AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 10(8): 19554–19585.
DOI: 10.3934/math.2025873
Received: 19 June 2025
Revised: 11 August 2025
Accepted: 14 August 2025
Published: 26 August 2025

Research article

Fault-tolerant coordination of robotic car teams via adaptive neural control and real-time fault isolation

Muflih Alhazmi[1], Waqar Ul Hassan[2], Mohammed M. A. Almazah[3], Saadia Rehman[4], Azmat Ullah Khan Niazi[4,*], Somayya Komal[2] and Nafisa A. Albasheir[5]

[1] Department of Mathematics, College of Sciences, Northern Border University, Arar, Saudi Arabia; Muflih.Alhazmi@nbu.edu.sa

[2] Department of Mathematics, Faculty of Sciences, University of Mianwali, 42200, Mianwali, Punjab, Pakistan; waqarulhassan439@gmail.com; somayya.komal@umw.edu.pk

[3] Department of Mathematics, College of Sciences and Arts (Muhyil), King Khalid University, Muhyil 61421, Saudi Arabia; mmalmazah@kku.edu.sa

[4] Department of Mathematics and Statistics, The University of Lahore, Sargodha 40100, Pakistan; saadiarehman468@gmail.com

[5] Department of Mathematics, College of Sciences and Arts (Magardah), King Khalid University, Magardah, 61421, Saudi Arabia; nalbasheir@kku.edu.sa

* **Correspondence:** Email: azmatullah.khan@math.uol.edu.pk.

Abstract: This paper investigates robust cooperative control strategies for multi-robotic car systems operating under sensor and actuator faults. In autonomous driving environments, the degradation or failure of sensors and actuators significantly affects the performance of the system, posing risks to formation control, velocity tracking, and safety. To address these challenges, we propose a robust neural control framework that integrates a dynamic adjustment neural network (DANN) with fault-tolerant design. This architecture enables each robotic car to adaptively learn the system dynamics and adjust control signals in real time, even in the presence of component faults. A fault detection and isolation (FDI) mechanism is incorporated to identify malfunctioning elements, allowing the control system to dynamically compensate and maintain coordinated behavior. Lyapunov-based analysis is employed to guarantee stability and convergence of the system. In addition to theoretical development, a detailed simulation example involving a team of robotic cars under various sensor and actuator fault scenarios is presented to demonstrate the effectiveness and robustness of the proposed control strategy. The results confirm reliable tracking performance, strong resilience, and improved formation stability under realistic fault conditions.

Keywords: robotic car systems; fault-tolerant control; dynamic adjustment neural network (DANN); sensor and actuator faults; cooperative control; fault detection and isolation (FDI); Lyapunov stability;

adaptive neural control; formation control; autonomous driving systems
Mathematics Subject Classification: 34H05, 93C10, 93D09

1. Introduction

Robots are becoming increasingly essential in industrial settings, offering high precision, adaptability, and reliability in complex tasks. To manage nonlinear uncertainties, neural observer-based output-feedback controllers have been proposed to enhance system stability [1]. Data-driven deep learning techniques are enhancing fault diagnosis in industrial components such as bearings [2], while quantized control of a heavy-lift launch vehicle under actuator faults and rate gyro malfunctions is achieved using a predefined-time observer to reconstruct the derivative of attitude tracking errors, enabling a quantized controller to drive errors to a small neighborhood of the origin within a precisely predefined time [3]. Gear transmissions are crucial in rotating machinery, and their faults can compromise the reliability of the equipment. To address sensor limitations, a tooth-backlash-inspired triboelectricity nanogenerator by using comb-shaped electrodes is proposed for self-powered gear condition monitoring [4]. Chatter detection in robotic milling [5] further demonstrates practical advancements in intelligent robotic systems. Robotic systems require high-resolution signal analysis and interference-resistant control for reliable performance in dynamic environments. The complex structure and nonstationary operating patterns of machinery result in multi-component signals with time-varying spectral characteristics, challenging traditional time–frequency analysis methods. To address this, adaptive synchronous demodulation transform (ASDT) has been proposed, which designs demodulation terms based on the spectral structure to render each signal component stationary without prior knowledge [6]. In radar-aided millimeter-wave (mmWave) Vehicle-to-Infrastructure (V2I) communications, a key challenge is mitigating mutual interference between vehicular radar and communication systems operating in the same frequency bands. Joint design of Multiple-input multiple-output (MIMO) transmit waveforms and receive filter banks for Road Side Unit (RSU)-mounted radar has been explored to handle such interference in spectrally crowded environments [7]. Stabilization techniques for partial differential equation–ordinary differential equation (PDE–ODE) systems with input saturation enhance the reliability of distributed robotic control, while the Scaling-basis chirplet transform (SBCT) provides sharper time–frequency representations for noisy, multi-component sensor data [8,9]. Moreover, robust preview control strategies such as Padé-approximation-based designs with Equivalent Input Disturbance (EID) compensation further strengthen the closed-loop stability of robotic systems under external disturbances [10]. Fault diagnosis plays a vital role in the reliable operation of Permanent Magnet Synchronous Machine (PMSM) drives, where lightweight multisource data fusion methods have been introduced to overcome the limitations of single-signal deep learning models [11, 12]. Recent studies have also applied machine learning to assess fatigue resilience under ice-induced vibrations in offshore wind turbines, highlighting its adaptability in harsh environmental conditions [13]. Furthermore, fault-tolerant converter topologies with adaptive hardware reconfiguration ensure power continuity with minimal redundancy, while discrete element method (DEM) based simulations provide insight into ballast track behavior under fastener failure conditions [14, 15]. Granular material classification, critical scenario generation

for intelligent vehicles, and high-fidelity ultrasonic radar testing have all contributed to advancing robotic perception and autonomous system evaluation [16–18]. Meanwhile, force feedback bilateral teleoperation and adaptive aero-engine anomaly monitoring represent key breakthroughs in remote manipulation and aviation safety under complex, high-risk environments [19, 20]. For precise vehicle trajectory planning, a twisted Gaussian risk model accounts for both lateral and longitudinal states of target vehicles to better assess collision risk [21]. To aid planetary rovers in escaping soft terrain entrapment, an optimized "sweeping-spinning" gait using Bayesian algorithms improves traversal capability [22]. A hybrid degradation model combining data and physics addresses non-linear fault evolution for better early warnings in mechanical systems [23]. Visual navigation benefits from causal intervention visual navigation (CIVN), which mitigates environmental confounders through causal front-door adjustments [24], while occlusion-aware binocular vision enhances multi-target robotic grasping accuracy [25]. Ensuring reliability in multi-state phased-mission systems (MS-PMSs) is crucial, and a new importance measure is introduced using multi-state multi-valued decision diagram (MSMDD) and Markov models to capture component degradation and phase transitions [26]. Addressing real-world engineering challenges, microwave-based deicing for railway wires faces alignment issues due to structural constraints under cold climates [27]. To improve nonlinear Multiple-input multiple-output (MIMO) systems with uncertain delays, an adaptive PI event-triggered control strategy is developed using a single radial basis function neural network (RBFNN) based parameter update [28]. Lane-keeping for autonomous heavy trucks under complex conditions is achieved via a novel memory-based event-triggered scheme adjusting control frequency dynamically [29]. Lastly, soft robots benefit from a fuzzy logic system (FLS) and barrier Lyapunov function (BLF) based adaptive control for hysteretic nonlinear systems, ensuring state constraints are strictly maintained [30]. To enhance autonomous driving, joint Light detection and ranging (LiDAR) based scene flow estimation and moving object segmentation are proposed, leveraging their geometric correlation to mutually boost accuracy [31]. In industrial automation, a hybrid deep learning model combining variational modal decomposition and particle swarm optimization improves magnetic encoders accuracy for robotic arms [32]. Vehicular crowdsensing security is strengthened via a conditional anonymization and leakage-resilient authentication scheme to address traceability and efficiency issues [33]. For connected and automated vehicles, a bi-level curvilinear coordination strategy enables safe and efficient platoon formation across lanes [34]. With the advancement of transportation systems, fast data dissemination has become a critical network requirement. Vehicle Ad-Hoc Networks (VANETs) leverage their high mobility and dynamic environment effectively using Ad-hoc On-demand Distance Vector (AODV) and Greedy Perimeter Stateless Routing (GPSR) routing protocols, which are widely applied in vehicular networks [35]. With the rapid development of driver-less technology, challenges such as varying vehicle densities, dynamic wireless channel conditions, and fluctuating bandwidth significantly increase offloading delays and energy consumption in conventional algorithms. To address this, the vehicle task offloading problem has been modeled as a Markov decision process, and a deep reinforcement learning-based task offloading algorithm using double deep Q-network (DDQN) has been proposed, incorporating a task-feature-based offloading mechanism to guide agents toward optimal decision-making strategies [36]. In the realm of driving safety, recognizing unfavorable driving states by using electroencephalography signals and functional connectivity has proven effective, though conventional connectivity methods face issues like spurious synchronization. To overcome this, a novel functional connectivity matrix construction method based on amplitude

envelope correlation with leakage correction, combined with an ensemble learning algorithm, has been developed to accurately identify unfavorable driving state across multiple electroencephalography frequency bands [37]. In complex systems, machine intelligence fault prediction (MIFP) plays a vital role in ensuring safe and reliable operation. To enhance interpretability and frequency-level understanding, a wavelet-guided network (WaveGNet) has been proposed. This physically interpretable network integrates discrete wavelet transform (DWT) principles and convolutional neural networks, enabling deep frequency decomposition and multi-resolution learning via a novel learnable frequency layer (FL-Layer) [38]. In robotic control, a prescribed performance adaptive robust control (PPARC) method has been introduced to ensure that trajectory tracking errors of uncertain robotic manipulators remain within predefined constraints. Through state transformation and Lyapunov-based stability analysis, the scheme guarantees robust performance under uncertainties and varying initial conditions [39]. Furthermore, wheeled mobile robots (WMRs) with variable wheelbases are gaining traction for their ability to navigate deformable terrains. Inspired by a worm's adaptive body stretching under environmental resistance, a biologically inspired control strategy has been proposed to manage the added control complexity introduced by variable-length configurations [40]. In multi-string photovoltaic power generation systems, capacitor degradation in DC-link circuits may lead to power outages; to mitigate this, a fault prognosis method utilizing busbar voltage and per-string cable current was proposed, overcoming uncertainties via an optimized 3D plane based on series resistance [41]. In the field of logistics and visual servoing, a neurodynamics-based visual servo predictive control (NVSPC) approach has been introduced for omnidirectional mobile robots (OMRs), integrating neurodynamic models into nonlinear error dynamics to ensure smooth trajectory tracking under physical and visual constraints [42]. Addressing parameter uncertainties and external disturbances in electro-hydrostatic actuators (EHAs), an actor-critic (AC) reinforcement learning strategy was developed using RBFNNs and a Lagrangian-based performance index for adaptive control [43]. In robotic machining, where low and configuration-dependent stiffness complicates surface roughness prediction, a mutual cross-attention fusion network (MCFN) has been proposed to fuse internal and external signals for robust prediction based on experimental datasets [44]. Furthermore, in high-speed maglev train (HSMT) systems, a novel fuzzy-based fault-tolerant controller was developed to cope with actuator failure, saturation, and dead zones by employing an adaptive fuzzy update law with auxiliary manifolds for real-time compensation [45]. For trajectory tracking in nonlinear two-link mechanical systems, the Udwadia–Kalaba formulation has been employed to construct constraint-driven dynamic equations, enabling precise tracking under uncertain and time-varying environments [46, 47]. Recent studies have focused on cooperative localization for multi-autonomous underwater vehicles (MAUVs), particularly under sensor failures. Notably, a filtering algorithm based on a 'two-master-one-slave' model has been proposed to handle compass failures, unknown course angles, and non-Gaussian measurement noise from outliers, ensuring robust localization performance [48].

Motivation: Autonomous multi-robot systems are increasingly deployed in dynamic environments where sensor degradation and actuator faults pose serious threats to stability, safety, and coordination. Existing control methods often fail to handle unknown nonlinearities and unpredictable faults in real time, motivating the need for a more adaptive and robust control approach.

Contributions: The main technical contributions of this work are as follows:

- Development of a novel adaptive neural control framework that addresses simultaneous unknown nonlinearities and actuator and sensor faults;

- Integration of a real-time fault detection and isolation (FDI) mechanism directly within the control architecture;
- Establishment of Lyapunov-based theoretical guarantees to ensure the uniform ultimate boundedness of the tracking errors;
- Demonstration of practical applicability in real-time multi-robot coordination without the need for precise fault models;
- Validation through detailed simulations showing resilient formation control and reliable tracking performance under multiple fault scenarios.

The structure of the rest of the paper is as follows:

(1) Section 1 provides literature review on robotic control systems.

(2) Section 2 presents the essential mathematical tools and notations required for system analysis, including graph-theoretic concepts.

(3) Section 3 defines the control objective and sets up the problem formulation under sensor and actuator faults in multi-robotic vehicle systems.

(4) Section 4 provides a detailed state-space representation of the system with nonlinear dynamics reflecting real-world behaviors.

(5) Section 5 introduces an adaptive control protocol based on dynamic adaptive neural networks (DANN) to address uncertainty and faults.

(6) Section 6 develops a real-time neural network controller and establishes stability using Lyapunov-based theoretical analysis.

(7) Section 8 offers numerical simulations to validate the robustness and effectiveness of the proposed method.

(8) Section 9 provides the conclusion of this paper and future research directions.

2. Preliminaries

2.1. Graph theory

Consider a weighted digraph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$ and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ are the sets of nodes and edges, respectively, with the index set $N = \{1, 2, 3, \ldots, N\}$. The adjacency matrix $A = [a_{pq}] \in \mathbb{R}^{N\times N}$ is defined as

$$\begin{cases} a_{pq} \neq 0 & \text{if } (n_p, n_q) \in \mathcal{E}, \\ a_{pq} = 0 & \text{otherwise.} \end{cases}$$

If $e_{pq} = (n_p, n_q) \in \mathcal{E}$, then we say that n_p points to n_q, and $a_{pq} > 0$ indicates that there is a cooperative relationship between the nodes n_p and n_q. Assume that $a_{pp} = 0$ for all $p \in N$. The set $N_p = \{q \in \mathcal{N} : e_{pq} \in \mathcal{E}\}$ represents the neighbors of node n_p. If $\overline{D}$ is a diagonal matrix such that $\overline{d}_p = \sum_{q\neq p} a_{pq}$, then the Laplacian matrix L of the digraph $\mathcal{G}$ is $\overline{D} - A$ and $L_N = \mathbf{0}_N$, where $\mathbf{1}_N$ and $\mathbf{0}_N$ are column matrices

of size $N \times 1$ with all entries 1 and 0, respectively.

$$L = \begin{bmatrix} 0 & 0_N \\ L_2 & L_1 \end{bmatrix},$$

where the matrices $L_1 \in \mathbb{R}^{N\times N}$ and $L_2 \in \mathbb{R}^{N\times 1}$, and L_1 is a nonsingular M-matrix.

2.2. *Robotic model*

We assume that, in a network of N robotic cars, there is one robot that acts as the leader while the remaining N robots are designated as followers. Both the leader and follower robots are equipped with sensors that monitor their respective states, including their position, velocity, and acceleration. The leader robot updates its state through sensor feedback to improve trajectory tracking and enhance the overall stability of the robotic car control system. Furthermore, a neural network-based controller is used to regulate the behavior of the robotic cars. The Neural network controller processes the data received from the sensors and computes the required control inputs. In the leader robotic car, the controller refines its motion to ensure the smooth and stable trajectory tracking of the system. Similarly, in the follower robotic cars, the controller computes the control actions that guide each robot in maintaining coordinated movement with the leader. The control input for each follower robot is adjusted through its actuator, ensuring that the desired state is achieved even in the presence of disturbances or any attacks. The time, and the neural network controller effectively compensates for uncertainties, sensor faults, and actuator attacks. This coordinated control framework ensures stable and precise movement for both the leader and follower robots, improving the system's performance and its resilience against external disturbances, as shown in Figure 1.

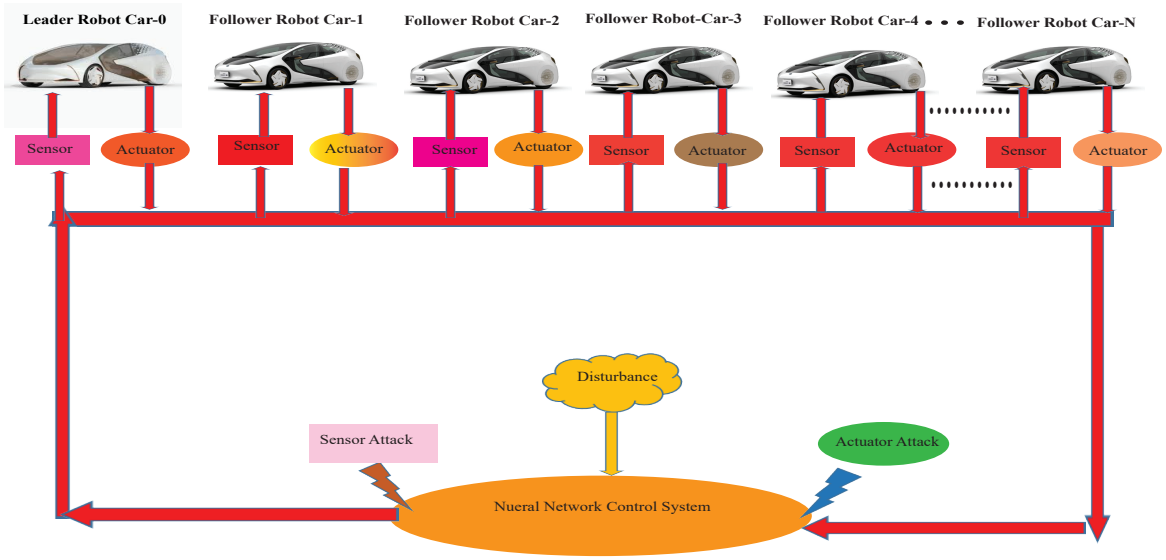

Figure 1. Control structure of robotic cars.

Lemma 2.1. *Let $V(t)$ be a positive and continuous function defined for all $t \geq 0$. Assume that it satisfies the following differential inequality:*

$$\dot{V}(t) \leq -\alpha V(t) + \beta,$$

where $\alpha > 0$ and $\beta > 0$ are constants. Then the solution satisfies the following upper bound:

$$V(t) \leq \left(V(0) - \frac{\beta}{\alpha}\right) e^{-\alpha t} + \frac{\beta}{\alpha}. \tag{2.1}$$

Lemma 2.2. *Given any real numbers x and y, and any positive scalar ε, the following holds:*

$$xy \leq \frac{\varepsilon}{2} x^2 + \frac{1}{2\varepsilon} y^2. \tag{2.2}$$

Additionally, if $xy \geq 0$, then the following inequality is also valid:

$$x^2 + y^2 \leq (x + y)^2. \tag{2.3}$$

Lemma 2.3. *For the vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ with the components a_i and b_i, the inequality below is satisfied:*

$$\left(\sum_{i=1}^{n} a_i b_i\right)^2 \leq \left(\sum_{i=1}^{n} a_i^2\right)\left(\sum_{i=1}^{n} b_i^2\right). \tag{2.4}$$

3. Problem formation

We assume that there is one leader robot and the remaining robots are followers, moving in the horizontal plane. For the graphical formation, we achieve consensus and design for a virtual-type leader robot that adjusts in such a way that each follower robot is set as follows.

The position $p_0(\tilde{k})$, velocity $v_0(\tilde{k})$, and acceleration $a_0(\tilde{k})$ of the leader robot are given as,

$$p_0(\tilde{t}) = \begin{bmatrix} p_{0,1}(\tilde{t}) \\ p_{0,2}(\tilde{t}) \\ p_{0,3}(\tilde{t}) \end{bmatrix}^T, v_0(\tilde{t}) = \begin{bmatrix} v_{0,1}(\tilde{t}) \\ v_{0,2}(\tilde{t}) \\ v_{0,3}(\tilde{t}) \end{bmatrix}^T \text{ and } a_0(\tilde{t}) = \begin{bmatrix} a_{0,1}(\tilde{t}) \\ a_{0,2}(\tilde{t}) \\ a_{0,3}(\tilde{t}), \end{bmatrix}^T. \tag{3.1}$$

All these three vectors are in 3D space.

Now, we assign the virtual leader robot state to each follower robot to ensure that every follower adheres to the virtual leader dynamics,

$$\begin{cases} p_0^i(\tilde{t}) = p_0(\tilde{t}) + q_p^i(\tilde{t}), \\ v_0^i(\tilde{t}) = v_0(\tilde{t}) + r_v^i(\tilde{t}), \\ a_0^i(\tilde{y}) = a_0(t), \end{cases} \tag{3.2}$$

where $q_0^i(\tilde{t})$ shows the offset of the follower of each robots relative to the leader robots in 3D space.

4. State-space modeling with nonlinear dynamics

Due to the structure of the aforementioned robot model, including its attitude, wind resistance, system signal attacks, and other physical factors, it consists of a large number of unknown variables, such that,

$$\begin{cases} \dot{p}_i(\tilde{t}) = v_i(\tilde{t}), \\ \dot{v}_i(\tilde{t}) = a_i(\tilde{t}), \\ \dot{a}_i(\tilde{t}) = h_i((p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t})) + g_i((p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t}))(\omega_i(\tilde{t}) + \eta_i(\tilde{t}) + \Delta\omega(\tilde{t})), \end{cases} \tag{4.1}$$

where $\omega_i(\tilde{t})$ is the control input, h_i and g_i are nonlinear functions, and η_i is the signal attack, which combines the sensor attack and the actuator attack. Similarly, the dynamics of the leader robot which is described as,

$$\begin{cases} \dot{p}_0(\tilde{t}) = v_0(\tilde{t}), \\ \dot{v}_0(\tilde{t}) = a_0(\tilde{t}), \\ \dot{a}_0(\tilde{t}) = h_0(p_0(\tilde{t}), v_0(\tilde{t}), a_0(\tilde{t})), \end{cases} \tag{4.2}$$

where $p_0(\tilde{t})$, $v_0(\tilde{t})$, and $a_0(\tilde{t})$ represent the position, velocity, and acceleration states of the leader robot.

Remark 4.1. *In multi-robot vehicle systems, sensor and actuator faults can significantly degrade the performance of coordination tasks. Sensor faults (e.g., bias, drift, or failure) distort the perceived state, leading to inaccurate inter-agent distance estimation or velocity tracking. Actuator faults, such as stuck actuators or loss of actuation force, impair the robot's ability to follow control commands, potentially leading to trajectory deviations, collisions, or formation collapse. Existing coordination methods often rely on fixed control laws and assume ideal sensing/actuation, which limits their applicability in uncertain, fault-prone environments. Many do not include real-time fault diagnosis mechanisms, making them reactive rather than proactive. This motivates the need for fault-tolerant adaptive control strategies that can learn and adjust to unknown faults during operation while ensuring system stability and performance.*

Assumption 4.1. The nonlinear dynamics of the leader robot and follower robots are bounded by positive constants, with the property that the nonlinear function associated with the leader robot, h_o, is bounded as $\|h_o\| \leq \xi$, with $\xi > 0$ for all $t \in \mathbb{R}^+$.

Assumption 4.2. For each follower robot i, the nonlinear function g_i is a symmetric matrix that is either positive-definite or negative-definite. Furthermore, the eigenvalues of g_i, denoted as $\Xi_1(g_i)$, $\Xi_2(g_i)$, …, $\Xi_m(g_i)$, satisfy the inequality:

$$0 < \Xi_1(g_i) \leq \Xi_2(g_i) \leq \cdots \leq \Xi_m(g_i) < \infty,$$

and the matrix g_i is bounded by positive constants.

We assume that $p_i = [p_{i,1}, p_{i,2}, p_{i,3}]^T$, $v_i = [v_{i,1}, v_{i,2}, v_{i,3}]^T$, and $a_i = [a_{i,1}, a_{i,2}, a_{i,3}]^T$ represent the position, velocity, and acceleration states of the robots in the X, Y, and Z directions.

Now we define the tracking error for position, velocity and acceleration as follows:

$$\begin{cases} \lambda_i^p(\tilde{t}) = p_i(\tilde{t}) - p_0(\tilde{t}) - q_p^i(\tilde{t}), \\ \lambda_i^v(\tilde{t}) = v_i(\tilde{t}) - v_0(\tilde{t}) - r_p^i(\tilde{t}), \\ \lambda_i^a(\tilde{t}) = a_i(\tilde{t}) - a_0(\tilde{t}). \end{cases} \tag{4.3}$$

Now we take the position, velocity, and acceleration of the follower robots, and take the derivative of Eq (4.3), and we get

$$\begin{cases} \dot{\lambda}_i^p(\tilde{t}) = & \dot{p}_0(\tilde{t}) + \dot{q}_0^i(\tilde{t}) - \dot{p}_i(\tilde{t}) = \dot{p}_0(\tilde{t}) - \dot{p}_i(\tilde{t}) = \lambda_i^v(\tilde{t}), \\ \dot{\lambda}_i^v(\tilde{t}) = & \dot{v}_0(\tilde{t}) + \dot{r}_0^i(\tilde{t}) - \dot{v}_i(\tilde{t}) = \dot{v}_0(\tilde{t}) - \dot{v}_i(\tilde{t}) = \lambda_i^a(\tilde{t}), \\ \lambda_i^a(\tilde{t}) = & \dot{a}_0(\tilde{t}) - \dot{a}_i(\tilde{t}) = -h_0(p_0(\tilde{t}), v_0(\tilde{t}), a_0(\tilde{t})) + h_i(p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t})) \\ & + g_i(p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t}))(\omega_i(\tilde{t}) + \eta_i(\tilde{t}) + \Delta\omega_i(\tilde{t})). \end{cases} \tag{4.4}$$

Equation (4.4), rewritten as

$$\dot{X}(\tilde{t}) = [\begin{bmatrix} 0 & -I_n & 0 \\ 0 & 0 & -I_n \\ 0 & 0 & 0 \end{bmatrix} \otimes I_m]X(\tilde{t}) + \begin{bmatrix} 0_{nm} \\ 0_{nm} \\ H_f \end{bmatrix} - \begin{bmatrix} 0_{nm} \\ 0_{nm} \\ H_o \end{bmatrix} + \begin{bmatrix} 0_{nm} \\ 0_{nm} \\ \tilde{\eta}_i \end{bmatrix} + \begin{bmatrix} 0_{nm} \\ 0_{nm} \\ \tilde{\omega}_i \end{bmatrix}. \tag{4.5}$$

Tracking error in compact form,

$$\begin{aligned} X =& [\lambda_1^{pT}, \lambda_2^{pT}, ..., \lambda_n^{pT}, \lambda_1^{vT}, \lambda_2^{vT}, ..., \lambda_n^{vT}, \lambda_1^{aT}, \lambda_2^{aT}, ...\lambda_n^{aT}]^T, \\ H_f =& [h_1^T, h_1^T, ..., h_n^T]^T, \quad H_0 = [h_o^T, ..., h_o^T]^T, \\ \tilde{\omega}_i =& [(\overline{\omega}_1 g_1)^T, (\overline{\omega}_2 g_2)^T, (\overline{\omega}_3 g_3)^T, ..., (\overline{\omega}_n g_n)^T]^T, \\ \overline{\omega} =& \omega_i(\tilde{t}) + \Delta\omega_i(\tilde{\tau}), \quad and \quad \tilde{\eta}_i = [\eta_1^T, \eta_2^T, ..., \eta_N^T]^T. \end{aligned}$$

By using Eq (4.3), which represents the tracking error, we can calculate the position error, velocity error, and acceleration error

$$\begin{cases} \mathcal{E}_i^p(\tilde{t}) = \sum_{j\in\mathcal{N}_i}\left(\lambda_i^p(\tilde{t}) - \lambda_j^p(\tilde{t})\right) + \lambda_i^p(\tilde{t}) - \sum_{j\in\mathcal{N}_i}\left(q_p^i(\tilde{t}) - q_p^j(\tilde{t})\right), \\ \mathcal{E}_i^v(\tilde{t}) = \sum_{j\in\mathcal{N}_i}\left(\lambda_i^v(\tilde{t}) - \lambda_j^v(\tilde{t})\right) + \lambda_i^v(\tilde{t}) - \sum_{j\in\mathcal{N}_i}\left(r_p^i(\tilde{t}) - r_p^j(\tilde{t})\right), \\ \mathcal{E}_i^a(\tilde{t}) = \sum_{j\in\mathcal{N}_i}\left(\lambda_i^a(\tilde{t}) - \lambda_j^a(\tilde{t})\right) + \lambda_i^a(\tilde{t}) - \sum_{j\in\mathcal{N}_i}\left(a_i(\tilde{t}) - a_j(\tilde{t})\right). \end{cases} \tag{4.6}$$

4.1. Impact of attacks on robotic systems

In leader-follower robotic control systems, several cyber and physical threats, including actuator attacks, sensor tampering, and denial-of-service (DoS) attacks, can significantly degrade the performance and stability of the system. These types of attacks target core components such as control inputs, sensing mechanisms, and communication channels, which are crucial for the proper functioning of coordinated robotic operations in control systems.

Below, we define these attacks mathematically within the framework of a leader-follower control system.

The actuator attack can be expressed in summation form over the set of neighboring robots as follows:

$$\omega_i^c(\tilde{t}) = \omega_i(\tilde{t}) + \sum_{j\in\mathcal{N}_i} \Xi_j \omega_j^b(\tilde{t}), \tag{4.7}$$

where $\mathcal{N}_i$ represents the set of neighboring robots influencing robot i, $\omega_i^c(\tilde{t})$ denotes the control input at robot i under attack, $\omega_i(\tilde{t})$ is the nominal control input of robot i, $\omega_j^b(\tilde{t})$ is the attack signal injected into the actuator of the neighboring robot j, and Ξ_j is a binary indicator, where $\Xi_j = 1$ signifies that robot j is under attack and $\Xi_j = 0$ otherwise. This formulation generalizes the actuator attack by considering multiple compromised robots in the network.

Similarly, a sensor attack disrupts the measurements that robots rely on for state estimation and feedback control, thereby leading to inaccurate readings. The corrupted sensor measurement at robot i is given by

$$Z_i^c(\tilde{t}) = Z_i(\tilde{t}) + \sum_{j\in\mathcal{N}_i} \Gamma_j Z_j^b(\tilde{t}), \tag{4.8}$$

where $\mathcal{N}_i$ represents the set of neighboring robots influencing robot i, $z_i^c(k)$ denotes the corrupted sensor reading of robot i, $Z_i(\tilde{t})$ is the true measurement of robot i, $Z_j^b(\tilde{t})$ is the attack signal applied to the sensor of the neighboring robot, and Γ_j is a binary indicator, where $\Gamma_j = 1$ signifies that robot is under attack, and $\Gamma_j = 0$ otherwise. The corrupted sensor data affects the robot's internal model, leading to erroneous feedback and poor coordination with the other robots in the system.

By combining Eqs (4.7) and (4.8) we propose a unified attack model for robot

$$\eta_i(\tilde{t}) = \sum_{j\in\mathcal{N}_i} \Xi_j \omega_j^b(\tilde{t}) + V \sum_{j\in\mathcal{N}_i} a_{ij}[\Gamma_j Z_j^b(\tilde{t}) - \Gamma_i(\tilde{t}) Z_i^b(\tilde{t})]. \tag{4.9}$$

4.1.1. Types and mechanisms of sensor and actuator failures

- Actuator manipulation: Modeled via Eq (4.7), malicious or accidental interference injects false control signals into the actuator, either within the robot itself or through compromised neighboring agents. This reduces or alters the effective control authority, causing slow motion, locked actuators, or drift.
- Sensor tampering: Captured in Eq (4.8), this includes bias, drift, or fabricated readings—either intentional or due to faults in the neighboring robots—that distort state estimation and inter-robot feedback loops.
- Integrated fault propagation: The unified attack model combines both input and measurement in Eq (4.9). It explicitly models how faults propagate through networked interactions, undermining coordination and triggering cascading failures. These models mathematically reflect real-world mechanisms—biased sensors, actuator saturation/loss, communication-based attacks—and illustrate how faults can compromise individual agents and ripple across formations.

4.2. Approximation of nonlinear functions using domain-adversarial neural networks

Domain-adversarial neural networks (DANNs) are employed to approximate the unknown nonlinearities represented by $\mathcal{H}_i(p_i, v_i, a_i)$. These functions describe the complex relationships between the system states (position p_i, velocity v_i, and acceleration a_i) and the respective outputs. Specifically, the functions are modeled using neural networks that map the state variables p_i, v_i, and a_i to the

respective outputs, incorporating activation functions and weight matrices to capture the underlying nonlinear behavior. By using DANNs, the nonlinear functions are effectively approximated through a multi-output network, which allows for joint learning of both h_i and g_i. The approximation takes the form

$$\mathcal{H}_i(p_i, v_i, a_i) = \begin{bmatrix} h_i(p_i, v_i, a_i) \\ g_i(p_i, v_i, a_i) \end{bmatrix} = \begin{bmatrix} \mathcal{G}_i^{*T}\mu_1(x_i) + \theta_i(x_i) \\ \mathcal{G}_i^{**T}\mu_2(x_i) + \theta_i^{**}(x_i) \end{bmatrix}, \tag{4.10}$$

where μ_1 and μ_2 are the activation functions, $\mathcal{G}_i^*$ and $\mathcal{G}_i^{**}$ are weight matrices, and θ_i and θ_i^{**} are the approximation errors for h_i and g_i, respectively.

The DANN was introduced in this study as an estimation approach to approximate unknown nonlinear dynamics and external disturbances in real time, thereby enabling adaptive control under uncertainty. In its earlier form [49], the DANN was described in general terms—highlighting its purpose of enhancing robustness—but without a formal specification of the network architecture, learning mechanism, or convergence guarantees.

4.3. Comparison and improved literature on DANNs

In the improved version, the DANN is formally defined as a single-hidden-layer feedforward neural network, where the input layer receives measurable system signals such as tracking errors and their derivatives. These inputs are processed through a hidden layer with nonlinear activation functions (e.g., sigmoid or radial basis functions), and the output layer provides an estimate of the system's unknown nonlinearities. The network weights are updated dynamically in real time using a Lyapunov-based adaptive learning law, allowing the network to adjust continuously without requiring offline training. This real-time adaptation is embedded in the control protocol through expressions like the weight update in [49], a Lyapunov function is constructed to ensure the uniform ultimate boundedness (UUB) of the tracking errors and the estimation errors, thereby guaranteeing convergence of the closed-loop system.

5. DANN-based adaptive control protocol design for robotic systems

In modern robotic systems, ensuring robust control performance is essential when dealing with unpredictable environmental conditions, sensor faults, actuator faults, and external attacks. These systems are highly vulnerable to disturbances such as time delays, sensor attacks, actuator attacks, and DoS attacks while operating in highly volatile and, at times, aggressive environments.

The presence of such challenges severely affects robotic systems, especially in mission-critical applications. To mitigate these issues, we propose a DANN-based control protocol that accounts for the complexities arising from system nonlinearities while maintaining robustness against external perturbations and faults. The neural network–based method is employed to estimate unknown nonlinearities in the system, enabling real-time adaptation informed by observed sensor data and actuator responses. The proposed protocol can handle sensor and actuator faults, manage time delays, and reduce the impact of attacks such as DoS, which are common in communication-heavy robotic systems.

Our control protocol aims to stabilize the robotic system by adjusting its actions based on real-time sensor information and by estimating optimal weight functions to address uncertainties and attacks.

By merging these components, we obtain

$$\omega_i(\tilde{t}) = -\mathcal{K}_i + \frac{\gamma_i}{\tilde{g}_i}, \mathcal{G}_i^T(\tilde{t})\|\mu_i(x_i(\tilde{t})\|^2 * [\mathcal{E}_i^p + \mathcal{E}_i^v + \mathcal{E}_i^a] + \eta_i(\tilde{t}) + \Delta\omega_i(\tilde{t}), \quad i = 1, 2, \ldots, n, \tag{5.1}$$

where $\mathcal{K}_i$ and γ_i are positive design constants that will be specified later, and $\mathcal{G}_i(\tilde{t})$ represents the estimate of the ideal weight $\mathcal{G}_i^*(t)$.

The controller is tailored to handle disturbances, time delays, sensor attacks, actuator attacks, and DoS attacks. By utilizing the DANN-based adaptive mechanism, the control protocol can adjust in real time to these challenges, guaranteeing that the robotic system remains stable and continues to operate optimally despite external difficulties and faults. This method ensures that the robotic system can function efficiently in real-world settings, where unpredictability and harmful attacks are unavoidable.

6. Adaptive dynamic neural network for real-time control of robotics vehicles

In contrast to traditional neural networks, such as RBFNNs and recurrent neural networks (RNNs), which are commonly utilized for time-series modeling and nonlinear function approximation. RBFNNs are recognized for their accuracy in pattern recognition, whereas RNNs are proficient in grasping temporal relationships. Nevertheless, both architectures necessitate substantial computational resources and demand extensive historical data, rendering them less appropriate for real-time applications where flexibility to dynamic changes is vital. To address these issues, we suggest a DANN explicitly tailored for robotic control. In contrast to conventional models that have rigid architectures, the DANN constantly adjusts its network design to improve its learning efficiency and function approximation for monitoring the position, velocity, and acceleration.

As shown in Figure 2, The DANN framework includes three main components: An input layer that captures real-time sensor inputs, a hidden layer featuring flexible neurons that change their configuration according to approximation errors, and an output layer that produces control functions. The learning system is intended to evaluate the originality of new data. When a new data sample satisfies the established novelty threshold, an extra neuron is added to the hidden layer. The center of the new neuron is determined by the incoming sample, and its output weights are adjusted according to the current approximation error of the network.

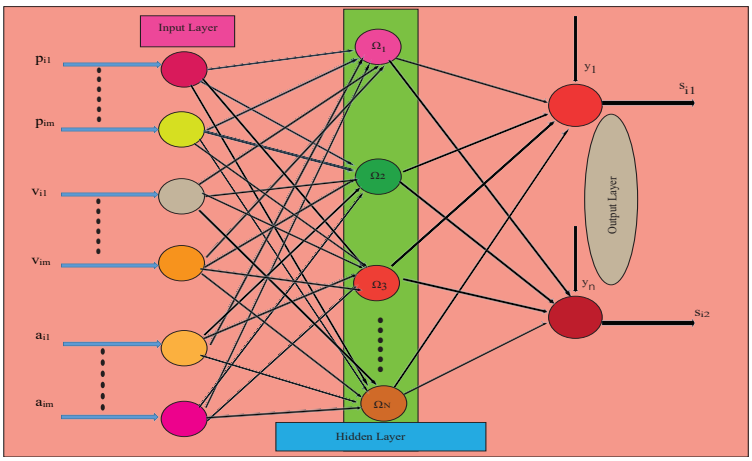

Figure 2. Dynamic neural network adjustment layer for robotic systems.

Furthermore, the activation function's width is adjusted according to the minimum distance between the new input and the existing neuron centers. To maintain efficiency, the network eliminates redundant neurons when their contribution is negligible. The overall adaptation process is illustrated in Figure 3, which highlights the rules governing neurons' addition, modification, and pruning.

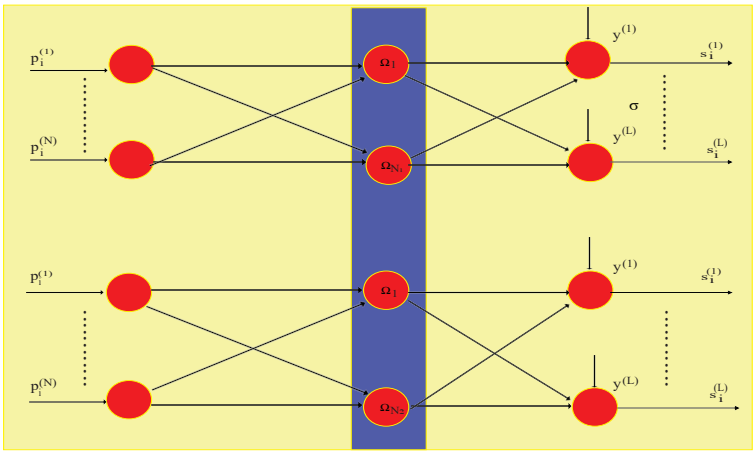

Figure 3. Illustration of the robots before and after applying DANN.

The dynamic nature of the DANN provides significant advantages in robotic control by improving robustness against disturbances, sensor noise, and time-varying uncertainties. The network's ability to restructure itself in real time ensures resilience against sensor faults, actuator failures, time delays, and cyber threats such as DoS attacks. By continuously learning from new patterns and adjusting its internal structure, the DANN enhances its control precision, stability, and reliability in complex and uncertain environments, making it highly effective for real-time motion control in robotic applications.

6.1. Neuronal distribution dissimilarity

To assess the level of dissimilarity between two probability density functions (PDFs), namely $g_p(\tau)$ and $f_q(\tau)$, one widely used method is the Kullback–Leibler information (KLI)

$$\mathcal{I}_{\mathcal{KM}}(g_p \parallel g_q) = \int_{\tau} g_p(\tau) \log\left(\frac{g_p(\tau)}{g_q(\tau)}\right) d\tau.$$

To evaluate the divergence between the two PDFs $g_p(\tau)$ and $g_q(\tau)$, one can define the symmetric Kullback–Leibler divergence as follows:

$$E_{KL}(g_p, g_q) = \mathcal{I}_{\mathcal{KM}(g_p \| g_q)} + \mathcal{I}_{\mathcal{KM}(g_q \| g_p)}.$$

Now, we consider the transformation function defined as:

$$\kappa_i(x) = \left\| \frac{z - \Omega_i}{s_i} \right\| = \frac{(x - \Omega_i)^T (x - \Omega_i)}{s_i^T s_i}, \tag{6.1}$$

where Ω_i and σ_i represent the center and width parameters associated with the $i - th$ neuron. These parameters shape the behavior of the corresponding basis function in the network. The activation

function for the $i-th$ hidden neuron is then expressed as

$$\Psi_i(x) = \exp\left(-\sum_{j=1}^{2n}\left(\frac{x^{(j)}-\Omega_i^{(j)}}{s_i^{(j)}}\right)^2\right) = \exp\left(-\kappa_i(x)\right),$$

where the exponent term quantifies the scaled Euclidean distance between the inputs.

7. Divergence-based discrepancy between neural units

To measure the statistical separation between two probability distributions $g_p(\tau)$ and $f_q(\tau)$, we employ a symmetric form of the Kullback–Leibler divergence, defined as

$$E_{\mathcal{K}}(g_p, g_q) = \int_\tau g_p(\tau)\ln\frac{g_p(\tau)}{g_q(\tau)}\,d\tau + \int_\tau g_q(\tau)\ln\frac{g_q(\tau)}{g_p(\tau)}\,d\tau.$$

This symmetric formulation allows for an unbiased comparison between the two distributions. In the context of neural models, this metric can be utilized to express the dissimilarity between the responses of two hidden neurons. We assume each neuron's activation is governed by a Gaussian-like radial basis function. For the $i-th$ neuron, we define the normalized squared distance as

$$\kappa_i(x) = \left\|\frac{x-\Omega_i}{s_i}\right\|^2 = \frac{(x-\Omega_i)^T(x-\Omega_i)}{s_i^T s_i}. \tag{7.1}$$

The neuron's output is modeled by the activation function,

$$\Psi_i(x) = \exp\left(-\kappa_i(x)\right). \tag{7.2}$$

Using the notion of information divergence, we define the difference between two neurons i and j based on their distributions as

$$\begin{aligned}\text{diff} = D = E_{\mathcal{KM}(g_p,g_q)} &= \int_\tau g_p(\tau)\left(\kappa_i-\kappa_j\right)d\tau + \int_\tau g_q(\tau)\left(\kappa_j-\theta_i\right)d\tau \\ &= \left(\frac{\Omega_i^T\mathbf{1}_{2n}}{2} + \frac{s_i^T s_i\left(\Omega_i^T-2\Omega_j^T\right)\mathbf{1}_{2n}}{2s_j^T\Omega_j}\right)\times\exp\left(-\frac{\Omega_i^T\Omega_i}{s_i^T s_i}\right) \\ &\quad + \left(\frac{\Omega_j^T\mathbf{1}_{2n}}{2} + \frac{s_j^T s_j\left(\Omega_j^T-2\Omega_i^T\right)\mathbf{1}_{2n}}{2s_i^T s_i}\right)\times\exp\left(-\frac{\Omega_j^T\Omega_j}{s_j^T s_j}\right).\end{aligned} \tag{7.3}$$

Here, $\mathbf{1}_{2n}$ is a vector of ones of the required dimension used for inner product summation. The exponential terms capture how strongly each neuron's activation is localized around its center.

7.1. Adaptive expansion of hidden neurons

To dynamically adjust the structure of the neural network, it is important to evaluate how closely the current input z aligns with the representation provided by the existing hidden neurons. This is

achieved by computing a distance-like measure between the input and the center of each hidden unit. The expression used for this purpose is given by

$$\text{dist} = 1 - \exp\left(-\left\|\frac{x-\Omega}{s}\right\|\right), \tag{7.4}$$

where c denotes the center of a hidden neuron and σ represents its spread, which controls the region of influence around the center. A lower value of dist indicates that the input z lies well within the effective domain of an existing hidden neuron, whereas a higher value implies inadequate representation. In such cases, the algorithm may introduce a new hidden neuron to better capture the underlying input distribution.

$$\begin{cases} \text{(i) Increase the hidden neuron count:} & \mathcal{N} \leftarrow \mathcal{N} + 1; \\ \text{(ii) Assign the center and spread for the new hidden neuron:} & \begin{cases} \Omega_N = x, \\ s_N = \lambda \cdot \text{dist}_{\min} \end{cases} \\ \text{(iii) Initialize the connection weights to the output layer:} & W_{\text{out}}^N = \dfrac{ae}{b\mathbf{1}_n + e}. \end{cases};$$

Figures 4–6 show the distance between Gaussian functions, the connection variation, and the fitting error with the number of neurons.

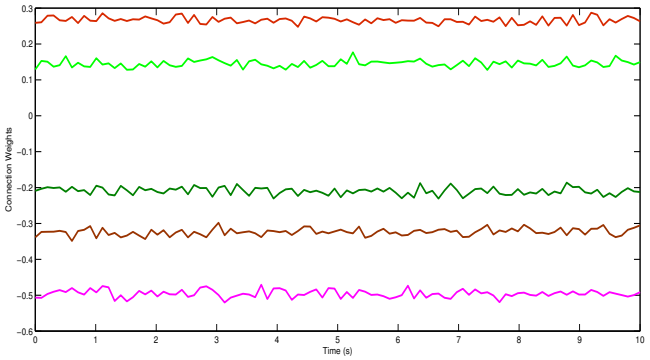

Figure 4. The distance between Gaussian functions.

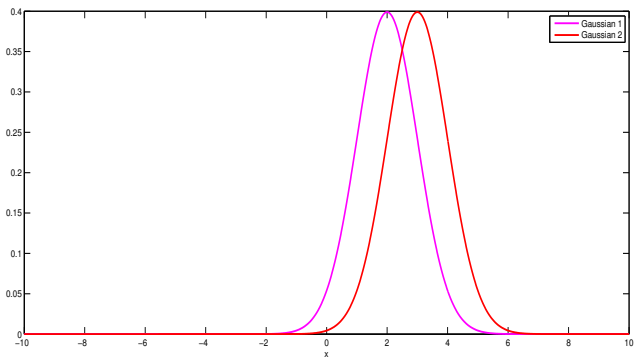

Figure 5. Connection variation of the layers.

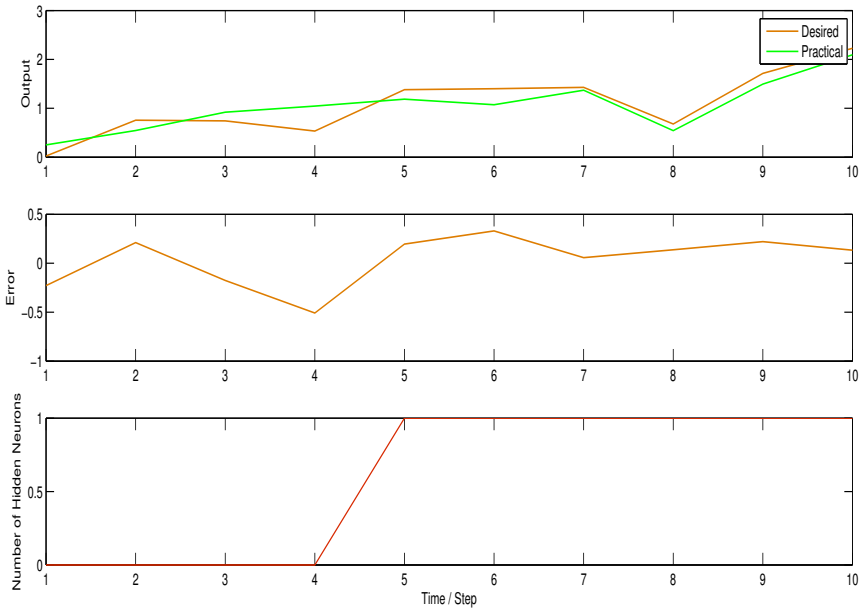

Figure 6. The fitting error with the number of neurons.

Remark 7.1. *Assumption 4.1 is rational and commonly adopted in the literature related to nonlinear multi-agent systems. In practical robotic systems, external disturbances and unknown nonlinearities such as h_o typically arise due to model uncertainties, friction, or environmental interactions. Assuming that these effects are bounded by a known constant ξ allows for a realistic yet analytically tractable framework. This boundedness is essential for constructing Lyapunov functions and proving the system's stability using adaptive or robust control techniques.*

Theorem 7.2 (Stability of the multi-agent system under the proposed control scheme)**.** *Consider a multi-agent system with the Lyapunov candidate function*

$$V(\tilde{t}) = \frac{1}{2}X^T(\tilde{t})\left[\begin{bmatrix} 2\tilde{L} & \tilde{L} & \tilde{L} \\ \tilde{L} & 2\tilde{L} & \tilde{L} \\ \tilde{L} & \tilde{L} & \tilde{L} \end{bmatrix} \otimes I_m\right] X^T(\tilde{t}) + \frac{1}{2}\sum_{i=1}^{n} \beta_i^{-1}\tilde{\mathcal{G}}_i^2(\tilde{t}), \tag{7.5}$$

where

$$\tilde{L} = L + B, B = diag\{b_1, b_2, ..., b_n\}, and\ \tilde{\mathcal{G}}_i(\tilde{t}) = \mathcal{G}_i(\tilde{t}) - \mathcal{G}_i^*(\tilde{t}),$$

under the control law (5.1) and the adaptive tuning law for $\mathcal{G}_i(\tilde{t})$, the system trajectories satisfy

$$V(\tilde{t}) \leq V(0)e^{-\kappa\tilde{t}} + \sum_{i}^{n}\left(\frac{1}{4\gamma_i} + \frac{\overline{h}_0^2}{4\Xi_i} + \frac{\overline{\theta}_i^2}{4\rho_i} + \frac{1}{2}\beta_i^{-1}\mathcal{G}_i^{*2}(\tilde{t})\right)\frac{(1 - e^{-\kappa\tilde{t}})}{\kappa}, \tag{7.6}$$

where $\kappa = \min\{\sigma_1\beta_1, \sigma_2\beta_2, \ldots, \sigma_n\beta_n\}$, Consequently, as $\tilde{t} \to \infty$, the multi-agent system is globally uniformly ultimately bounded (GUUB), and the tracking error converges to a residual set whose size depends on the disturbance bounds, attack, and approximation errors.

Proof. First, we design the following Lyapunov candidate function,

$$V(\tilde{t}) = \frac{1}{2}X^T(\tilde{t})\left[\begin{bmatrix} 2\tilde{L} & \tilde{L} & \tilde{L} \\ \tilde{L} & 2\tilde{L} & \tilde{L} \\ \tilde{L} & \tilde{L} & \tilde{L} \end{bmatrix} \otimes I_m\right] X^T(\tilde{t}) + \frac{1}{2}\sum_{i=1}^{n} \beta_i^{-1}\tilde{\mathcal{G}}_i^2(\tilde{t}), \tag{7.7}$$

where $\tilde{L} = L + B$, $B = \text{diag}\{b_1, b_2, ..., b_n\}$, and $\tilde{\mathcal{G}}_i(\tilde{t}) = \mathcal{G}_i(\tilde{t}) - \mathcal{G}_i^*(\tilde{t})$.

By taking the derivative of Eq (7.7) with respect to time, we have

$$\begin{aligned}
\dot{V}(\tilde{t}) =& -\frac{1}{2}X^T(\tilde{t})\left[\left[\begin{bmatrix} 0 & -I_n & 0 \\ 0 & 0 & -I_n \\ 0 & 0 & 0 \end{bmatrix}^T \begin{bmatrix} 2\tilde{L} & \tilde{L} & \tilde{L} \\ \tilde{L} & 2\tilde{L} & \tilde{L} \\ \tilde{L} & \tilde{L} & \tilde{L} \end{bmatrix} + \begin{bmatrix} 2\tilde{L} & \tilde{L} & \tilde{L} \\ \tilde{L} & 2\tilde{L} & \tilde{L} \\ \tilde{L} & \tilde{L} & \tilde{L} \end{bmatrix}\begin{bmatrix} 0 & -I_n & 0 \\ 0 & 0 & -I_n \\ 0 & 0 & 0 \end{bmatrix}\right] \otimes I_m\right] X(\tilde{t}) \\
&+ X^T(\tilde{t})\left[\begin{bmatrix} 2\tilde{L} & \tilde{L} & \tilde{L} \\ \tilde{L} & 2\tilde{L} & \tilde{L} \\ \tilde{L} & \tilde{L} & \tilde{L} \end{bmatrix} \otimes I_m\right]\begin{bmatrix} 0_{nm} \\ 0_{nm} \\ H_f \end{bmatrix} - X^T(\tilde{t})\left[\begin{bmatrix} 2\tilde{L} & \tilde{L} & \tilde{L} \\ \tilde{L} & 2\tilde{L} & \tilde{L} \\ \tilde{L} & \tilde{L} & \tilde{L} \end{bmatrix} \otimes I_m\right]\begin{bmatrix} 0_{nm} \\ 0_{nm} \\ H_0 \end{bmatrix} \\
&+ X^T(\tilde{t})\left[\begin{bmatrix} 2\tilde{L} & \tilde{L} & \tilde{L} \\ \tilde{L} & 2\tilde{L} & \tilde{L} \\ \tilde{L} & \tilde{L} & \tilde{L} \end{bmatrix} \otimes I_m\right]\begin{bmatrix} 0_{nm} \\ 0_{nm} \\ \tilde{\eta}_i \end{bmatrix} + X^T(\tilde{t})\left[\begin{bmatrix} 2\tilde{L} & \tilde{L} & \tilde{L} \\ \tilde{L} & 2\tilde{L} & \tilde{L} \\ \tilde{L} & \tilde{L} & \tilde{L} \end{bmatrix} \otimes I_m\right]\begin{bmatrix} 0_{nm} \\ 0_{nm} \\ \tilde{\omega}_i \end{bmatrix} \\
&+ \sum_{i=1}^{n} \beta_i^{-1}\tilde{\mathcal{G}}_i(\tilde{t})\dot{\mathcal{G}}_i(\tilde{t})
\end{aligned} \tag{7.8}$$

$$\begin{aligned}
=& X^T(\tilde{t})\left[\begin{bmatrix} 0 & -\tilde{L} & -\frac{\tilde{L}}{2} \\ -\tilde{L} & -\tilde{L} & -\frac{3\tilde{L}}{2} \\ -\frac{\tilde{L}}{2} & -\frac{3L}{2} & -\tilde{L} \end{bmatrix} \otimes I_m\right] X(\tilde{t}) + \sum_{i}^{n}(\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t}))^T h_i((p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t})) \\
&- \sum_{i}^{n}(\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t}))^T h_0((p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t})) + \sum_{i}^{n}(\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) \\
&+ \mathcal{E}_i^a(\tilde{t}))^T g_i(p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t})) * (\omega_i(\tilde{t}) + 2\eta_i(\tilde{t}) + 2\Delta\omega_i(\tilde{t})) + \sum_{i=1}^{n} \beta_i^{-1}\tilde{\mathcal{G}}_i(\tilde{t})\dot{\mathcal{G}}_i(\tilde{t}).
\end{aligned} \tag{7.9}$$

By applying the DANN approximation of the follower's unknown nonlinearity to Eq (7.9),

$$\begin{aligned}
=& X^T(\tilde{t})\left[\begin{bmatrix} 0 & -\tilde{L} & -\frac{\tilde{L}}{2} \\ -\tilde{L} & -\tilde{L} & -\frac{3\tilde{L}}{2} \\ -\frac{\tilde{L}}{2} & -\frac{3L}{2} & -\tilde{L} \end{bmatrix} \otimes I_m\right] X(\tilde{t}) + \sum_{i}^{n}(\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t}))^T(\mathcal{G}_i^T(\tilde{t})\mu_i(x_i(\tilde{t})) + \theta_i(x_i(\tilde{t}))) \\
&- \sum_{i}^{n}(\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t}))^T h_0((p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t})) + \sum_{i}^{n}(\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t}))^T g_i(p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t})) \\
&* (\omega_i(\tilde{t}) + \eta_i(\tilde{t}) + \Delta\omega_i(\tilde{t})) + \sum_{i=1}^{n} \beta_i^{-1}\tilde{\mathcal{G}}_i(\tilde{t})\dot{\mathcal{G}}_i(\tilde{t}).
\end{aligned} \tag{7.10}$$

By using Young's and Cauchy inequality, we have

$$(\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t}))^T \mathcal{G}_i^T(\tilde{t})\mu_i(x_i(\tilde{t})) \leq \gamma_i \left((\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t}))^T \mathcal{G}_i^T(\tilde{t})\mu_i(x_i(\tilde{t}))\right)^2 + \frac{1}{4\gamma_i}$$

$$\leq \gamma_i \mathcal{G}_i^T(\tilde{t})\|\mu_i(x_i(\tilde{t})\|^2 \|\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t})\|^2 + \frac{1}{4\gamma_i} - (\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t}))^T h_0((p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t}))$$

$$\leq \Xi_i \|\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t})\|^2 + \frac{\overline{h}_0^2}{4\Xi_i}(\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t}))^T \theta_i(x_i(\tilde{t}))$$

$$\leq \rho_i \|\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t})\|^2 + \frac{\overline{\theta}_i^2}{4\rho_i}. \tag{7.11}$$

Using Eq (7.11) in Eq (7.10), we have

$$\dot{V}(\tilde{t}) \leq - X^T(\tilde{t}) \left[\begin{bmatrix} 0 & -\tilde{L} & -\frac{\tilde{L}}{2} \\ -\tilde{L} & -\tilde{L} & -\frac{3\tilde{L}}{2} \\ -\frac{\tilde{L}}{2} & -\frac{3L}{2} & -\tilde{L} \end{bmatrix} \otimes I_m \right] X(\tilde{t}) + \sum_i^n \gamma_i \mathcal{G}_i^T(\tilde{t})\|\mu_i(x_i(\tilde{t})\|^2 \|\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t})\|^2$$

$$\sum_i^n (\Xi_i + \rho_i)\|\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t})\|^2 + \sum_i^n (\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t}))^T g_i(p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t}))$$

$$(\omega_i(\tilde{t}) + \eta_i(\tilde{t}) + \Delta\omega_i(\tilde{t})) - \sum_{i=1}^n \beta_i^{-1} \tilde{\mathcal{G}}_i(\tilde{t})\mathcal{G}_i(\tilde{t}) + \sum_i^n (\frac{1}{4\gamma_i} + \frac{\overline{h}_0^2}{4\Xi_i} + \frac{\overline{\theta}_i^2}{4\rho_i}). \tag{7.12}$$

Now using Eq (5.1) in Eq (7.12), we have

$$\dot{V}(\tilde{t}) \leq - X^T(\tilde{t}) \left[\begin{bmatrix} 0 & -\tilde{L} & -\frac{\tilde{L}}{2} \\ -\tilde{L} & -\tilde{L} & -\frac{3\tilde{L}}{2} \\ -\frac{\tilde{L}}{2} & -\frac{3L}{2} & -\tilde{L} \end{bmatrix} \otimes I_m \right] X(\tilde{t}) + \sum_i^n \gamma_i \mathcal{G}_i^T(\tilde{t})\|\mu_i(x_i(\tilde{t})\|^2 \|\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t})$$

$$+ \mathcal{E}_i^a(\tilde{t})\|^2 \sum_i^n (\Xi_i + \rho_i)\|\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t})\|^2 - \sum_i^n (\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t}))^T g_i(p_i(\tilde{t}), v_i(\tilde{t}), a_i(\tilde{t}))$$

$$\times (\mathcal{K}_i + \frac{\gamma_i}{\tilde{g}_i}, \mathcal{G}_i^T(\tilde{t})\|\mu_i(x_i(\tilde{t})\|^2 \times [\mathcal{E}_i^p + \mathcal{E}_i^v + \mathcal{E}_i^a] + 2\eta_i(\tilde{t}) + 2\Delta\omega_i(\tilde{t})) - \sum_{i=1}^n \beta_i^{-1} \tilde{\mathcal{G}}_i(\tilde{t})\mathcal{G}_i(\tilde{t})$$

$$+ \sum_i^n (\frac{1}{4\gamma_i} + \frac{\overline{h}_0^2}{4\Xi_i} + \frac{\overline{\theta}_i^2}{4\rho_i}). \tag{7.13}$$

By Assumption 4.2, the inequality (7.13) becomes

$$\dot{V}(\tilde{t}) \leq - X^T(\tilde{t}) \left[\begin{bmatrix} 0 & -\tilde{L} & -\frac{\tilde{L}}{2} \\ -\tilde{L} & -\tilde{L} & -\frac{3\tilde{L}}{2} \\ -\frac{\tilde{L}}{2} & -\frac{3L}{2} & -\tilde{L} \end{bmatrix} \otimes I_m \right] X(\tilde{t}) + \sum_i^n \gamma_i \mathcal{G}_i^T(\tilde{t})\|\mu_i(x_i(\tilde{t})\|^2 \|\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t})$$

$$+ \mathcal{E}_i^a(\tilde{t})\|^2 \sum_i^n (\Xi_i + \rho_i)\|\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t})\|^2 - \sum_i^n (\tilde{g}_i \mathcal{K}_i + \gamma_i \mathcal{G}_i^T(\tilde{t})\|\mu_i(x_i(\tilde{t})\|^2 + \frac{2\eta_i(\tilde{t}) + 2\Delta\omega_i(\tilde{t}))}{\zeta})$$

$$\times \|\mathcal{E}_i^p + \mathcal{E}_i^v + \mathcal{E}_i^a\|^2 - \sum_{i=1}^n \beta_i^{-1} \tilde{\mathcal{G}}_i(\tilde{t})\mathcal{G}_i(\tilde{t}) + \sum_i^n (\frac{1}{4\gamma_i} + \frac{\overline{h}_0^2}{4\Xi_i} + \frac{\overline{\theta}_i^2}{4\rho_i}) \tag{7.14}$$

$$\leq - X^T(\tilde{t})\left[\begin{bmatrix} 0 & -\tilde{L} & -\frac{\tilde{L}}{2} \\ -\tilde{L} & -\tilde{L} & -\frac{3\tilde{L}}{2} \\ -\frac{\tilde{L}}{2} & -\frac{3L}{2} & -\tilde{L} \end{bmatrix} \otimes I_m\right] X(\tilde{t}) + \sum_i^n (\Xi_i + \rho_i)\|\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t})\|^2 - \sum_i^n (\tilde{g}_i\mathcal{K}_i$$

$$+ \frac{2\eta_i(\tilde{t}) + 2\Delta\omega_i(\tilde{t}))}{\zeta}) \times \|\mathcal{E}_i^p + \mathcal{E}_i^v + \mathcal{E}_i^a\|^2 - \sum_{i=1}^n \beta_i^{-1}\tilde{\mathcal{G}}_i(\tilde{t})\mathcal{G}_i(\tilde{t}) + \sum_i^n (\frac{1}{4\gamma_i} + \frac{\overline{h}_0^2}{4\Xi_i} + \frac{\overline{\theta}_i^2}{4\rho_i}). \tag{7.15}$$

Let $\frac{2\eta_i(\tilde{t})+2\Delta\omega_i(\tilde{t})}{\zeta} = \omega_0$. If $\tilde{g}_i\mathcal{K}_i \geq \Xi_i + \rho_i + \omega_0 + v_o$, then Eq (7.15) becomes

$$\begin{aligned}
\overline{V}(\tilde{t}) =& - X^T(\tilde{t})\left[\begin{bmatrix} 0 & -\tilde{L} & -\frac{\tilde{L}}{2} \\ -\tilde{L} & -\tilde{L} & -\frac{3\tilde{L}}{2} \\ -\frac{\tilde{L}}{2} & -\frac{3L}{2} & -\tilde{L} \end{bmatrix} \otimes I_m\right] X(\tilde{t}) - \sum_i^n v_o\|\mathcal{E}_i^p(\tilde{t}) + \mathcal{E}_i^v(\tilde{t}) + \mathcal{E}_i^a(\tilde{t})\|^2 \\
& - \sum_{i=1}^n \beta_i^{-1}\tilde{\mathcal{G}}_i(\tilde{t})\mathcal{G}_i(\tilde{t}) + \sum_i^n (\frac{1}{4\gamma_i} + \frac{\overline{h}_0^2}{4\Xi_i} + \frac{\overline{\theta}_i^2}{4\rho_i}) \\
\leq & - X^T(\tilde{t})\left[v_o \begin{bmatrix} \tilde{L} & 0 & 0 \\ 0 & \tilde{L} & 0 \\ 0 & 0 & \tilde{L} \end{bmatrix}^T \begin{bmatrix} \tilde{L} & 0 & 0 \\ 0 & \tilde{L} & 0 \\ 0 & 0 & \tilde{L} \end{bmatrix} - \begin{bmatrix} 0 & \tilde{L} & \frac{\tilde{L}}{2} \\ \tilde{L} & \tilde{L} & \frac{3\tilde{L}}{2} \\ \frac{\tilde{L}}{2} & \frac{3\tilde{L}}{2} & \tilde{L} \end{bmatrix} \otimes I_m\right] \\
& - \sum_{i=1}^n \beta_i^{-1}\tilde{\mathcal{G}}_i(\tilde{t})\mathcal{G}_i(\tilde{t}) + \sum_i^n (\frac{1}{4\gamma_i} + \frac{\overline{h}_0^2}{4\Xi_i} + \frac{\overline{\theta}_i^2}{4\rho_i}).
\end{aligned} \tag{7.16}$$

Because $\tilde{\mathcal{G}}_i(\tilde{t})\mathcal{G}_i(\tilde{t}) = \frac{1}{2}(\tilde{\mathcal{G}}_i(\tilde{t}) + \mathcal{G}_i(\tilde{t}) - \mathcal{G}_i^{2*}(\tilde{t}))$, which implies that

$$-\beta_i^{-1}\tilde{\mathcal{G}}_i(\tilde{t})\mathcal{G}_i(\tilde{t}) \leq -\frac{1}{2}\beta_i^{-1}\sum_i^n(-\tilde{\mathcal{G}}_i(\tilde{t}) + \mathcal{G}_i^{2*}(\tilde{t})). \tag{7.17}$$

By using the equation above, Eq (7.16) becomes

$$\begin{aligned}
\overline{V}(\tilde{t}) \leq & - X^T(\tilde{t})\left[v_o \begin{bmatrix} \tilde{L} & 0 & 0 \\ 0 & \tilde{L} & 0 \\ 0 & 0 & \tilde{L} \end{bmatrix}^T \begin{bmatrix} \tilde{L} & 0 & 0 \\ 0 & \tilde{L} & 0 \\ 0 & 0 & \tilde{L} \end{bmatrix} - \begin{bmatrix} 0 & \tilde{L} & \frac{\tilde{L}}{2} \\ \tilde{L} & \tilde{L} & \frac{3\tilde{L}}{2} \\ \frac{\tilde{L}}{2} & \frac{3\tilde{L}}{2} & \tilde{L} \end{bmatrix} \otimes I_m\right] \\
& - \frac{1}{2}\sum_{i=1}^n \beta_i^{-1}\tilde{\mathcal{G}}_i(\tilde{t}) + \sum_i^n (\frac{1}{4\gamma_i} + \frac{\overline{h}_0^2}{4\Xi_i} + \frac{\overline{\theta}_i^2}{4\rho_i} + \frac{1}{2}\beta_i^{-1}\mathcal{G}_i^{*2}(\tilde{t})),
\end{aligned} \tag{7.18}$$

let

$$\mathcal{A} = \begin{bmatrix} \tilde{L} & 0 & 0 \\ 0 & \tilde{L} & 0 \\ 0 & 0 & \tilde{L} \end{bmatrix}^T \begin{bmatrix} \tilde{L} & 0 & 0 \\ 0 & \tilde{L} & 0 \\ 0 & 0 & \tilde{L} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 0 & \tilde{L} & \frac{\tilde{L}}{2} \\ \tilde{L} & \tilde{L} & \frac{3\tilde{L}}{2} \\ \frac{\tilde{L}}{2} & \frac{3\tilde{L}}{2} & \tilde{L} \end{bmatrix} \text{ and } C = \begin{bmatrix} 2\tilde{L} & \tilde{L} & \tilde{L} \\ \tilde{L} & 2\tilde{L} & \tilde{L} \\ \tilde{L} & \tilde{L} & \tilde{L} \end{bmatrix},$$

we define

$$v_0 > \Xi_{\min}^{-1}(\mathcal{A})\left(\Xi_{\max}(\Delta) + \frac{\kappa}{2}\Xi_{\max}(C)\right), \tag{7.19}$$

where $\Xi_{\min}(\mathcal{A})$, $\Xi_{\max}(\mathcal{B})$, and $\Xi_{\max}(C)$ denote the smallest eigenvalue of the Matrix $\mathcal{A}$, the largest eigenvalue of the Matrix $\mathcal{B}$, and the largest eigenvalue of the Matrix C, respectively. Here

$$\kappa = \min\{\sigma_1\beta_1, \sigma_2\beta_2, \ldots, \sigma_n\beta_n\}, \tag{7.20}$$

we can thus rewrite Eq (7.18) as

$$\dot{V}(\tilde{t}) \leq -\frac{\kappa}{2}X^T(\tilde{t})(C \otimes I_m)X(\tilde{t}) - \frac{1}{2}\sum_{i=1}^{n}\beta_i^{-1}\tilde{\mathcal{G}}_i^2(t) + \sum_{i=1}^{n}\left(\frac{1}{4\beta_i} + \frac{f_o^2}{4\lambda_i} + \frac{\varepsilon_i^2}{4\delta_i} + \frac{1}{2}\gamma_i^{-1}W_i^{*2}(t)\right)$$
$$-\kappa V(\tilde{t}) + \sum_{i}^{n}(\frac{1}{4\gamma_i} + \frac{\overline{h}_0^2}{4\Xi_i} + \frac{\overline{\theta}_i^2}{4\rho_i} + \frac{1}{2}\beta_i^{-1}\mathcal{G}_i^{*2}(\tilde{t})). \tag{7.21}$$

It follows that

$$V(\tilde{t}) \leq V(0)e^{-\kappa\tilde{t}} + \frac{\Omega}{\kappa}\left(1 - e^{-\kappa\tilde{t}}\right), \tag{7.22}$$

where

$$\Omega = \sum_{i}^{n}(\frac{1}{4\gamma_i} + \frac{\overline{h}_0^2}{4\Xi_i} + \frac{\overline{\theta}_i^2}{4\rho_i} + \frac{1}{2}\beta_i^{-1}\mathcal{G}_i^{*2}(\tilde{t})). \tag{7.23}$$

7.2. Explanation of parameter selection and resulting decay rate

From the inequality derived at the end of the theorem:

$$V(\tilde{t}) \leq V(0)e^{-\kappa\tilde{t}} + \frac{\Omega}{\kappa}\left(1 - e^{-\kappa\tilde{t}}\right), \tag{7.24}$$

we observe that the Lyapunov function $V(\tilde{t})$ exponentially decays to a residual set determined by $\frac{\Omega}{\kappa}$, where

$$\Omega = \sum_{i=1}^{n}\left(\frac{1}{4\gamma_i} + \frac{\overline{h}_0^2}{4\Xi_i} + \frac{\overline{\theta}_i^2}{4\rho_i} + \frac{1}{2}\beta_i^{-1}\mathcal{G}_i^{*2}(\tilde{t})\right). \tag{7.25}$$

The parameters γ_i, Ξ_i, and ρ_i directly appear in the denominators of the additive terms within Ω. Thus, by selecting sufficiently large values for these parameters, we can make each term in Ω arbitrarily small, effectively reducing the ultimate bound $\frac{\Omega}{\kappa}$ of the Lyapunov function.

Moreover, the term $-\kappa V(\tilde{t})$ in the derivative of $V(\tilde{t})$ ensures that the system exhibits exponential decay. Therefore, careful selection of γ_i, Ξ_i, and ρ_i enhances the rate of convergence and minimizes the residual error, ensuring that the tracking error is ultimately bounded within a small neighborhood of the origin. This satisfies the sufficient conditions of Lemma 2.1,thereby guaranteeing the practical stability of the closed-loop system.

Remark 7.3. *The Lyapunov function $V(\tilde{t})$ is non-negative because it is constructed as a sum of two non-negative terms:*

- *A quadratic form $X^T(\tilde{t})(Q \otimes I_m)X(\tilde{t})$ with $Q \geq 0$, ensuring the term is always non-negative.*
- *The squared estimation error term $\sum_{i=1}^{n}\beta_i^{-1}\tilde{\mathcal{G}}_i^2(\tilde{t}) \geq 0$, which is also non-negative by definition.*

Therefore, $V(\tilde{t}) \geq 0$ for all $\tilde{t}$, and it serves as a valid Lyapunov candidate function.

□

8. Numerical examples

In the numerical experiments, we present two examples to authenticate the theoretical proof of our result, each involving two different communication topologies for robotic vehicles. We assume there are six follower robots and one leader, as shown in Figure 7.

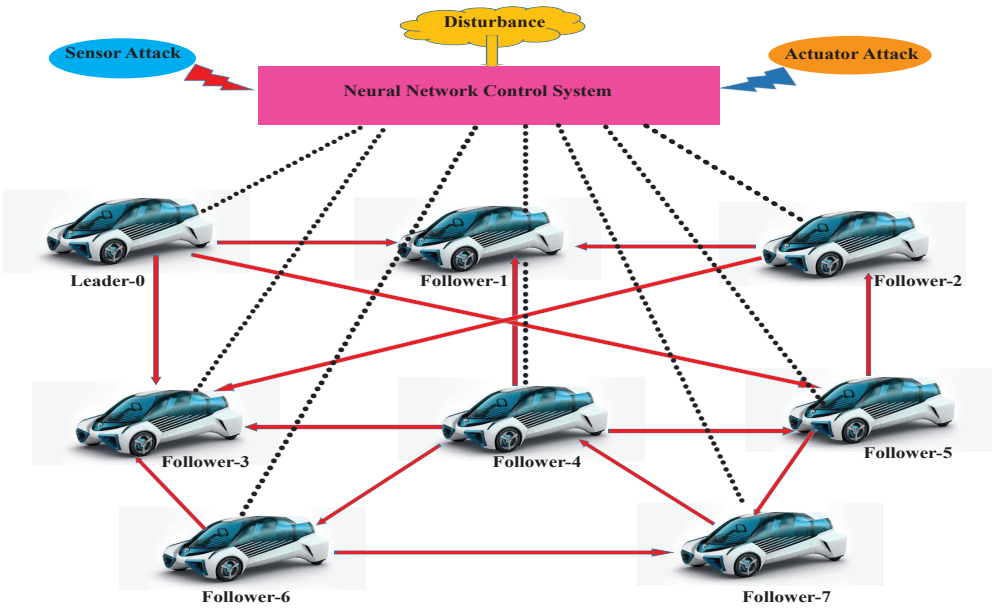

Figure 7. Structure of one-leader and six-follower robotic system.

Adjacency and Laplacian matrices between the robotic vehicles are taken as

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix},$$

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}.$$

We assume the following function from Eq (4.3), $\omega_j^b(\tilde{t}) = \sin(\tilde{t})$, $Z_j^b(\tilde{t}) = \cos(\tilde{t})$, $\omega_i(\tilde{t}) = \cos(\tilde{t})$, $Z_i(\tilde{t}) = \sin(\tilde{t})$, and $\Delta\omega_i(\tilde{t}) = 20cos(0.5)$ and we choose the parameters, $\Xi_j = 1$, $\Gamma_j = 1$, $V = 2$, $t = 10$. Thus, the unified attack model at time $t = 10$ becomes

$$\eta_i(10) = 2\sin(10) + 2\left[\cos(10) - \sin(10)\right] \approx 1.6782.$$

Let the leader's velocity $v_0(\tilde{t}) \in \mathbb{R}^3$ be defined as

$$v_0(\tilde{t}) = \begin{bmatrix} 3\sin(\tilde{t}) \\ 1 + \cos(2\tilde{t} + 1) \\ \cos\left(0.25\tilde{t}^2 + 1\right) \end{bmatrix},$$

then acceleration defined as

$$a_0(\tilde{t}) = \begin{bmatrix} 3cos(\tilde{t}) \\ -2sin(2\tilde{t} + 1) \\ -0.5tsin(0.25\tilde{t}^2 + 1) \end{bmatrix},$$

$$p_0(\tilde{t}) = \begin{bmatrix} 5 \\ 10 \\ 5 \end{bmatrix}, v_0(\tilde{t}) = \begin{bmatrix} 10 \\ 15 \\ 10 \end{bmatrix}.$$

The initial position of follower vehicle i is a random vector within the range $[-20, 20]$. The dynamics of the leader defined as

$$\dot{p}_0(\tilde{t}) = v_0(\tilde{t}),$$

$$\dot{v}_0(\tilde{t}) = a_o(\tilde{t})$$

and

$$\dot{a}_0(\tilde{t}) = \begin{bmatrix} -sin(p_{01}(\tilde{t}) + 0.2(1 - v_{01}(t)cos(a_{01}) \\ -cos(p_{01}(\tilde{t}) + 0.3(1 + v_{01}(t)sin(a_{01}) \end{bmatrix}.$$

The followers' dynamics of robots

$$\dot{p}_i(\tilde{t}) = v_i(\tilde{t}),$$

$$\dot{v}_i(\tilde{t}) = a_i(\tilde{t})$$

and

$$\dot{a}_i(\tilde{t}) = \begin{bmatrix} -a_{i1}sin(p_{i1}(\tilde{t}) + b_{i1}(1 - v_{i1}(t)cos(a_{i1}) \\ a_{i2}cos(p_{i2}(\tilde{t}) + b_{i2}(1 + c_{i2}v_{i2}(t)sin(a_{i2}) \end{bmatrix} \times \begin{bmatrix} \alpha_{i1} + a_{i1}cos(v_{i1}) & 0 \\ 0 & \alpha_{i2}sin(v_{i2}a_{i2}) \end{bmatrix} \begin{bmatrix} \omega_{i1} \\ \omega_{i2} \end{bmatrix} \begin{bmatrix} \eta_{i1} + \Delta\omega_{i1} \\ \eta_{i2} + \Delta\omega_{i2} \end{bmatrix},$$

where

$$i = 1, 2, ...7,$$

$$a = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 0.9 \\ 0 & 1 \\ -1 & 0.5 \\ 0.5 & -0.3 \\ 0.3 & -0.4 \end{bmatrix}, b = \begin{bmatrix} 1.1 & 1.2 \\ 1.5 & 0.3 \\ -0.5 & 0.5 \\ -1 & 0.6 \\ 0.8 & 1 \\ 0.7 & -0.2 \end{bmatrix},$$

$$c = \begin{bmatrix} 0.1 & 0.5 \\ -0.5 & 0.5 \\ 0.6 & 0.7 \\ 1 & 1.5 \\ 0 & 1.3 \\ 0.8 & 0.9 \end{bmatrix}, \alpha = \begin{bmatrix} 0.2 & 1.3 \\ 0.2 & 0.6 \\ 0.5 & 3.2 \\ -0.4 & 0.6 \\ 0.6 & 3.5 \\ 2.1 & 3 \end{bmatrix}.$$

This time-varying velocity vector guides the desired trajectory of the leader in a 3D space. The followers aim to track the leader while maintaining the desired formation and ensuring robust consensus under the given interaction topology.

Figures 8 and 9 show the trajectories of positions p_i for $i = 0, 1, 2, \ldots, 7$. Figures 10 and 11 show the trajectories of velocity v_i for $i = 0, 1, 2, \ldots, 7$. Figures 12 and 13 show the trajectories of acceleration a_i for $i = 0, 1, 2, \ldots, 7$. Figures 14 and 15 represent the output error of the DANN and the self-tuning procedure of the DANN, respectively. Figures 16 and 17 represent the self-tuning procedure of hidden neurons in the DANN and the total number of neurons, respectively.

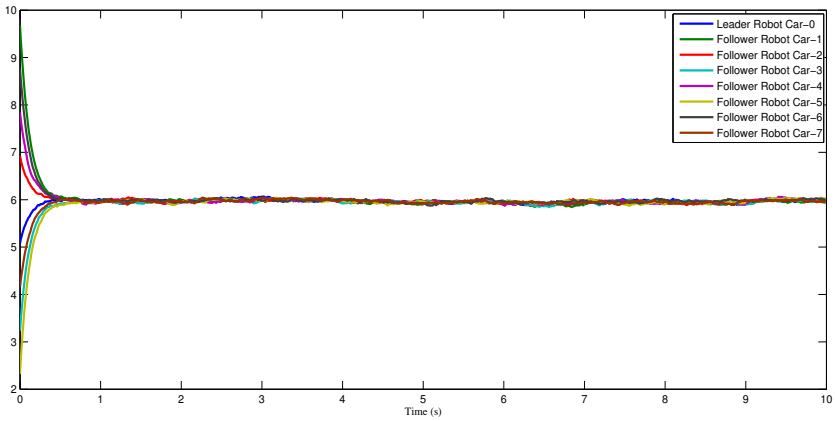

Figure 8. The trajectories of position p_{i1} for $i = 0, 1, 2, ..., 7$.

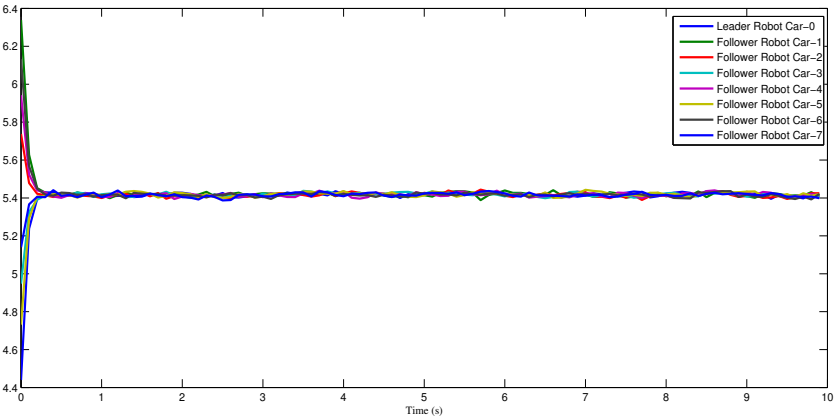

Figure 9. The trajectories of position p_{i2} for $i = 0, 1, 2, ..., 7$.

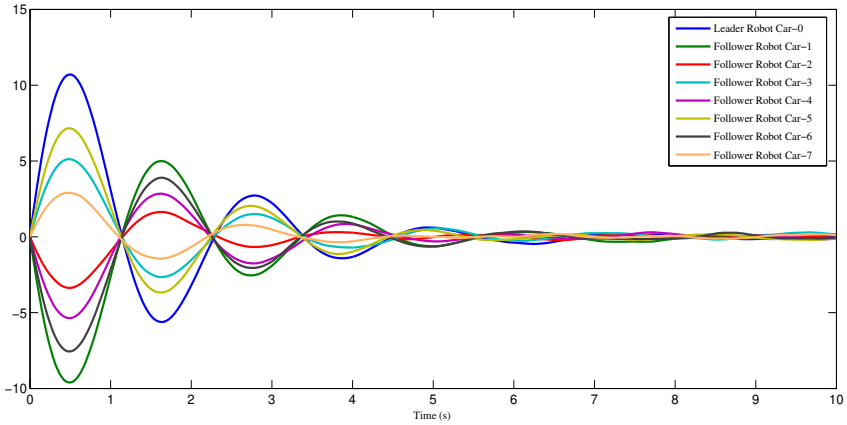

Figure 10. The trajectories of velocity v_{i1} for $i = 0, 1, 2, ..., 7$.

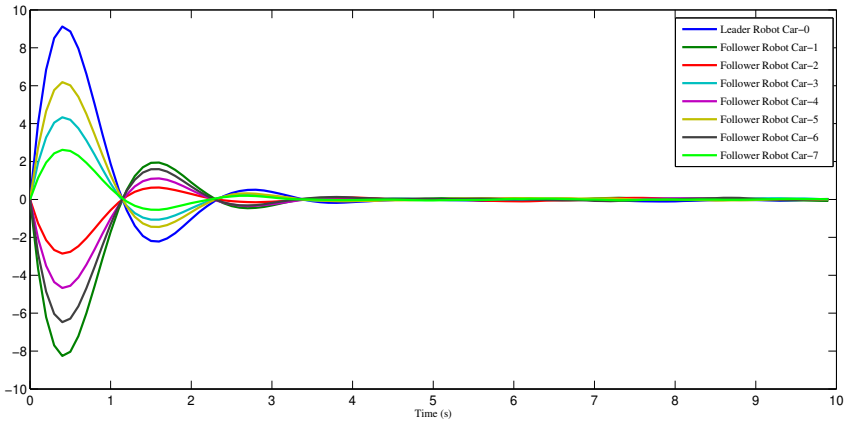

Figure 11. The trajectories of velocity v_{i2} for $i = 0, 1, 2, ..., 7$.

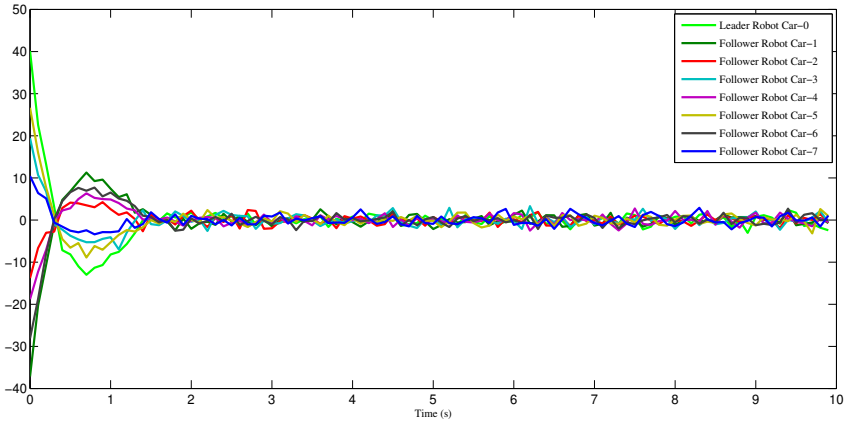

Figure 12. The trajectories of acceleration a_{i1} for $i = 0, 1, 2, ..., 7$.

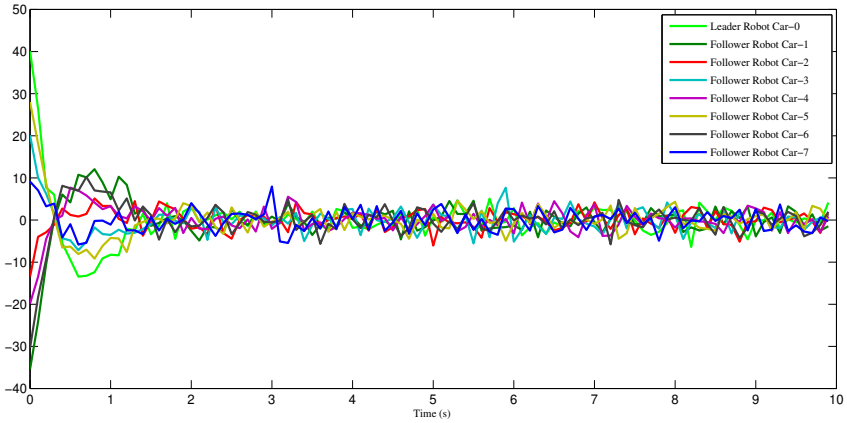

Figure 13. The trajectories of acceleration a_{i2} for $i = 0, 1, 2, ..., 7$.

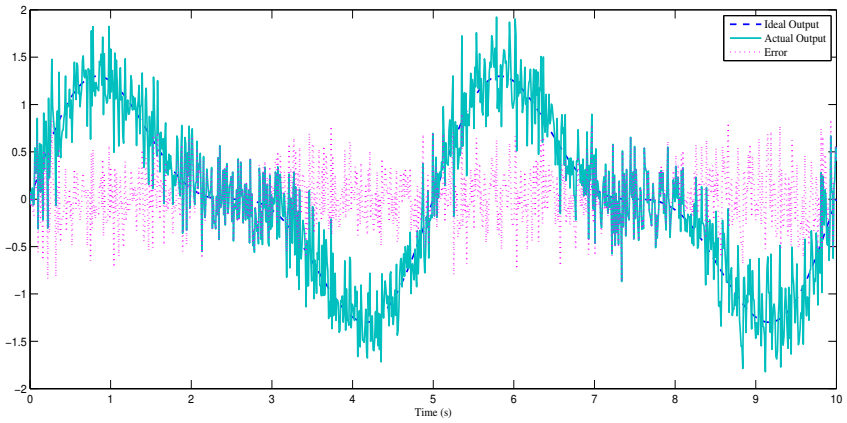

Figure 14. Output error of the DANN.

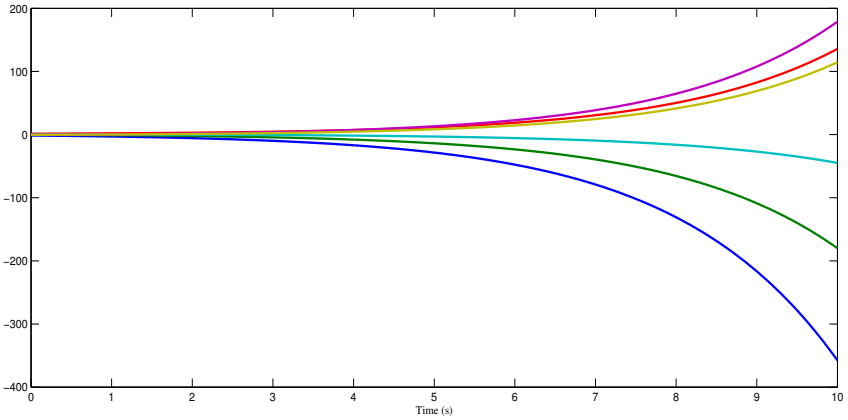

Figure 15. The self-tuning procedure of the DANN.

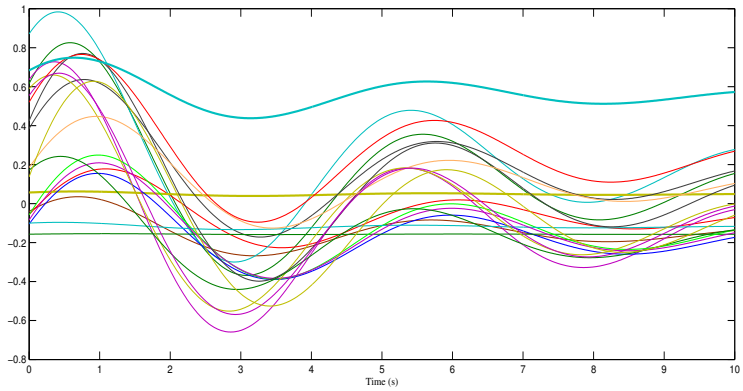

Figure 16. The self-tuning procedure of the hidden neurons in the DANN.

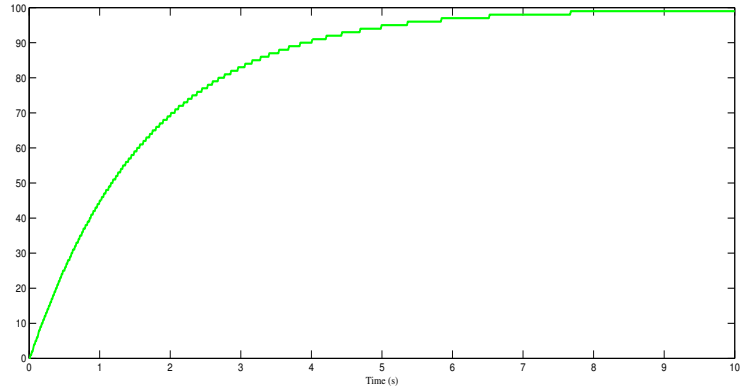

Figure 17. Total number of neurons in the DANN.

8.1. Discussion of challenges and contributions compared with existing works

Compared with existing methods, our proposed approach faces and overcomes several significant challenges.

- **Unknown fault dynamics:** Traditional approaches often assume that the fault models are known or can be approximated offline. In contrast, we consider real-time faults with unknown characteristics. To address this, we employ adaptive neural networks that learn the system's nonlinearities online, without relying on predefined fault structures.
- **Real-time fault isolation:** Most existing control frameworks lack mechanisms for immediate fault isolation. Our design incorporates a real-time estimation structure that isolates the effects of sensor and actuator faults during operation.
- **Stability under concurrent faults:** Ensuring consensus stability when both the sensors and actuators fail simultaneously is nontrivial. We address this challenge by designing a composite Lyapunov function that guarantees bounded system trajectories under such adverse conditions.
- **Scalability and implementation:** Many fault-tolerant control methods are computationally intensive or require full-state measurements. In contrast, our proposed controller is distributed,

relies only on local information, and is lightweight enough for real-time deployment in embedded robotic systems.

These advances improve the system's reliability and resilience compared with existing results in fault-tolerant multi-agent coordination.

Remark 8.1. *The proposed DANN-based fault-tolerant control approach shares conceptual similarities with the distributed estimator-based fuzzy containment control in [50] and the dynamic event-triggered fuzzy adaptive pinning control in [51], yet it provides an enhanced adaptive learning capability and fault resilience mechanism. These features make it a promising candidate for integration with our extension of the cited frameworks, enabling improved performance under more challenging and uncertain operational conditions.*

9. Conclusions

This paper presented a fault-tolerant neural control framework for cooperative coordination in multi-robotic car systems operating under sensor and actuator failures. The proposed approach leverages a DANN to adaptively handle unknown nonlinearities and partial component malfunctions without requiring prior fault models. By embedding an adaptive estimation mechanism and a real-time fault detection and isolation process directly into the control loop, the method provides strong resilience against unpredictable disturbances and degraded hardware performance. Stability and convergence guarantees were rigorously established using Lyapunov-based analysis, ensuring bounded tracking errors and reliable formation consensus.

The framework offers several notable advantages, including a distributed adaptive neural controller capable of real-time adjustments to faults, an integrated FDI mechanism for rapid isolation and compensation, and proven robustness in maintaining both velocity tracking and formation stability across diverse fault scenarios. Simulation results confirm that the method is well-suited for safety-critical and autonomous vehicular applications where sensor and actuator reliability cannot be guaranteed.

Looking ahead, further work will focus on enhancing the fault model to reflect more realistic operating conditions, such as stochastic disturbances, intermittent and unpredictable fault activation patterns, and correlated multi-source failures. Additional efforts will include integrating comprehensive performance metrics—such as settling time and control effort—for more rigorous evaluation, as well as benchmarking against advanced fault-tolerant strategies. Beyond these extensions, an important research direction lies in deploying the framework for heterogeneous robot teams with network-induced delays, exploring scalability in large-scale cooperative systems, and ultimately implementing hardware-based real-time experiments to verify and refine the approach in real-world environments.

Author contributions

Muflih Alhazmi: Software, project administration; Waqar Ul Hassan: Writing original draft, writing-review and editing; Mohammed M. A. Almazah: Conceptualization; Saadia Rehman: Formal analysis; Azmat Ullah Khan Niazi: Writing original draft, writing-review and editing, supervision;

Somayya Komal: Validation; Nafisa A. Albasheir: Resources. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through the Large Research Project under grant number RGP. 2/103/46 and Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia, for funding this research work through the project number NBU-FPEJ-2025-871-05.

Code availability

The code is considered an intellectual property of the University of Lahore, Sargodha campus, and therefore is not publicly available.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. J. Lv, X. Ju, C. Wang, Neural network prescribed-time observer-based output-feedback control for uncertain pure-feedback nonlinear systems, *Expert Syst. Appl.*, **264** (2025), 125813. https://doi.org/10.1016/j.eswa.2024.125813
2. A. Tong, J. Zhang, L. Xie, Intelligent fault diagnosis of rolling bearing based on Gramian angular difference field and improved dual attention residual network, *Sensors*, **24** (2024), 2156. https://doi.org/10.3390/s24072156
3. Q. Wang, L. Chen, G. Xiao, P. Wang, Y. Gu, J. Lu, Elevator fault diagnosis based on digital twin and PINNs-e-RGCN, *Sci. Rep.*, **14** (2024), 30713. https://doi.org/10.1038/s41598-024-78784-7
4. S. Wang, C. Zheng, T. Ma, T. Wang, S. Gao, Q. Dai, et al., Tooth backlash inspired comb-shaped single-electrode triboelectric nanogenerator for self-powered condition monitoring of gear transmission, *Nano Energy*, **123** (2024), 109429. https://doi.org/10.1016/j.nanoen.2024.109429
5. K. Deng, L. Yang, Y. Lu, S. Ma, Multitype chatter detection via multichannel internal and external signals in robotic milling, *Measurement*, **229** (2024), 114417. https://doi.org/10.1016/j.measurement.2024.114417
6. M. F. Li, Y. Liu, T. Wang, F. Chu, Z. Peng, Adaptive synchronous demodulation transform with application to analyzing multicomponent signals for machinery fault diagnostics, *Mech. Syst. Signal Process.*, **191** (2023), 110208. https://doi.org/10.1016/j.ymssp.2023.110208

7. Y. Yao, F. Shu, X. Cheng, H. Liu, P. Miao, L. Wu, Automotive radar optimization design in a spectrally crowded V2I communication environment, *IEEE Trans. Intell. Transp. Syst.*, **24** (2023), 8253–8263. https://doi.org/10.1109/TITS.2023.3264507
8. X. Xu, B. Li, Semi-global stabilization of parabolic PDE–ODE systems with input saturation, *Automatica*, **171** (2025), 111931. https://doi.org/10.1016/j.automatica.2024.111931
9. M. Li, T. Wang, F. Chu, Q. Han, Z. Qin, M. J. Zuo, Scaling-basis chirplet transform, *IEEE Trans. Ind. Electron.*, **68** (2020), 8777–8788. https://doi.org/10.1109/TIE.2020.3013537
10. Y. H. Lan, J. Y. Zhao, Improving track performance by combining Padé-approximation-based preview repetitive control and equivalent-input-disturbance, *J. Electr. Eng. Technol.*, **19** (2024), 3781–3794. https://doi.org/10.1007/s42835-024-01830-x
11. J. Hang, G. Qiu, M. Hao, S. Ding, Improved fault diagnosis method for permanent magnet synchronous machine system based on lightweight multi-source information data layer fusion, *IEEE Trans. Power Electron.*, **39** (2024), 13808–13817. https://doi.org/10.1109/TPEL.2024.3432163
12. W. He, J. Hang, S. Ding, L. Sun, W. Hua, Robust diagnosis of partial demagnetization fault in PMSMs using radial air-gap flux density under complex working conditions, *IEEE Trans. Ind. Electron.*, **71** (2024), 12001–12010. https://doi.org/10.1109/TIE.2024.3349520
13. T. Wu, H. Yang, P. Wang, C. Zhang, M. Zhang, Data-driven fatigue reliability evaluation of offshore wind turbines under floating ice loading, *J. Struct. Eng.*, **150** (2024), 05024004. https://doi.org/10.1061/JSENDH.STENG-13476
14. Z. Zeng, C. Zhu, S. M. Goetz, Fault-tolerant multiparallel three-phase two-level converters with adaptive hardware reconfiguration, *IEEE Trans. Power Electron.*, **39** (2024), 3925–3930. https://doi.org/10.1109/TPEL.2024.3350186
15. H. Xu, X. Cui, Y. Liu, Y. Xu, X. Ling, Y. Li, Effect of fastening system failure on the micromechanical properties of railway ballast bed, *KSCE J. Civ. Eng.*, **29** (2025), 100233. https://doi.org/10.1016/j.kscej.2025.100233
16. Z. Zhang, G. Chen, W. Chen, R. Jia, G. Chen, L. Zhang, A joint learning of force feedback of robotic manipulation and textual cues for granular materials classification, *IEEE Robot. Autom. Lett.*, **10** (2025), 7166–7173. https://doi.org/10.1109/LRA.2025.3575322
17. B. Zhu, R. Tang, J. Zhao, P. Zhang, W. Li, X. Cao, et al., Critical scenarios adversarial generation method for intelligent vehicles testing based on hierarchical reinforcement architecture, *Accid. Anal. Prev.*, **215** (2025), 108013. https://doi.org/10.1016/j.aap.2025.108013
18. B. Zhu, X. Cao, P. Zhang, J. Zhao, J. Han, R. Tang, High-fidelity ultrasonic radar in-the-loop accelerated test for automatic parking systems, *IEEE T. Intell. Transp. Syst.*, 2025, 1–13. https://doi.org/10.1109/TITS.2025.3574505
19. J. Tian, Y. Zhou, L. Yin, S. A. Alqahtani, M. Tang, S. Lu, et al., Control structures and algorithms for force feedback bilateral teleoperation systems: A comprehensive review, *Comput. Model. Eng. Sci.*, **142** (2025), 973–1019. https://doi.org/10.32604/cmes.2024.057261

20. J. Wang, Y. Song, T. He, A novel adaptive monitoring framework for detecting the abnormal states of aero-engines with maneuvering flight data, *Reliab. Eng. Syst. Safe.*, **258** (2025), 110910. https://doi.org/10.1016/j.ress.2025.110910

21. Z. Zhou, Y. Wang, G. Zhou, K. Nam, Z. Ji, C. Yin, A twisted Gaussian risk model considering target vehicle longitudinal-lateral motion states for host vehicle trajectory planning, *IEEE Trans. Intell. Transp. Syst.*, **24** (2023), 13685–13697. https://doi.org/10.1109/TITS.2023.3298110

22. J. Guo, Y. Li, B. Huang, L. Ding, H. Gao, M. Zhong, An online optimization escape entrapment strategy for planetary rovers based on Bayesian optimization, *J. Field Robot.*, **41** (2024), 2518–2529. https://doi.org/10.1002/rob.22361

23. T. Li, H. Shi, X. Bai, N. Li, K. Zhang, Rolling bearing performance assessment with degradation twin modeling considering interdependent fault evolution, *Mech. Syst. Signal Process.*, **224** (2025), 112194. https://doi.org/10.1016/j.ymssp.2024.112194

24. X. Zhao, T. Wang, Y. Li, B. Zhang, K. Liu, D. Liu, et al., Target-driven visual navigation by using causal intervention, *IEEE Trans. Intell. Veh.*, **9** (2024), 1294–1304. https://doi.org/10.1109/TIV.2023.3288810

25. L. Li, A. Cherouat, H. Snoussi, T. Wang, Grasping with occlusion-aware ally method in complex scenes, *IEEE Trans. Autom. Sci. Eng.*, **22** (2024), 5944–5954. https://doi.org/10.1109/TASE.2024.3434610

26. C. Zhang, J. Qiao, S. Wang, R. Chen, H. Dui, Y. Zhang, et al., Importance measures based on system performance loss for multi-state phased-mission systems, *Reliab. Eng. Syst. Saf.*, **256** (2025), 110776. https://doi.org/10.1016/j.ress.2024.110776

27. G. Du, H. Zhang, H. Yu, P. Hou, J. He, S. Cao, L. Ma, Study on automatic tracking system of microwave deicing device for railway contact wire, *IEEE Trans. Instrum. Meas.*, **73** (2024), 1–11. https://doi.org/10.1109/TIM.2024.3446638

28. J. Wang, Y. Wu, C. L. P. Chen, Z. Liu, W. Wu, Adaptive PI event-triggered control for MIMO nonlinear systems with input delay, *Inform. Sci.*, **677** (2024), 120817. https://doi.org/10.1016/j.ins.2024.120817

29. F. Ding, K. Zhu, J. Liu, C. Peng, Y. Wang, J. Lu, Adaptive memory event triggered output feedback finite-time lane keeping control for autonomous heavy truck with roll prevention, *IEEE Trans. Fuzzy Syst.*, **32** (2024), 6607–6621. https://doi.org/10.1109/TFUZZ.2024.3454344

30. X. Zhang, Y. Liu, X. Chen, Z. Li, C. Y. Su, Adaptive pseudoinverse control for constrained hysteretic nonlinear systems and its application on dielectric elastomer actuator, *IEEE/ASME Trans. Mechatron.*, **28** (2023), 2142–2154. https://doi.org/10.1109/TMECH.2022.3231263

31. X. Chen, J. Cui, Y. Liu, X. Zhang, J. Sun, R. Ai, et al., Joint scene flow estimation and moving object segmentation on rotational LiDAR data, *IEEE Trans. Intell. Transp. Syst.*, **25** (2024), 17733–17743. https://doi.org/10.1109/TITS.2024.3432755

32. W. Jiang, B. Zheng, D. Sheng, X. Li, A compensation approach for magnetic encoder error based on improved deep belief network algorithm, *Sens. Actuators A Phys.*, **366** (2024), 115003. https://doi.org/10.1016/j.sna.2023.115003

33. J. Xiao, Y. Ren, J. Du, Y. Zhao, S. Kumari, M. J. Alenazi, et al., CALRA: Practical conditional anonymous and leakage-resilient authentication scheme for vehicular crowdsensing communication, *IEEE Trans. Intell. Transp. Syst.*, **26** (2025), 1273–1285. https://doi.org/10.1109/TITS.2024.3488741
34. J. Wu, Y. Wang, C. Yin, Curvilinear multilane merging and platooning with bounded control in curved road coordinates, *IEEE Trans. Veh. Technol.*, **71** (2022), 1237–1252. https://doi.org/10.1109/TVT.2021.3131751
35. J. Joshi, A. K. Singh, Performance analysis of routing protocols for vehicular networks, *Int. J. Veh. Inform. Commun. Syst.*, **9** (2024), 60–80. https://doi.org/10.1504/IJVICS.2024.136276
36. S. Zhang, Feature-aware task offloading and scheduling mechanism in vehicle edge computing environment, *Int. J. Veh. Inform. Commun. Syst.*, **9** (2024), 415–433. https://doi.org/10.1504/IJVICS.2024.142101
37. J. Chen, Y. Wang, Y. Cui, H. Wang, K. Polat, F. Alenezi, EEG-based multi-band functional connectivity using corrected amplitude envelope correlation for identifying unfavorable driving states, *Comput. Methods Biomech. Biomed. Eng.*, 2025. https://doi.org/10.1080/10255842.2025.2488502
38. H. Wang, Y. F. Li, T. Men, L. Li, Physically interpretable wavelet-guided networks with dynamic frequency decomposition for machine intelligence fault prediction, *IEEE Trans. Syst. Man Cybern. Syst.*, **54** (2024), 4863–4875. https://doi.org/10.1109/TSMC.2024.3389068
39. F. Wang, K. Chen, S. Zhen, X. Chen, H. Zheng, Z. Wang, Prescribed performance adaptive robust control for robotic manipulators with fuzzy uncertainty, *IEEE Trans. Fuzzy Syst.*, **32** (2024), 1318–1330. https://doi.org/10.1109/TFUZZ.2023.3323090
40. H. Qi, L. Ding, M. Zheng, L. Huang, H. Gao, G. Liu, et al., Variable wheelbase control of wheeled mobile robots with worm-inspired creeping gait strategy, *IEEE Trans. Robot.*, **40** (2024), 3271–3289. https://doi.org/10.1109/TRO.2024.3400947
41. G. Lu, Q. Lin, D. Zheng, P. Zhang, Online degradation fault prognosis for DC-link capacitors in multistring-connected photovoltaic boost converters subject to cable uncertainties, *IEEE J. Emerg. Sel. Topics Power Electron.*, **13** (2025), 1107–1117. https://doi.org/10.1109/JESTPE.2024.3502255
42. D. He, Y. Lin, Z. Dai, S. X. Yang, Neurodynamics-based visual servo predictive control for improving smooth movement of logistics omnidirectional robots, *IEEE Trans. Ind. Electron.*, 2025, 1–10. https://doi.org/10.1109/TIE.2025.3585028
43. Y. Li, Y. Jiang, J. Lu, C. Tan, Improved active disturbance rejection control for electro-hydrostatic actuators via actor-critic reinforcement learning, *Eng. Appl. Artif. Intell.*, **158** (2025), 111485. https://doi.org/10.1016/j.engappai.2025.111485
44. Z. Wang, D. Gao, Y. Lu, K. Deng, Z. Yuan, M. Huang, et al., A mutual cross-attention fusion network for surface roughness prediction in robotic machining process using internal and external signals, *J. Manuf. Syst.*, **82** (2025), 284–300. https://doi.org/10.1016/j.jmsy.2025.06.018
45. W. Ji, X. Lin, Y. Sun, G. Lin, A. Vulevic, Intelligent fault-tolerant control for high-speed maglev transportation based on error-driven adaptive fuzzy online compensator, *IEEE Trans. Intell. Transp. Syst.*, 2025, 1–10. https://doi.org/10.1109/TITS.2025.3549624

46. H. Liu, S. Zhen, X. Liu, H. Zheng, L. Gao, Y. H. Chen, Robust approximate constraint following control design for collaborative robots system and experimental validation, *Robotica*, **42** (2024), 3957–3975. https://doi.org/10.1017/S0263574724001760

47. C. Guo, J. Hu, Time base generator-based practical predefined-time stabilization of high-order systems with unknown disturbance, *IEEE Trans. Circuits Syst. II*, **70** (2023), 2670–2674. https://doi.org/10.1109/TCSII.2023.3242856

48. B. Xu, X. Wang, J. Zhang, Y. Guo, A. A. Razzaqi, A novel adaptive filtering for cooperative localization under compass failure and non-gaussian noise, *IEEE Trans. Veh. Technol.*, **71** (2022), 3737–3749. https://doi.org/10.1109/TVT.2022.3145095

49. P. Wang, M. Gao, J. Li, A. Zhang, Dynamic adjustment neural network-based cooperative control for vehicle platoons with state constraints, *Int. J. Appl. Math. Comput. Sci.*, **34** (2024), 211–224. https://doi.org/10.61822/amcs-2024-0015

50. H. Ma, Q. Zhou, H. Ren, Z. Wang, Distributed estimator-based fuzzy containment control for nonlinear multiagent systems with deferred constraints, *IEEE Trans. Fuzzy Syst.*, **33** (2025), 2074–2083. https://doi.org/10.1109/TFUZZ.2025.3550864

51. H. Ren, L. Cao, H. Ma, H. Li, Dynamic event-triggered-based fuzzy adaptive pinning control for multiagent systems with output saturation, *IEEE Trans. Fuzzy Syst.*, **33** (2025), 1277–1286. https://doi.org/10.1109/TFUZZ.2024.3519186

AIMS Press