times 6}$ is computed from the adjacency A by $L = D - A$, $D = \mathrm{diag}(A\mathbf{1})$. This topology can be changed at will by editing the set $\mathcal{E}$; L can be rebuilt for any N and any undirected edge list.

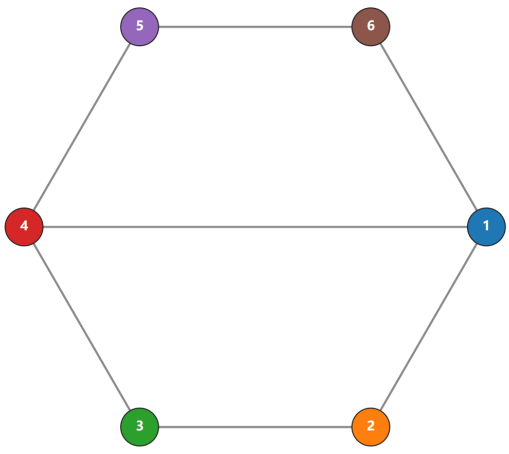

Figure 1. The graph of 6 agents with edge pairs.

Agent Dynamics and Disturbance: Each agent follows $A = \begin{bmatrix} 0 & 1 \\ -1 & -0.4 \end{bmatrix}, B = I_2$, $C = I_2$. Agent-wise disturbances are $d_i(t) = \begin{bmatrix} 0.5\sin(0.6t + \varphi_i) \\ 0.5\cos(0.5t + \psi_i) \end{bmatrix}$, where phases (φ_i, ψ_i) are evenly distributed in $[0, \pi)$ and $[0, \pi/2)$, respectively.

4.2. Controller and Implementation Details

4.2.1. Self-Learning Control Law

$u_i(t) = k_1\, u_i(t-\tau) + k_2\, s_i(t)$, $\quad s_i(t) = \sum_{j=1}^{6} a_{ij}(x_i - x_j)$, with parameters:

$$k_1 = 0.90, \quad k_2 = -2.5, \quad \tau = 0.01 \text{ s}.$$

The differential equations are integrated by RK4 with step $\Delta t = 0.001$ s and horizon $T = 30$ s. The learning delay is implemented by an integer buffer of $N_\tau = \tau/\Delta t = 10$ steps. Optional actuator constraints are enabled: $\|u_i(t)\|_\infty \le 2, \|\dot{u}_i(t)\|_\infty \le 50$ (per second), applied componentwise. Initial states are sampled uniformly within $\|x_i(0)\| \le 0.8$.

4.2.2. Baselines and Metrics

A non-learning baseline is $u_i(t) = k_2\, s_i(t)$. I report the disagreement norm $\|s(t)\|_2$ and other matrices, the cumulative control effort $J_u = \int_0^T \sum_{i=1}^{6} \|u_i(t)\|_2\, dt$, and representative state trajectories.

4.3. Results

This section evaluates the proposed self-learning consensus control strategy against a proportional baseline (BL) on a network of $N = 6$ second-order agents under bounded actuation and exogenous disturbances. Unless otherwise stated, both methods use the same initial conditions, network topology, and simulation parameters. I report four complementary views: agent trajectories, disagreement norm, raw control inputs, and cumulative control effort.

4.3.1. State Trajectories and Cohesion

Figure 2 compares the first state component $x_{i,1}(t)$ of all agents for self-learning control (top) and BL (bottom). For clarity, each panel includes an inset that zooms into the last 20% of the horizon. The self-learning control trajectories remain tightly clustered throughout the transient and converge nearly synchronously, while the BL trajectories exhibit noticeable dispersion near the end of the horizon. The inset highlights that self-learning control maintains a smaller inter-agent spread during the final convergence phase, indicating stronger cohesion.

https://doi.org/10.3390/appliedmath6030037

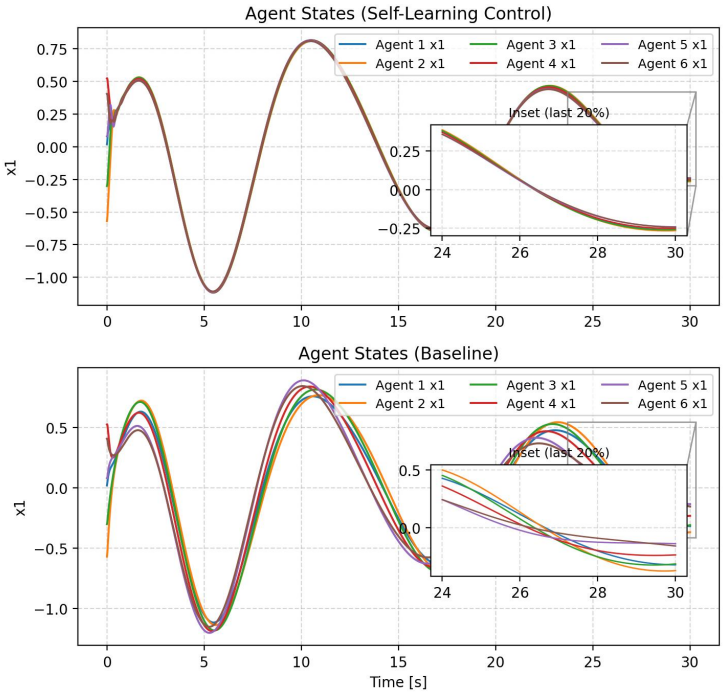

Figure 2. Agent state trajectories ($x_{i,1}(t)$) for Self-Learning (**top**) and Baseline (**bottom**). Insets zoom into the last 20% of the horizon, highlighting the tighter clustering achieved by the proposed method.

4.3.2. Disagreement Norm

Figure 3 depicts the time evolution of the disagreement norm $\|s(t)\|$, where $s(t) = LX(t)$ with L the Laplacian. The self-learning control curve decays rapidly and remains close to zero for the remainder of the horizon, whereas the BL curve shows a slower decay with residual oscillations. This confirms that SL accelerates consensus formation and effectively suppresses disturbance-induced disagreements.

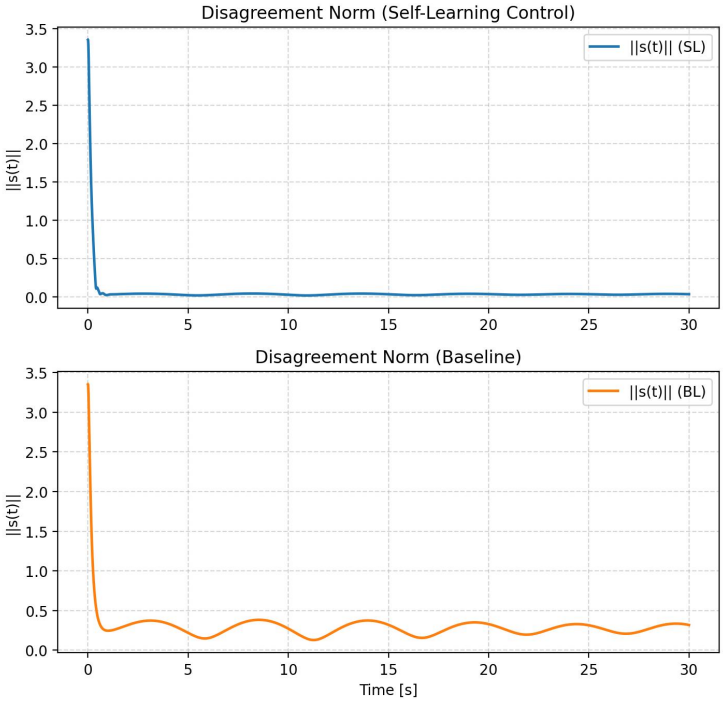

Figure 3. Disagreement norm $\|s(t)\|$ for Self-Learning (**top**) and Baseline (**bottom**). The proposed controller exhibits a faster decay and lower steady-level disagreement under disturbances.

4.3.3. Control Inputs (Raw)

Figure 4 presents the raw control inputs for all agents and both input channels: $u_{i,1}(t)$ (solid) and $u_{i,2}(t)$ (dashed) with consistent colors per agent. Despite its faster convergence, SL does not incur persistent high-amplitude actuation; after a short transient, all channels settle to small magnitudes comparable to or lower than BL. This indicates that the proposed memory-based feedback primarily reshapes the early transient rather than sustaining aggressive control.

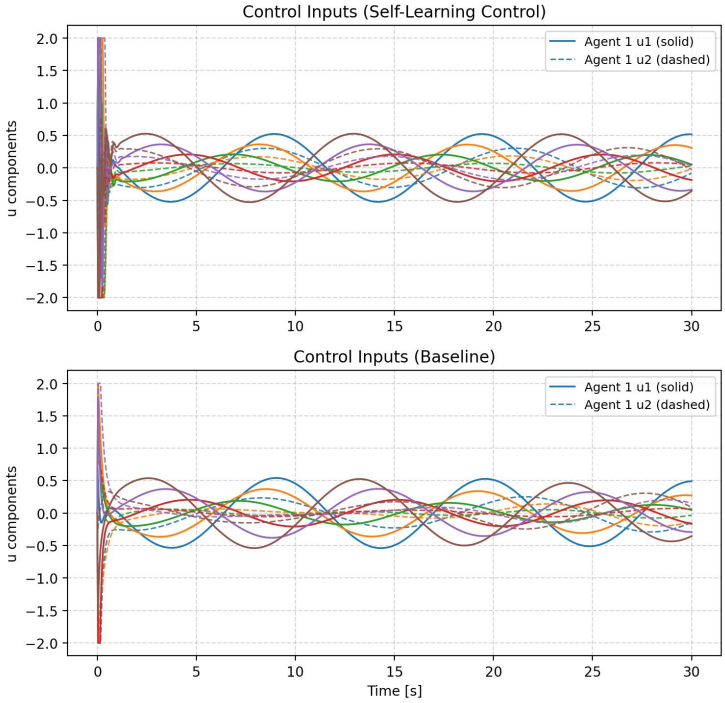

Figure 4. Raw control inputs for all agents and both channels: $u_{i,1}(t)$ (solid) and $u_{i,2}(t)$ (dashed). The Self-Learning control concentrates effort in the early transient without sustained high-amplitude actuation.

Remark 5. *Since this study primarily focuses on undirected graph scenarios, directed topologies, switching topologies, and similar scenarios are not covered. I should note that, in the Data Availability Statement of the manuscript, I have made the source code open on http://doi.org/10.13140/RG.2.2.31052.48003, allowing readers to modify the code to test the specific scenario needed.*

4.3.4. Summary

The steady-phase statistics in Table 1 (computed for $t \geq 5$ s) show that the proposed self-learning control maintains substantially tighter consensus than the proportional baseline (BL). Specifically, self-learning control reduces the mean disagreement by about 89% (0.0297 vs. 0.2703) and the maximum steady-phase disagreement by an order of magnitude (0.0403 vs. 0.3831), while its variance is two orders of magnitude smaller (4.58×10^{-5} vs. 4.81×10^{-3}), indicating markedly improved coherence and disturbance rejection in the stable regime. This enhanced accuracy is achieved with only a modest increase in cumulative control effort (J_u), reflecting an efficient trade-off between precision and effort during steady operation.

Table 1. Steady-phase performance (custom_pairs, $N = 6$, $T = 30$ s; steady phase defined as $t \geq 5$ s).

Method	J_u	$\|s\|_{\text{final}}$	Average$\|s\|$	max $\|s\|$	Var$[\|s\|]$
self-learning control	24.31	0.0316	0.0297	0.0403	4.58×10^{-5}
BL	21.79	0.318	0.2703	0.3831	4.81×10^{-3}

5. Conclusions

I have proposed here a consensus control strategy for multiagent systems based on a self-learning control that reuses prior control inputs. With a Lyapunov-type analysis inspired by our self-learning control framework, I established UUB consensus under bounded disturbances and derived clear tuning insights. The approach is algebraically simple, computationally light, and effective for accelerating transients while preserving robustness. Future research will focus on nonlinear systems, time-varying learning intensity, and dynamic networks. I will also explore low-bit-rate communication and directed graph topologies, comparing the proposed approach with existing methods. Additionally, the framework will be extended to output-feedback scenarios using learning observers.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/appliedmath6030037/s1, Supplementary Materia File. Figure S1: Cumulative Control Effort (Self-Learning Control) and Cumulative Control Effort (Baseline); Figure S2: Disagreement Norm (Self-Learning Control) and Disagreement Norm (Baseline); Figure S3: Agent States (Self-Learning Control) and Agent States (Baseline); Figure S4: Control Inputs (Self-Learning Control) and Control Inputs (Baseline); Figure S5: The graph of 6 agents with edge pairs; Table S1: Summary Metrics; HTML: Network Topology (N = 6, topology = "custom_pairs").

Funding: Supported by the National Natural Science Foundation of China (No. 62573211).

Data Availability Statement: The original contributions presented in this study are included in the Supplementary Materials. Further inquiries can be directed to the corresponding author.

Acknowledgments: Thank you to the editors and anonymous reviewers.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Zhu, W.; Pu, H.; Wang, D.; Li, H. Event-based consensus of second-order multi-agent systems with discrete time. *Automatica* **2017**, *79*, 78–83. [CrossRef]
2. Jiang, Y.; Fan, J.L.; Gao, W.N.; Chai, T.Y.; Lewis, F.L. Cooperative Adaptive Optimal Output Regulation of Discrete-Time Nonlinear Multi-Agent Systems. *Automatica* **2020**, *121*, 109149. [CrossRef]
3. Yi, X.; Liu, K.; Dimarogonas, D.V.; Johansson, K.H. Dynamic event-triggered and self-triggered control for multi-agent systems. *IEEE Trans. Autom. Control* **2018**, *64*, 3300–3307. [CrossRef]
4. Doostmohammadian, M. Single-bit consensus with finite-time convergence: Theory and applications. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 3332–3338. [CrossRef]
5. Zhang, C.; Ahn, C.K.; Wu, J.; He, W. Online-learning control with weakened saturation response to attitude tracking: A variable learning intensity approach. *Aerosp. Sci. Technol.* **2021**, *117*, 106981. [CrossRef]
6. Dai, M.Z.; Xiao, F.; Wei, B. Event-triggered and quantized self-triggered control for multi-agent systems based on relative state measurements. *J. Frankl. Inst.* **2019**, *356*, 3711–3732. [CrossRef]
7. Ruan, X.; Feng, J.; Xu, C.; Wang, J. Observer-based dynamic event-triggered strategies for leader-following consensus of multi-agent systems with disturbances. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 3148–3158. [CrossRef]
8. Amirkhani, A.; Barshooi, A.H. Consensus in multi-agent systems: A review. *Artif. Intell. Rev.* **2022**, *55*, 3897–3935. [CrossRef]
9. Chen, F.; Ren, W. On the control of multi-agent systems: A survey. *Found. Trends Syst. Control* **2019**, *6*, 339–499. [CrossRef]
10. Ni, J.; Zhao, Y.; Cao, J.; Li, W. Fixed-time practical consensus tracking of multi-agent systems with communication delay. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1319–1334. [CrossRef]
11. Ji, Z.; Wang, Z.; Lin, H.; Wang, Z. Controllability of multi-agent systems with time-delay in state and switching topology. *Int. J. Control* **2010**, *83*, 371–386. [CrossRef]

12. Yu, X.; Yang, F.; Zou, C.; Ou, L. Stabilization parametric region of distributed PID controllers for general first-order multi-agent systems with time delay. *IEEE/CAA J. Autom. Sin.* **2019**, *7*, 1555–1564. [CrossRef]
13. Zhang, C.; Xiao, B.; Wu, J.; Li, B. On low-complexity control design to spacecraft attitude stabilization: An online-learning approach. *Aerosp. Sci. Technol.* **2021**, *110*, 106441. [CrossRef]
14. Zhang, C. Self-Learning Control under Practical Actuation. *Aerosp. Eng. Commun.* **2026**, *1*, 36–46. https://www.icck.org/article/abs/aec.2025.320719 (accessed on 6 February 2026).
15. Tan, L.; Jin, G.; Zhou, S.; Wang, L. A Model-Free Online Learning Control for Attitude Tracking of Quadrotors. *Appl. Sci.* **2024**, *14*, 980. [CrossRef]
16. Li, C.; Wang, Y.; Ahn, C.K.; Zhang, C.; Wang, B. Milli-Hertz Frequency Tuning Architecture Toward High Repeatable Micromachined Axi-Symmetry Gyroscopes. *IEEE Trans. Ind. Electron.* **2023**, *70*, 6425–6434. [CrossRef]
17. Zhang, C.; Wu, J.; Ahn, C.K.; Fei, Z.; Wei, C. Learning Observer and Performance Tuning-Based Robust Consensus Policy for Multiagent Systems. *IEEE Syst. J.* **2022**, *16*, 431–439. [CrossRef]
18. Mesbahi, M.; Egerstedt, M. *Graph Theoretic Methods in Multiagent Networks*; Princeton University Press: Princeton, NJ, USA , 2010.