Int. J. Appl. Math. Comput. Sci., *2024*, *Vol. 34*, *No. 2*, *211–224*
DOI: 10.61822/amcs-2024-0015

DYNAMIC ADJUSTMENT NEURAL NETWORK–BASED COOPERATIVE CONTROL FOR VEHICLE PLATOONS WITH STATE CONSTRAINTS

PING WANG [a], MIN GAO [a], JUNYU LI [a,*], ANGUO ZHANG [b]

[a] School of Mechanical and Electrical Engineering
Hefei Technology College
No. 2, Daihe Road, Xinzhan District, Hefei 230009, China
e-mail: 1152475543@qq.com

[b] Institute of Microelectronics
University of Macau
Avenida da Universidade, Taipa 999007, Macau, China

This paper addresses the challenge of managing state constraints in vehicle platoons, including maintaining safe distances and aligning velocities, which are key factors that contribute to performance degradation in platoon control. Traditional platoon control strategies, which rely on a constant time-headway policy, often lead to deteriorated performance and even instability, primarily during dynamic traffic conditions involving vehicle acceleration and deceleration. The underlying issue is the inadequacy of these methods to adapt to variable time-delays and to accurately modulate the spacing and speed among vehicles. To address these challenges, we propose a dynamic adjustment neural network (DANN) based cooperative control scheme. The proposed strategy employs neural networks to continuously learn and adjust to time varying conditions, thus enabling precise control of each vehicle's state within the platoon. By integrating a DANN into the platoon control system, we ensure that both velocity and inter-vehicular spacing adapt in response to real-time traffic dynamics. The efficacy of our proposed control approach is validated using both Lyapunov stability theory and numeric simulation, which confirms substantial gains in stability and velocity tracking of the vehicle platoon.

Keywords: vehicle platoon, dynamic adjustment neural network (DANN), cooperative control, state constraint.

1. Introduction

The advent of advanced vehicular technology, alongside the evolution of the transportation infrastructure and the sophistication of communication networks, has facilitated the emergence of cooperative vehicle platoon systems. Foundational research in this area, as highlighted by Li *et al.* (2015; 2020b), has been the cornerstone on which these cutting-edge systems have been conceptualized and developed.

Central to this contemporary transport scheme is the concept of vehicle platooning. This technique organizes a sequence of vehicles into a cohesive convoy, maintaining minimal yet safe distances, guided by intelligent control systems dedicated to optimizing navigation efficiency (Hu *et al.*, 2023; Huang *et al.*, 2019). The implementation of such systems has been shown to yield multiple transport-related benefits including, but not limited to, the enhancement of road safety (Dutta *et al.*, 2022; Zhang *et al.*, 2021), reduction in energy and fuel consumption (Earnhardta *et al.*, 2022; Li *et al.*, 2021), as well as relieving traffic congestion, thereby improving the overall comfort for passengers.

The effectiveness of vehicle platooning is intrinsically tied to the efficacy of its control algorithms (Chang *et al.*, 2019; Prayitno *et al.*, 2023). These algorithms are integral for real-time adjustment of inter-vehicle distances and coordinating the velocities of individual vehicles within the platoon. Their role becomes even more pronounced as they tackle and adapt to dynamic traffic conditions and environmental changes (Wang *et al.*, 2023). The ingenuity of these control systems thus directly influences the platooning system's overall efficiency, seamlessly blending each vehicle's movements into a collective, autonomous

[*] Corresponding author

flow that speaks to a future of smarter, safer, and more sustainable transportation.

Historically, the quest for maintaining platoon formation and ensuring stability has led to the exploration of various cooperative control techniques. These have included the implementation of a linear quadratic regulator (LQR) (Liu *et al*., 2018; Okamoto *et al*., 2004), model predictive control (MPC) (Hu *et al*., 2022; Wang *et al*., 2022; Zhou *et al*., 2012), and adaptive sliding mode control (SMC) (Gao *et al*., 2018; Liu *et al*., 2019). Such methodologies have been pivotal in addressing specific platooning challenges. However, they often fall short in managing the uncertainties and non-linear dynamics characteristic of real-world driving conditions.

Neural networks have emerged as a beacon of hope for devising control algorithms that excel in adaptability, robustness, and high performance for vehicle platooning systems (Huang *et al*., 2023; Liang *et al*., 2020; Liu *et al*., 2022; Wu *et al*., 2022). These neural network-based strategies have proven effective in preserving platoon stability and safety amidst intricacies such as complex topologies (Li *et al*., 2020a; Feng *et al*., 2022), undetermined parameters (Liu *et al*., 2019; Peng *et al*., 2021), disruptions (Zhao *et al*., 2017; Li *et al*., 2022), and communication lags (Li *et al*., 2019; Gong *et al*., 2023). However, they typically overlook the intricacies of state constraints and the dynamic interplay between vehicles (Gao *et al*., 2016), potentially undermining the fidelity and the safeguarding mechanisms of the platoon system.

Our investigation presents an innovative neural network model designed to encapsulate the state constraints and the vehicular dynamics. We introduce the dynamic adjustment neural network (DANN), a sophisticated model characterized by its ability for online learning and dynamic adjustment of hidden neuron quantities and weights. The DANN encompasses an adept skill set for approximating system nonlinearities accurately while preserving a compact network size. This unique attribute empowers the DANN to dynamically tweak its network structure during the control process, providing a robust, computationally friendly solution adaptable to a spectrum of uncertain conditions. In the realm of neural networks, the DANN presents a significant advantage with its ability to adjust its weights and structure adaptively through online learning. This renders the DANN highly efficient for approximating nonlinear functions with high accuracy, particularly in scenarios where the model dynamics may change over time or due to environmental factors. Unlike traditional RBFNNs and recurrent neural networks, which require extensive storage and computational resources, DANNs offer a flexible, resource-conscious alternative that dynamically alters their neuron parameters and counts to match input data novelty.

This paper makes several contributions to the field:

(i) We unveil the DANN model, which ensures stable and safe platoon configurations in adherence to strict state constraints and accommodates on-the-fly modulation of control parameters in alignment with prevailing states and objectives.

(ii) We advocate for an advanced, neural network-driven cooperative control strategy that incorporates considerations for state constraints and dynamic inter-vehicle exchanges, overcoming the shortcomings of traditional frameworks.

The paper is organized as follows. Section 2 delineates the preliminaries relevant to graph theory and frames the problem. Section 3 details the intricacies of the DANN model. Section 4 explicates our neural network-centric cooperative control algorithm, articulating its foundational principles and rationale. Section 5 empirically validates our algorithm's efficiency through simulation outcomes. Concluding remarks are presented in Section 6, encapsulating the essence and implications of our research.

2. Preliminaries and problem formulation

2.1. Graph theory. Consider a directed graph $\mathcal{G} = (V, E, A)$, composed of n nodes indexed by the set $V = \{1, 2, \dots, n\}$. The set $E \subseteq V \times V$ represents the edges of the graph, consisting of the ordered node pairs. The associated adjacency weight matrix $A = [w_{ij}] \in \mathbb{R}^{n\times n}$ is a nonnegative matrix, where the entry $w_{ij} > 0$ signifies the existence of an edge from node j to node i, indicated by $(j, i) \in E$; conversely, $w_{ij} = 0$ denotes the absence of such an edge. It is crucial to distinguish that in an undirected graph the presence of an edge $(j, i) \in E$ automatically implies the presence of $(i, j) \in E$. This reciprocity does not necessarily apply in the context of a directed graph, where the existence of $(j, i) \in E$ does not infer that (i, j) is also in E.

Additionally, each node i has a neighbor set $\mathcal{N}_i$ which comprises the nodes from which node i is able to receive information, reflecting the directional nature of communications within the graph.

Define the adjacency Laplacian matrix $L = [l_{ij}] \in \mathbb{R}^{n\times n}$ as

$$l_{ii} = \begin{cases} \sum_{k\in\mathcal{N}_i} w_{ik} > 0 & \text{if } i = j, \\ -w_{ij} < 0 & \text{if } i \neq j. \end{cases} \tag{1}$$

What should be paid attention to is that for an undirected graph, L is a symmetric semi-positive matrix.

Definition 1. (*Strong and weak connectivity*) Consider a directed graph $\mathcal{G}$. The graph is said to be strongly connected if, for every pair of distinct nodes v_i and v_j, there is a bidirectional path such that a path from node v_i to node v_j exists, along with a path from node v_j to node v_i. Conversely, the graph is termed weakly connected if, for any pair of distinct nodes v_i and v_j, there is at least a unidirectional path—meaning there exists a path from v_i to v_j or from v_j to v_i, but not necessarily both.

For undirected graphs, the definition simplifies: The graph is considered connected if there is a path linking any pair of distinct nodes.

Definition 2. (*In-degree and out-degree, balance and Laplacian*) Within a directed graph $\mathcal{G}$ with the weighted adjacency matrix $A = [a_{ij}]$, the in-degree and out-degree for a node v_i are quantitatively defined as follows:

$$d_{\text{in}}(v_i) = \sum_{j=1}^{n} a_{ji}, \quad d_{\text{out}}(v_i) = \sum_{j=1}^{n} a_{ij}. \tag{2}$$

When the in-degree and out-degree are equivalent, that is, $d_{\text{in}}(v_i) = d_{\text{out}}(v_i)$ for each node v_i, then the graph $\mathcal{G}$ is described as a balanced graph.

Moreover, by constructing a diagonal matrix $D = \operatorname{diag}(d_{\text{out}}(v_1), \dots, d_{\text{out}}(v_n))$, we define the Laplacian matrix L for the graph $\mathcal{G}$ as

$$L = D - A. \tag{3}$$

An important property of the Laplacian matrix is made evident by the equation $LI_n = 0$, where I_n represents the identity matrix of order n.

2.2. Related lemmas.

Lemma 1. (Corets *et al.*, 2002) *Suppose there exists a bounded continuous function $V(t) > 0$ for $\forall t \geq 0$. If $\dot{V}(t) \leq -\alpha V(t) + \beta$, where both α and β are positive constant parameters, then*

$$V(t) \leq \left(V(0) - \frac{\beta}{\alpha}\right) e^{-\alpha t} + \frac{\beta}{\alpha}. \tag{4}$$

Lemma 2. (Young's inequality (Mitrinovic *et al.*, 1993b)) *For any $x, y \in \mathbb{R}$ and $\epsilon > 0$, the following inequalities hold:*

$$\begin{aligned} xy &\leq \frac{\epsilon}{2}x^2 + \frac{1}{2\epsilon}y^2, \\ x^2 + y^2 &\leq (x+y)^2 \quad \text{if } xy \geq 0. \end{aligned} \tag{5}$$

Lemma 3. (Cauchy inequality (Mitrinovic *et al.*, 1993a)) *For any $x > 0, y > 0 \in \mathbb{R}^n$, the following inequity holds:*

$$\left(\sum_{i=1}^{n} a_i b_i\right)^2 \leq \sum_{i=1}^{n} a_i^2 \sum_{i=1}^{n} b_i^2. \tag{6}$$

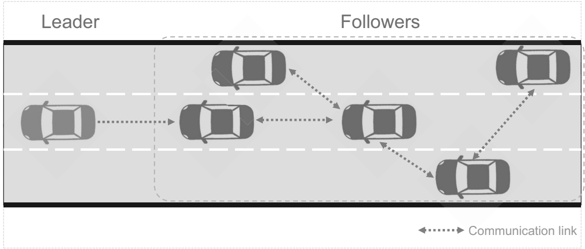

Fig. 1. Schematic diagram of the vehicle formation, where the leftmost vehicle is the leader, the vehicles in the dotted box are the followers, and they all communicate with each other through the V2V/V2I technology.

2.3. Problem formulation. For the above formation control problem, we can set one of the vehicles as the leader and the rest as the followers, as shown in Fig. 1. To achieve a specific graphical formation, a virtual leader state can be set for each follower by means of an offset as follows:

(i) given the position $x_o(t)$ and the velocity $v_o(t)$ of the leader, where

$$x_o(t) = [x_{o,1}(t), x_{o,2}(t)]^T, \quad v_o(t) = [v_{o,1}(t), v_{o,2}(t)]^T$$

are both expressed as vectors in a two-dimensional space,

(ii) set the virtual leader state for each follower:

$$\begin{cases} \bar{x}_o^{(i)}(t) = x_o(t) + o_x^{(i)}(t), \\ \bar{v}_o^{(i)}(t) = v_o(t), \end{cases} \tag{7}$$

where $o_x^{(i)}(t)$ represents the offset of follower vehicle i relative to the leader in a two-dimensional space, and it also represents that it is desirable that all the vehicles have the same velocity.

The state space modeling of the above system, due to the structure, attitude, wind resistance, system disturbance and other physical factors, involves a large number of unknown nonlinear items in mathematical dynamics:

$$\begin{cases} \dot{x}_i(t) = v_i(t), \\ \dot{v}_i(t) = f_i\big(x_i(t), v_i(t)\big) + g_i\big(x_i(t), v_i(t)\big)u_i(t), \end{cases} \tag{8}$$

where $x_i(t)$, $v_i(t)$, and $u_i(t)$ are the position, velocity state, and control input of the follower vehicles, respectively, while f_i and g_i are nonlinear functions that conform to a certain type of constraint.

Additionally, the dynamics formulation of the leader vehicle is described as follows:

$$\begin{cases} \dot{x}_o(t) = v_o(t), \\ \dot{v}_o(t) = f_o\big(x_o(t), v_o(t)\big), \end{cases} \tag{9}$$

where $x_o(t)$ and $v_o(t)$ denote the position and velocity of the leader, respectively, while f_o is an unknown smooth function. Each vehicle has two degrees of freedom in its state, which means that the vehicle can move in the directions of the X and Y axes.

Assumption 1. The nonlinear function f_o of the leader vehicle is bounded by a positive constant, that is, $\forall t \in \mathbb{R}^+$, $\| f_o \| \leq \alpha$.

Assumption 2. The nonlinear function g_i of follower i is either a negative-definite or a positive-definite symmetric matrix, and its eigenvalues $\lambda_1(g_i), \ldots, \lambda_m(g_i)$ satisfy

$$0 < \underline{g}_i \leq \| \lambda_1(g_i) \|, \ldots, \| \lambda_m(g_i) \| < \infty, \tag{10}$$

with $\underline{g}_i$ being a positive constant.

Let $x_i = [x_{i1}, x_{i2}]^T$ and $v_i = [v_{i1}, v_{i2}]^T$ be the position and velocity state in the X and Y directions, respectively. The position and velocity states of the leader vehicle are

$$\begin{aligned} x_o &= [x_{o1}, x_{o2}]^T, \\ v_o &= [v_{o1}, v_{o2}]^T. \end{aligned} \tag{11}$$

Further, define the tracking error as

$$\begin{aligned} \zeta_i^x(t) &= x_i(t) - x_o(t) - o_x^{(i)}(t), \\ \zeta_i^v(t) &= v_i(t) - v_o(t). \end{aligned} \tag{12}$$

Note that, in a formation period, the position offset between the leader and the followers is kept constant. Thus, taking the derivative of Eqn. (12) during a formation period, we have

$$\begin{aligned} \dot{\zeta}_i^x(t) =& \dot{x}_o(t) + \dot{o}_x^{(i)}(t) - \dot{x}_i(t) \\ =& \dot{x}_o(t) - \dot{x}_i(t) = \zeta_i^v(t), \\ \dot{\zeta}_i^v(t) =& \dot{v}_o(t) - \dot{v}_i(t) \\ =& - f_o(x_o(t), v_o(t)) + f_i(x_i(t), v_i(t)) \\ & + g_i(x_i(t), v_i(t))\, u_i(t), \quad i = 1, 2, \ldots, n. \end{aligned} \tag{13}$$

Rewrite Eqn. (13) in a compact form as

$$\begin{aligned} \dot{Z}(t) = & \left(\begin{pmatrix} 0 & -I_n \\ 0 & 0 \end{pmatrix} \otimes I_m \right) Z(t) \\ & + \begin{pmatrix} 0_{nm} \\ F_f \end{pmatrix} - \begin{pmatrix} 0_{nm} \\ F_o \end{pmatrix} + \begin{pmatrix} 0_{nm} \\ U \end{pmatrix}, \end{aligned} \tag{14}$$

where the tracking error compact vector

$$Z = \left[\zeta_1^{xT}, \zeta_2^{xT}, \ldots, \zeta_n^{xT}, \zeta_1^{vT}, \zeta_2^{vT}, \ldots, \zeta_n^{vT} \right]^T,$$

$$F_f = \left\{ f_1^T, f_2^T, \ldots, f_n^T \right\}^T,$$

$$F_o = \left\{ f_o^T, \ldots, f_o^T \right\}^T,$$

$$U = \left\{ (g_1 u_1)^T, (g_2 u_2)^T, \ldots, (g_n u_n)^T \right\}^T,$$

and $\otimes$ denotes the Kronecker product.

Similarly, Eqn. (12) can also be represented by the tracking error defined in Eqn. (12),

$$\begin{aligned} e_i^x(t) &= \sum_{j \in \mathcal{N}_i} \left(\zeta_i^x(t) - \zeta_j^x(t) \right) + \zeta_i^x(t) \\ &\quad - \sum_{j \in \mathcal{N}_i} \left(o_x^{(i)}(t) - o_x^{(j)}(t) \right), \\ e_i^v(t) &= \sum_{j \in \mathcal{N}_i} \left(\zeta_i^v(t) - \zeta_j^v(t) \right) + \zeta_i^v(t). \end{aligned} \tag{15}$$

DANNs are used to approximate the unknown nonlinearity $f_i(\cdot)$,

$$f_i(x_i(t), v_i(t)) = W_i^{*T}(t) S(z_i(t)) + \varepsilon_i(z_i(t)), \tag{16}$$

where $z_i(t) = [x_i(t), v_i(t)]^T$, $W_i^*(t)$ is the ideal DANN weight matrix at time step t and $\varepsilon_i(t)$ is the associated approximation error.

Design the DANN approximation-based adaptive control protocol

$$\begin{aligned} u_i(t) = & - \left(k_i + \frac{\beta_i}{\underline{g}_i} \widehat{W}_i^T(t) \left\| S_i(z_i(t)) \right\|^2 \right) \\ & \times \left(e_i^x(t) + e_i^v(t) \right), \quad i = 1, 2, \ldots, n, \end{aligned} \tag{17}$$

where k_i and β_i are positive design constant parameters to be given later, while $\widehat{W}_i(t)$ is the estimation of ideal weight $W_i^*(t)$.

3. Dynamic adjustment neural network

The conventional radial basis function neural network (RBFNN) is recognized for its high computational precision in learning and fitting time series data, while recurrent neural networks (RNNs) are not only capable of achieving high-accuracy data fitting, but also exhibit compatibility with new patterns in time series data. However, these models necessitate the use of an extensive reservoir to capture the characteristics of time series data, requiring substantial storage and computational resources. As a result, these models may not be suitable for situations with dynamic or environmentally-induced variations.

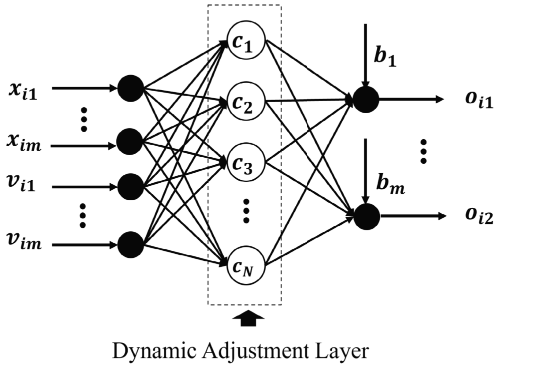

Fig. 2. Fundamental network structure for dynamic adjustment.

To remedy this limitation, we propose a streamlined efficacious neural network architecture termed the DANN. This architecture permits adaptive adjustments to both the connection weights and the network structure via online learning to maintain a high level of precision in approximating nonlinear functions. Rooted in the principles of the RBFNN, the DANN is equipped to introduce new hidden neurons as needed or to utilize the gradient descent method for fine-tuning the output weights and the centers of hidden neurons.

During the learning process, the DANN consistently evaluates the "novelty" of each new data sample. Should the novelty criterion of the sample be met, the network proceeds to integrate a fresh neuron into the hidden layer. The centre of this new neuron is set to match the value of the recently received sample. Furthermore, the output weights of the new neuron are determined based on the current network output error, and the center width is established as the minimum distance between the input value and the existing neuron centers. The DANN is also designed to dynamically prune superfluous hidden neurons to optimize its structure.

As delineated in Fig. 2, the DANN framework is composed of an input layer, an output layer, and an intermediary dynamic adjustment layer. The dynamic adjustment layer is particularly versatile, with the capability to update both the quantity and parameters of its neurons in real-time, adhering to the rule set elucidated in Fig. 3.

3.1. Neuron distance.

Given two probability distribution functions (PDFs), namely, $f_p(\tau)$ and $f_q(\tau)$, as shown in Fig. 4, the Kullback–Leibler information (KLI) $I_{KL}\left(f_p \parallel f_q\right)$ between them is calculated as

$$I_{KL}\left(f_p \parallel f_q\right) = \int_\tau f_p(\tau) \ln \frac{f_p(\tau)}{f_q(\tau)} \,\mathrm{d}\tau \tag{18}$$

Further, the Kullback–Leibler divergence (KLD) $D_{KL}\left(f_p, f_q\right)$ can be seen as a measurement to quantify the distance of $f_p(\tau)$ and $f_q(\tau)$ by

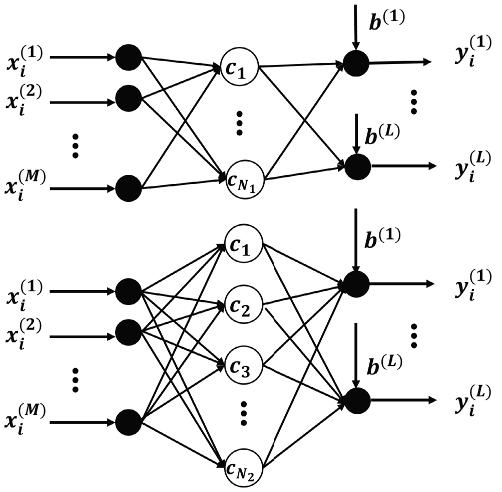

Fig. 3. Schematic diagram of structural changes before and after DANN dynamic adjustment.

$$D_{KL}\left(f_p, f_q\right) = I_{KL}\left(f_p \parallel f_q\right) + I_{KL}\left(f_q \parallel f_p\right). \tag{19}$$

Let

$$\theta_i(z) = \left\| \frac{z - c_i}{\sigma_i} \right\| = \frac{\left(x - c_i\right)^T \left(x - c_i\right)}{\sigma_i^T \sigma_i}, \tag{20}$$

where $c_i,\ \sigma_i$ are the unit center and spread constant of neuron i, and the i-th kernel of the hidden neurons is

$$\begin{aligned}\phi_i(z) &= \exp\left(-\sum_{j=1}^{2m} \frac{\left(z^{(j)} - c_i^{(j)}\right)^2}{\left(\sigma_i^{(j)}\right)^2}\right)\\ &= \exp\left(-\theta_i(z)\right).\end{aligned} \tag{21}$$

We design the "difference" between two hidden neurons

$$\begin{aligned}\text{dif} =& D_{KL}\left(f_p, f_q\right)\\ =& \int_\tau f_p(\tau) \ln \frac{f_p(\tau)}{f_q(\tau)} \,\mathrm{d}\tau + \int_\tau f_q(\tau) \ln \frac{f_q(\tau)}{f_p(\tau)} \,\mathrm{d}\tau\\ =& \int_\tau f_p(\tau) \left(\theta_i - \theta_j\right) \,\mathrm{d}\tau + \int_\tau f_q(\tau)(\theta_j - \theta_i) \,\mathrm{d}\tau\\ =& \left(\frac{c_i^T \mathbf{1}_{2m}}{2} + \frac{\sigma_i^T \sigma_i \left(c_i^T - 2c_j^T\right) \mathbf{1}_{2m}}{2\sigma_j^T \sigma_j}\right)\\ & \times \exp\left(-\frac{c_i^T c_i}{\sigma_i^T \sigma_i}\right)\\ & + \left(\frac{c_j^T \mathbf{1}_{2m}}{2} + \frac{\sigma_j^T \sigma_j \left(c_j^T - 2c_i^T\right) \mathbf{1}_{2m}}{2\sigma_i^T \sigma_i}\right)\\ & \times \exp\left(-\frac{c_j^T c_j}{\sigma_j^T \sigma_j}\right)\end{aligned}$$

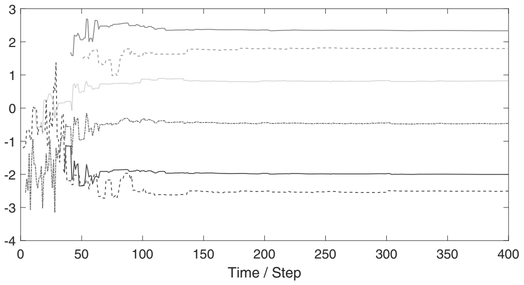

Fig. 4. Distance between Gaussian functions $f_q(\tau)$ and $f_p(\tau)$ with the same kernel center c.

$$+m\sqrt{\pi}\sigma_i^T\mathbf{1}_{2m}\left(\frac{2\left(\mathbf{1}_{2m}^T-c_j^T\right)c_i}{\sigma_j^T\sigma_j}\right.$$
$$\left.+\frac{2\left(\mathbf{1}_{2m}^T-c_i^T\right)c_j}{\sigma_i^T\sigma_i}-1\right). \quad (22)$$

3.2. Growing of hidden neurons. Calculate the "distance" dist between the neural network input z and a hidden neuron as

$$\text{dist}=1-\exp\left(-\left\|\frac{z-c}{\sigma}\right\|\right), \quad (23)$$

where c, σ are respectively the unit center and spread constant of the hidden neuron.

Denote by $\text{dist}_{\min}=\min(\text{dist})$ the minimum value of dist among z and all the hidden neurons, by $\text{dif}_{\min}=\min(\text{dif})$ the minimum value of 'dif' among all the pairs of two hidden neurons. If $\text{dist}_{\min}\geq D_{\text{th}}^a$ and $\text{dif}_{\min}\geq D_{\text{th}}^n$, with D_{th}^a and D_{th}^n being an online adaptive thresholds for generating new neurons in the hidden layer of neural network, then a new hidden neuron will be initialized and added, while the connection weights of the neural network will also be updated. This procedure works under the following rules:

(i) Update the number of hidden neurons:

$$N\leftarrow N+1. \quad (24)$$

(ii) Set the central and central width of the newly added hidden neuron:

$$c_N=z, \quad (25)$$

$$\sigma_N=\lambda\cdot\text{dist}_{\min}. \quad (26)$$

(iii) Initialize the connection weights from the new neuron to output layer:

$$W_{\text{out}}^N=\frac{ae}{b\mathbf{1}_m+e}. \quad (27)$$

Here, $W_{\text{out}}^N\in\mathbb{R}^m$, $e\in\mathbb{R}^m$ is the network output error-related residual vector, a and b are constant scalars, and $\mathbf{1}_m\in\mathbb{R}^m$ denotes the m-dimensional vector of ones.

3.3. Pruning of hidden neurons. Suppose that a DANN with h hidden neurons is capable to approximate the target function under the accuracy requirement; then, if the number of hidden layer nodes exceeds h, how to prune the redundant hidden neurons? Without loss of generality, we consider the case of a DANN with $h+1$ hidden neurons where only one can be pruned as redundant.

We consider deleting the hidden layer node $h+1$ that contributes less to the network output for a long time, and calculate the output of each hidden layer node as

$$o_{i,k}=w_i\exp\left(-\frac{\|z_k-c_i\|^2}{\sigma_i^2}\right),$$
$$i=1,2,\dots,m. \quad (28)$$

Subsequently, we identify the maximal absolute value, denoted as $\|o_{\max,k}\|$, across the outputs of the hidden layer nodes. We then define the normalized contribution, $r_{i,k}$, of each hidden layer node's output to the overall network output by calculating the ratio of $\|o_{i,k}\|$ to $\|o_{\max,k}\|$ as follows:

$$r_{i,k}=\left\|\frac{o_{i,k}}{o_{\max,k}}\right\|. \quad (29)$$

If the consecutive P steps of $r_{i,k}$ are less than the set threshold δ, the corresponding hidden layer node is deleted, and then the learning of the next sample data pair will be carried out.

3.4. Updating network weights. If it is unnecessary to add or prune any hidden neuron, then what needs to be done is update the output weights according to network output related error and the hidden activation along the following rules:

$$W_{\text{out}}^i\leftarrow W_{\text{out}}^i+\lambda\cdot\text{Proj}\left(W_{\text{out}}^i,\ \theta(z)ePB\right), \quad (30)$$

where λ is a positive constant, B is the gradient vector of the network output with respect to W_{out}^i, P is the covariance matrix, and $\text{Proj}(\cdot)$ is a projection function defined as

$$\text{Proj}(a,b)=\frac{b\cdot a}{|a|}. \quad (31)$$

Further, e denotes the system error which represents a specific term in a task. We design the update law of output weights as

$$W_{\text{out}}^N\leftarrow W_{\text{out}}^N+\lambda\cdot\frac{ae}{b\mathbf{1}_m+e}\theta(z),$$
$$c_i\leftarrow c_i+\lambda\cdot\Delta c_i,\quad i=1,2,\dots,N, \quad (32)$$

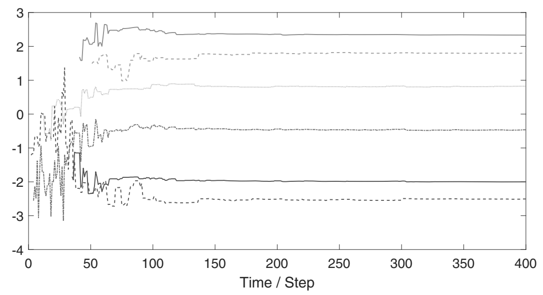

Fig. 5. Connection weights variation of the output layer over time.

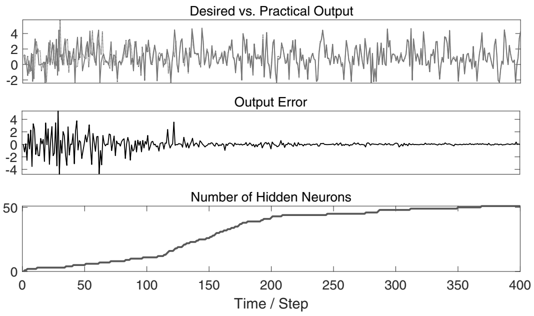

Fig. 6. Comparison between the practical output and desired values (top), the fitting error (center), and the number of hidden neurons over time (bottom).

where

$$\Delta c_i = \frac{2\,(z-c_i)\,S(z){W_{\text{out}}^{i}}^{T} e}{\sigma_i^2}.$$

In Fig. 5, we illustrate the variations in the connection weights of the output layer over time. Figure 6 depicts the fitting error between the actual outputs and the desired targets. It is evident that the DANN actively fine-tunes both the quantity of neurons and the synaptic weights within the dynamic adjustment layer in response to changing conditions. As time progresses, the fitting error consistently diminishes, reflective of the ongoing adjustment in the number of nodes within the dynamic adjustment layer, conforming to the previously outlined updating protocols.

Furthermore, Fig. 6 underscores that, by incrementally increasing the number of nodes to an optimal count, the network achieves enhanced expansiveness and significantly improved fitting capabilities. This expansion is a strategic maneuver, governed by the network's intrinsic mechanisms, to refine its predictive accuracy and adaptability.

4. Main results

Consider the Lyapunov candidate function as

$$\begin{aligned} V(t) = {} & \frac{1}{2} Z^T(t)\left(\begin{pmatrix} 2\widetilde{L} & \widetilde{L} \\ \widetilde{L} & \widetilde{L} \end{pmatrix} \otimes I_m\right) Z^T(t) \\ & + \frac{1}{2}\sum_{i=1}^{n} \gamma_i^{-1} \widetilde{W}_i^2(t), \end{aligned} \tag{33}$$

where $\widetilde{L} = L + B, B = \operatorname{diag}\{b_1, b_2, \dots, b_n\}$, $\widetilde{W}_i(t) = \widehat{W}_i(t) - W_i^*(t)$.

Taking the derivative of $V(t)$ with respect to time t, we have

$$\begin{aligned} \dot{V}(t) = {} & -\frac{1}{2} Z^T(t)\Big(\Big(\begin{pmatrix} 0 & -I_n \\ 0 & 0 \end{pmatrix}^T \begin{pmatrix} 2\widetilde{L} & \widetilde{L} \\ \widetilde{L} & \widetilde{L} \end{pmatrix} \\ & + \begin{pmatrix} 2\widetilde{L} & \widetilde{L} \\ \widetilde{L} & \widetilde{L} \end{pmatrix}\begin{pmatrix} 0 & -I_n \\ 0 & 0 \end{pmatrix}\Big) \otimes I_m\Big) Z(t) \\ & + Z^T(t)\left(\begin{pmatrix} 2\widetilde{L} & \widetilde{L} \\ \widetilde{L} & \widetilde{L} \end{pmatrix} \otimes I_m\right)\begin{pmatrix} 0_{nm} \\ F_f \end{pmatrix} \\ & - Z^T(t)\left(\begin{pmatrix} 2\widetilde{L} & \widetilde{L} \\ \widetilde{L} & \widetilde{L} \end{pmatrix} \otimes I_m\right)\begin{pmatrix} 0_{nm} \\ F_o \end{pmatrix} \\ & + Z^T(t)\left(\begin{pmatrix} 2\widetilde{L} & \widetilde{L} \\ \widetilde{L} & \widetilde{L} \end{pmatrix} \otimes I_m\right)\begin{pmatrix} 0_{nm} \\ U \end{pmatrix} \\ & + \sum_{i=1}^{n} \gamma_i^{-1} \widetilde{W}_i(t) \dot{\widehat{W}}_i(t) \qquad (34) \\ = {} & -Z^T(t)\left(\begin{pmatrix} 0 & -\widetilde{L} \\ -\widetilde{L} & -\widetilde{L} \end{pmatrix} \otimes I_m\right) Z(t) \\ & + \sum_{i}^{n} \left(e_i^x(t) + e_i^v(t)\right)^T f_i\left(x_i(t), v_i(t)\right) \\ & - \sum_{i}^{n} \left(e_i^x(t) + e_i^v(t)\right)^T f_o\left(x_i(t), v_i(t)\right) \\ & + \sum_{i}^{n} \left(e_i^x(t) + e_i^v(t)\right)^T g_i\left(x_i(t), v_i(t)\right) u_i(t) \\ & + \sum_{i=1}^{n} \gamma_i^{-1} \widetilde{W}_i(t) \dot{\widehat{W}}_i(t). \end{aligned}$$

Substituting the follower's unknown nonlinearity by the DANN approximation, Eqn. (34) becomes

$$\dot{V}(t) = -\,Z^T(t)\left(\begin{pmatrix} 0 & -\widetilde{L} \\ -\widetilde{L} & -\widetilde{L} \end{pmatrix} \otimes I_m\right) Z(t)$$

$$
\begin{aligned}
&+\sum_{i}^{n}\left(e_i^x(t)+e_i^v(t)\right)^T\ \Big(\widehat{W}_i^T(t)S_i\left(z_i(t)\right)\\
&+\varepsilon_i\left(z_i(t)\right)\big)\\
&-\sum_{i}^{n}\left(e_i^x(t)+e_i^v(t)\right)^T f_o\left(x_i(t),v_i(t)\right)\\
&+\sum_{i}^{n}\left(e_i^x(t)+e_i^v(t)\right)^T g_i\left(x_i(t),v_i(t)\right)u_i(t)\\
&+\sum_{i=1}^{n}\gamma_i^{-1}\widetilde{W}_i(t)\dot{\widehat{W}}_i(t).
\end{aligned}
\tag{35}
$$

By Lemmas 2 (Young's inequality) and 3 (Cauchy inequality), we can obtain

$$
\begin{aligned}
&\left(e_i^x(t)+e_i^v(t)\right)^T\widehat{W}_i^T(t)S_i\left(z_i(t)\right)\\
&\qquad\le\beta_i\Big(\left(e_i^x(t)+e_i^v(t)\right)^T\widehat{W}_i^T(t)S_i\left(z_i(t)\right)\Big)^2\\
&\qquad\quad+\frac{1}{4\beta_i}\\
&\qquad\le\beta_i\widehat{W}_i^T(t)\left\|S_i\left(z_i(t)\right)\right\|^2\left\|e_i^x(t)+e_i^v(t)\right\|^2\\
&\qquad\quad+\frac{1}{4\beta_i}\\
&\qquad\quad-\left(e_i^x(t)+e_i^v(t)\right)^T f_o\left(x_i(t),v_i(t)\right)\\
&\qquad\le\lambda_i\left\|e_i^x(t)+e_i^v(t)\right\|^2\\
&\qquad\quad+\frac{\overline{f}_o^{\,2}}{4\lambda_i}\left(e_i^x(t)+e_i^v(t)\right)^T\varepsilon_i\left(z_i(t)\right)\\
&\qquad\le\delta_i\left\|e_i^x(t)+e_i^v(t)\right\|^2+\frac{\overline{\varepsilon}_i^2}{4\delta_i},
\end{aligned}
\tag{36}
$$

where β_i, λ_i and δ_i are positive constants.

By using Eqn. (36), we have

$$
\begin{aligned}
&\dot{V}(t)\\
&\le-Z^T(t)\left(\begin{pmatrix}0&-\widetilde{L}\\-\widetilde{L}&-\widetilde{L}\end{pmatrix}\otimes I_m\right)Z(t)\\
&\quad+\sum_{i}^{n}\beta_i\widehat{W}_i^T(t)\left\|S_i\left(z_i(t)\right)\right\|^2\left\|e_i^x(t)+e_i^v(t)\right\|^2\\
&\quad+\sum_{i}^{n}\left(\lambda_i+\delta_i\right)\left\|e_i^x(t)+e_i^v(t)\right\|^2\\
&\quad+\sum_{i}^{n}\left(e_i^x(t)+e_i^v(t)\right)^T g_i\left(x_i(t),v_i(t)\right)u_i(t)\\
&\quad-\sum_{i=1}^{n}\gamma_i^{-1}\widetilde{W}_i(t)\widehat{W}_i(t)\\
&\quad+\sum_{i}^{n}\left(\frac{1}{4\beta_i}+\frac{\overline{f}_o^{\,2}}{4\lambda_i}+\frac{\overline{\varepsilon}_i^2}{4\delta_i}\right).
\end{aligned}
\tag{37}
$$

By substituting the control protocol (13) and the adaptive weight tuning method into Eqn. (37), we obtain

$$
\begin{aligned}
\dot{V}(t)\le&-Z^T(t)\left(\begin{pmatrix}0&-\widetilde{L}\\-\widetilde{L}&-\widetilde{L}\end{pmatrix}\otimes I_m\right)Z(t)\\
&+\sum_{i}^{n}\beta_i\widehat{W}_i^T(t)\left\|S_i\left(z_i(t)\right)\right\|^2\left\|e_i^x(t)+e_i^v(t)\right\|^2\\
&+\sum_{i}^{n}\left(\lambda_i+\delta_i\right)\left\|e_i^x(t)+e_i^v(t)\right\|^2\\
&-\sum_{i}^{n}\left(e_i^x(t)+e_i^v(t)\right)^T g_i\left(x_i(t),v_i(t)\right)\Big(k_i\\
&-\frac{\beta_i}{\underline{g}_i}\widehat{W}_i^T(t)\parallel S_i\left(z_i(t)\right)\parallel^2\Big)\left(e_i^x(t)+e_i^v(t)\right)\\
&-\sum_{i=1}^{n}\gamma_i^{-1}\widetilde{W}_i^T(t)\widehat{W}_i(t)\\
&+\sum_{i}^{n}\left(\frac{1}{4\beta_i}+\frac{\overline{f}_o^{\,2}}{4\lambda_i}+\frac{\overline{\varepsilon}_i^2}{4\delta_i}\right).
\end{aligned}
\tag{38}
$$

Based on Assumption 2, the above inequality can be rewritten as

$$
\begin{aligned}
\dot{V}(t)\le&-Z^T(t)\left(\begin{pmatrix}0&-\widetilde{L}\\-\widetilde{L}&-\widetilde{L}\end{pmatrix}\otimes I_m\right)Z(t)\\
&+\sum_{i}^{n}\beta_i\widehat{W}_i^T(t)\left\|S_i\left(z_i\right)\right\|^2\left\|e_i^x(t)+e_i^v(t)\right\|^2\\
&+\sum_{i}^{n}\left(\lambda_i+\delta_i\right)\left\|e_i^x(t)+e_i^v(t)\right\|^2\\
&-\sum_{i}^{n}\Big(\underline{g}_ik_i-\beta_i\,\widehat{W}_i^T(t)\parallel S_i\left(z_i(t)\right)\parallel^2\Big)\\
&\times\left\|e_i^x(t)+e_i^v(t)\right\|^2\\
&-\sum_{i=1}^{n}\gamma_i^{-1}\widetilde{W}_i^T(t)\widehat{W}_i(t)+\sum_{i}^{n}\left(\frac{1}{4\beta_i}+\frac{\overline{f}_o^{\,2}}{4\lambda_i}+\frac{\overline{\varepsilon}_i^2}{4\delta_i}\right)\\
=&-Z^T(t)\left(\begin{pmatrix}0&-\widetilde{L}\\-\widetilde{L}&-\widetilde{L}\end{pmatrix}\otimes I_m\right)Z(t)\\
&+\sum_{i}^{n}\left(\lambda_i+\delta_i\right)\left\|e_i^x(t)+e_i^v(t)\right\|^2\\
&-\sum_{i}^{n}\underline{g}_ik_i\left\|e_i^x(t)+e_i^v(t)\right\|^2\\
&-\sum_{i=1}^{n}\gamma_i^{-1}\widetilde{W}_i^T(t)\widehat{W}_i(t)\\
&+\sum_{i}^{n}\left(\frac{1}{4\beta_i}+\frac{\overline{f}_o^{\,2}}{4\lambda_i}+\frac{\overline{\varepsilon}_i^2}{4\delta_i}\right).
\end{aligned}
\tag{39}
$$

Let $\underline{g}_i k_i \geq \lambda_i + \delta_i + k_0$. Then Eqn. (39) becomes

$$\begin{aligned}
\dot{V}(t) \leq & -Z^T(t) \left(\begin{pmatrix} 0 & -\widetilde{L} \\ -\widetilde{L} & -\widetilde{L} \end{pmatrix} \otimes I_m \right) Z(t) \\
& - \sum_{i}^{n} k_0 \left\| e_i^x(t) + e_i^v(t) \right\|^2 \\
& - \sum_{i=1}^{n} \gamma_i^{-1} \widetilde{W}_i^T(t) \widehat{W}_i(t) \\
& + \sum_{i}^{n} \left(\frac{1}{4\beta_i} + \frac{\overline{f}_o^{\,2}}{4\lambda_i} + \frac{\overline{\varepsilon}_i^2}{4\delta_i} \right) \\
\leq & -Z^T(t)\,(k_0) \begin{pmatrix} \widetilde{L} & 0 \\ 0 & \widetilde{L} \end{pmatrix}^T \begin{pmatrix} \widetilde{L} & 0 \\ 0 & \widetilde{L} \end{pmatrix} \\
& - \begin{pmatrix} 0 & \widetilde{L} \\ \widetilde{L} & \widetilde{L} \end{pmatrix} \quad \otimes I_m)\, Z(t) \\
& - \sum_{i=1}^{n} \gamma_i^{-1} \widetilde{W}_i^T(t) \dot{\widehat{W}}_i(t) \\
& + \sum_{i}^{n} \left(\frac{1}{4\beta_i} + \frac{\overline{f}_o^{\,2}}{4\lambda_i} + \frac{\overline{\varepsilon}_i^2}{4\delta_i} \right). \qquad (40)
\end{aligned}$$

Due to the fact that we designed $\widetilde{W}_i^T(t)\widehat{W}_i(t) = \frac{1}{2}\left(\widetilde{W}_i^2(t) + \widehat{W}_i^2(t) - W_i^{*2}(t)\right)$, we have

$$\begin{aligned}
& -\sum_{i=1}^{n} \gamma_i^{-1} \widetilde{W}_i^T(t) \widehat{W}_i(t) \\
& \qquad \leq \frac{1}{2}\gamma_i^{-1} \sum_{i=1}^{n} \left(-\widetilde{W}_i^2(t) + W_i^{*2}(t) \right). \qquad (41)
\end{aligned}$$

Substituting Eqn. (41) into (40), it becomes

$$\begin{aligned}
& \dot{V}(t) \\
& \leq -Z^T(t) \left(k_0 \begin{pmatrix} \widetilde{L} & 0 \\ 0 & \widetilde{L} \end{pmatrix}^T \begin{pmatrix} \widetilde{L} & 0 \\ 0 & \widetilde{L} \end{pmatrix} \right. \\
& \left. - \begin{pmatrix} 0 & \widetilde{L} \\ \widetilde{L} & \widetilde{L} \end{pmatrix} \right) \otimes I_m Z(t) - \frac{1}{2} \sum_{i=1}^{n} \gamma_i^{-1} \widetilde{W}_i^2(t) \\
& + \sum_{i}^{n} \left(\frac{1}{4\beta_i} + \frac{\overline{f}_o^{\,2}}{4\lambda_i} + \frac{\overline{\varepsilon}_i^2}{4\delta_i} + \frac{1}{2}\gamma_i^{-1} W_i^{*2}(t) \right). \qquad (42)
\end{aligned}$$

Define

$$\Lambda = \begin{pmatrix} \widetilde{L} & 0 \\ 0 & \widetilde{L} \end{pmatrix}^T \begin{pmatrix} \widetilde{L} & 0 \\ 0 & \widetilde{L} \end{pmatrix},$$

$$\Delta = \begin{pmatrix} 0 & \widetilde{L} \\ \widetilde{L} & \widetilde{L} \end{pmatrix}, \quad \Omega = \begin{pmatrix} \widetilde{2L} & \widetilde{L} \\ \widetilde{L} & \widetilde{L} \end{pmatrix}.$$

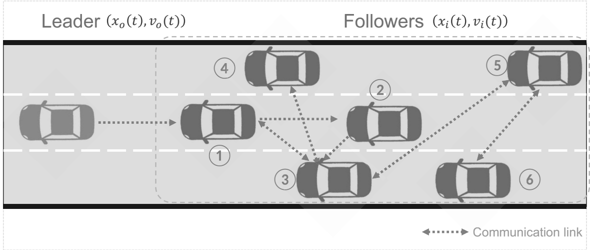

Fig. 7. Vehicle platoon in a simulation setting; the dotted arrow shows the communication direction between vehicles.

Taking

$$k_0 > \lambda_{\min}^{-1}(\Lambda) \left(\lambda_{\max}(\Delta) + \omega/2 \cdot \lambda_{\max}(\Omega) \right),$$

where $\lambda_{\min}(\Lambda)$, $\lambda_{\max}(\Delta)$ and $\lambda_{\max}(\Omega)$ denote the smallest eigenvalue of matrix Λ, the largest eigenvalue of matrix Δ and the largest eigenvalue of matrix Ω, respectively, $\omega = \min\{\sigma_1\gamma_1, \sigma_2\gamma_2, \dots, \sigma_n\gamma_n\}$; thus, we can rewrite Eqn. (42) as

$$\begin{aligned}
\dot{V}(t) \leq & -\frac{\omega}{2} Z^T(t) \left(\Omega \otimes I_m \right) Z(t) \\
& - \frac{1}{2} \sum_{i=1}^{n} \gamma_i^{-1} \widetilde{W}_i^2(t) \\
& + \sum_{i}^{n} \left(\frac{1}{4\beta_i} + \frac{\overline{f}_o^{\,2}}{4\lambda_i} + \frac{\overline{\varepsilon}_i^2}{4\delta_i} + \frac{1}{2}\gamma_i^{-1} W_i^{*2}(t) \right) \\
= & -\omega V(t) \\
& + \sum_{i}^{n} \left(\frac{1}{4\beta_i} + \frac{\overline{f}_o^{\,2}}{4\lambda_i} + \frac{\overline{\varepsilon}_i^2}{4\delta_i} + \frac{1}{2}\gamma_i^{-1} W_i^{*2}(t) \right).
\end{aligned}$$
(43)

Then we have

$$V(t) \leq V(0)e^{-\omega t} + \frac{\rho}{\omega}\left(1 - e^{-\omega t}\right), \qquad (44)$$

where

$$\rho = \sum_{i}^{n} \left(\frac{1}{4\beta_i} + \frac{\overline{f}_o^{\,2}}{4\lambda_i} + \frac{\overline{\varepsilon}_i^2}{4\delta_i} + \frac{1}{2}\gamma_i^{-1} W_i^{*2}(t) \right). \qquad (45)$$

It should be noted that if we chose appropriate values of parameters β_i, λ_i and δ_i, we could obtain the positive constant ρ. By Lemma 1, the tracking error of each follower agent can be decreased to a small neighborhood of zero, which implies that the leader-following vehicle platoon consensus can be achieved.

5. Simulation results

Design the leader-following vehicle platoon topology as shown in Fig. 7. Set

$$v_o = \left[6\sin(t),\ 1 + \sin(2t+1), 2\cos\left(0.5t^2 + 1\right) \right]^T,$$

$$x_o(0) = [10, 15, 10]^T.$$

The initial position of follower vehicle i is a random vector in the range of $[-10,\ 10]$.

The connection matrix of followers can be constructed by

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

and the corresponding Laplacian matrix is

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

On the other hand, the connection vector from the leader to the followers is

$$C_l = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

The leader's dynamics is given by

$$\dot{x}_o(t) = v_o(t)$$

and

$$\begin{aligned} &\dot{v}_o(t) \\ &= \begin{pmatrix} -\sin(x_{o1}(t)) + 0.1\left(1 - x_{o1}^2(t)\right)\cos(v_{o1}(t)) \\ -\cos(x_{o2}(t)) - 0.3\left(1 + x_{o2}^2(t)\right)\sin(v_{o2}(t)) \end{pmatrix}, \end{aligned}$$

while the followers' dynamics are given by

$$\begin{aligned} \dot{x}_i(t) =& v_i(t), \\ \dot{v}_i(t) =& \begin{pmatrix} -a_{i1}\sin(x_{i1}) + b_{i1}(1 - x_{i1})\cos(c_{i1}v_{l1}) \\ a_{i2}\cos(x_{i2}) + b_{i2}(1 + c_{i2}x_{i2})\sin(v_{i2}) \end{pmatrix} \\ &+ \begin{pmatrix} \beta_{i1}\cos(x_{i1})\,v_{i1} & 0 \\ 0 & \sin^2(x_{i2}v_{i2}) + \beta_{i2} \\ 0 & 0 \end{pmatrix} \\ &\times \begin{pmatrix} u_{i1} \\ u_{i2} \end{pmatrix}, \quad i = 1, \dots, 6, \end{aligned}$$

where

$$a = \begin{bmatrix} 1 & 2 \\ 0.1 & 0.5 \\ 0.5 & 1 \\ -1 & 0.4 \\ 0.9 & -0.1 \\ 1.2 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1.2 & 1.2 \\ 1.1 & -0.5 \\ 0.5 & 0.1 \\ 0.9 & 2.4 \\ -0.2 & 1.1 \\ 0.2 & 1.6 \end{bmatrix}$$

Table 1. Performance comparison with other classical models.

Model	Convergence time	Steady-state error
RBFNN	632	4.2×10^{-2}
BPNN	603	2.5×10^{-2}
Fuzzy Set	744	5.8×10^{-2}
DANN	**428**	**7.8** $\times 10^{-3}$

$$c = \begin{bmatrix} 0.3 & 0.1 \\ 1.2 & 0.2 \\ -0.9 & 1.8 \\ 2.1 & 2.2 \\ 1.1 & 0.5 \\ 1.2 & 2 \end{bmatrix}, \quad \beta = \begin{bmatrix} 1 & 0.8 \\ -0.2 & 1.6 \\ 1.3 & 1 \\ 2.0 & 2.9 \\ -0.1 & 0.8 \\ 1.8 & 1.1 \end{bmatrix}.$$

Additionally, we also constructed nonlinear term fitting methods based on the radial basis function neural network (RBFNN), the backpropagation neural network (BPNN), and a fuzzy set model. Both the RBFNN and BPNN have a three-layer structure with 30 nodes in the hidden layer, and the dimension of the fuzzy set is 20. We primarily compared the number of convergence time steps and the steady-state error of the system. The results, as shown in Table 1, indicate that the proposed DANN achieves the best control performance.

It can be seen from the change diagram of the system state that, for both the position (see Fig. 8) and velocity (see Fig. 9), the changes are relatively intense at the middle time moment, and the output fluctuation of the corresponding DANN is also relative large (see Fig. 10), while the system output error is large compared with other time instants. Figures 11 to 14 show the internal variation and structure values of the DANN during system running. As observed while by combining Figs. 11 and 12, there is a notable increase in the number of hidden layer neurons within the DANN structure when significant fluctuations arise in the fitting error. This increment is designed to enhance the network's fitting capabilities. In Figs. 13 and 14, prior to approximately 300 time steps, the DANN primarily adjusts the output weights and the centers of hidden layer neurons to approximate the desired output. After about the 300-time-step threshold, the network begins to expand the number of hidden layer neurons, mainly improving the performance by augmenting the scale of the network. Thus, the DANN begins to significantly increase the number of hidden neurons at this point in order to better fit the characteristics of the new input samples. In addition, the spread constants of the hidden layer neurons are all at a relatively small level, so that the hidden neurons are more sensitive to the input and can respond significantly to changes in the input. On the other hand, the unit center distribution of hidden neurons is also relatively scattered (see Fig. 15), which helps the neural network to better respond to different inputs.

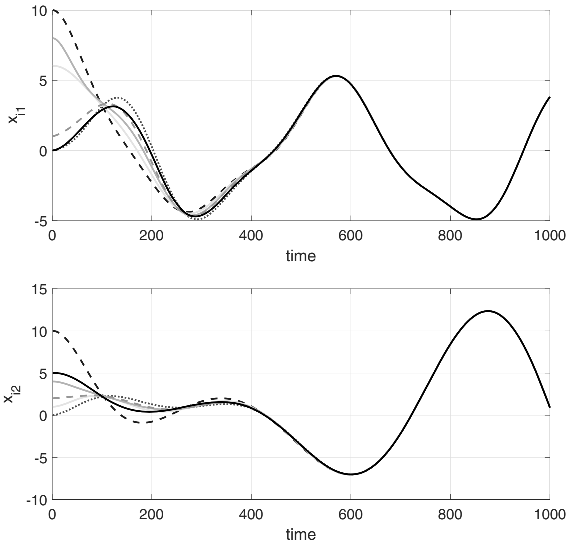

Fig. 8. Tracking consensus performance of position states x_{i1} and x_{i2}, $i = 1, 2, \ldots, 6$, where x_{i1} and x_{i2} denote the velocities along the directions of the X and Y axes, respectively.

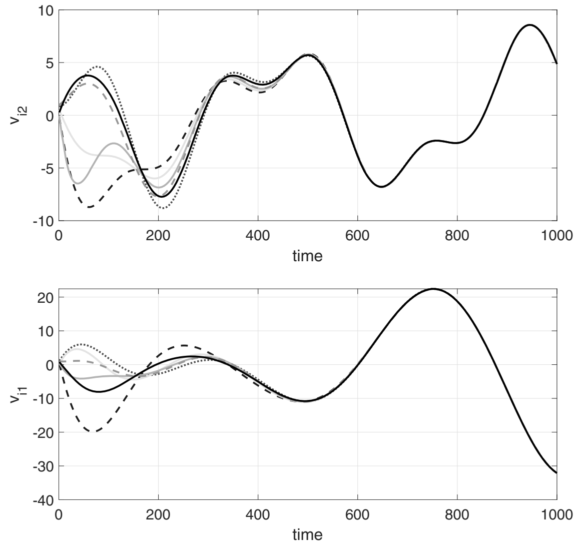

Fig. 9. Tracking consensus performance of velocity states v_{i1} and v_{i2}, $i = 1, 2, \ldots, 6$, where v_{i1}, v_{i2} denote the velocities along the directions of the X and Y axes, respectively.

6. Conclusion

In this paper, we introduced a mathematical model addressing the vehicle cooperative control issue and subsequently derived an adaptive control protocol that leverages the DANN. Our approach commenced with a meticulous mathematical analysis and verification of system control stability, leading to the identification of specific conditions required for certain system parameters. Numerical simulations not only depicted the evolution of the state consensus among the agents, but also delved into the internal dynamics of the DANN. It was observed that the DANN is not merely capable of dynamically altering its architecture in response to operational demands, but it also possesses the flexibility to refine its parameters for precise fitting of nonlinear expressions, effectively enhancing the network's inherent adaptability and precision in dynamic settings.

Looking ahead, future work could expand upon this foundation by exploring the integration of the DANN with more complex control systems, assessing the network's performance in real-world scenarios with varying degrees of unpredictability, and further refining the network's architecture for optimized adaptability and computational efficiency. The exploration of distributed DANN structures to emphasize enhanced scalability and reduced computational loads in multi-agent systems indicates a promising direction for future research.

Acknowledgment

This work is supported by the Anhui Provincial Key Scientific Research Projects of Universities under the grant 2022AH052231, the Anhui Provincial Major Scientific Research Projects of Universities under the grant 2022AH040300, the Natural Science Research Funding Project of Hefei Technology College under the grant 2022Akjcx06, the Quality Engineering Project of the Anhui Provincial Education Department under the grant 2020kcszjxtd67, and the Anhui Province Scientific Research Planning Project under the grant 2023AH052544.

References

Chang, B.-J., Hwang, R.-H., Tsai, Y.-L., Yu, B.-H. and Liang, Y.-H. (2019). Cooperative adaptive driving for platooning autonomous self driving based on edge computing, *International Journal of Applied Mathematics and Computer Science* **29**(2): 213–225, DOI: 10.2478/amcs-2019-0016.

Corets, J., Martinez, S., Karatas, T. and Bullo, F. (2002). Coverage control for mobile sensing networks, *International Conference on Robotics and Automation, Washington, USA*, pp. 1327–1332.

Dutta, R.G., Hu, Y., Yu, F., Zhang, T. and Jin, Y. (2022). Design and analysis of secure distributed estimator for vehicular platooning in adversarial environment, *IEEE Transactions on Intelligent Transportation Systems* **23**(4): 3418–3429.

Earnhardta, C., Groelke, B., Borek, J., Evan, Pelletier, Brennan, S. and Vermillion, C. (2022). Cooperative exchange-based platooning using predicted fuel-optimal

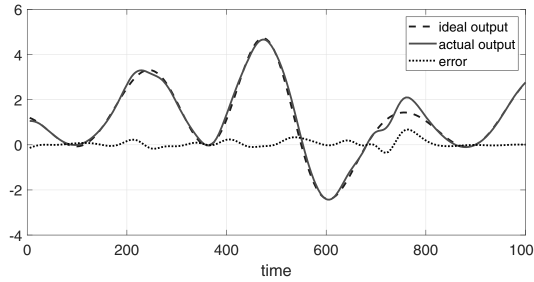

Fig. 10. Output error of the DANN (it should be noted that the ideal output is only estimated during our simulation experiment).

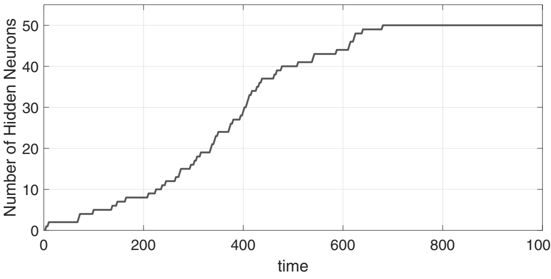

Fig. 11. Number of hidden neurons of the DANN.

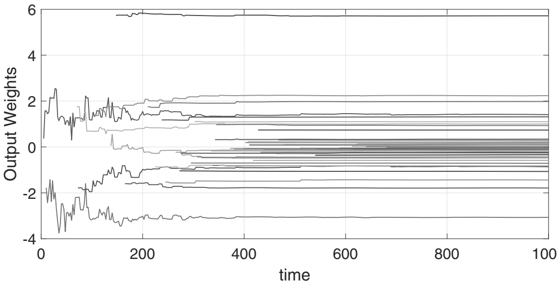

Fig. 12. Self-tuning procedure of the output weights of the DANN.

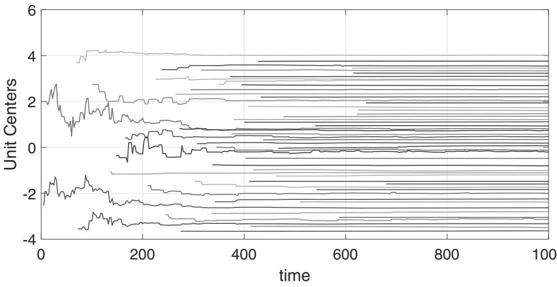

Fig. 13. Self-tuning procedure of unit centers of hidden neurons of the DANN.

operation of heavy-duty vehicles, *IEEE Transactions on Intelligent Transportation Systems* **23**(10): 17312–17324.

Feng, G., Dang, D. and He, Y. (2022). Robust coordinated control of nonlinear heterogeneous platoon interacted by uncertain topology, *IEEE Transactions on Intelligent Transportation Systems* **23**(6): 4982–4992.

Gao, F., Hu, X., Li, S.E., Li, K. and Sun, Q. (2018). Distributed adaptive sliding mode control of vehicular platoon with uncertain interaction topology, *IEEE Transactions on Industrial Electronics* **65**(8): 6352–6361.

Gao, F., Li, S. E., Zheng, Y. and Kum, D. (2016). Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay, *IET Intelligent Transport Systems* **10**(7): 503–513.

Gong, S., Zheng, M., Hu, J. and Zhang, A. (2023). Event-triggered cooperative control for high-order nonlinear multi-agent systems with finite-time consensus, *International Journal of Applied Mathematics and Computer Science* **33**(3): 439–448, DOI: 10.34768/amcs-2023-0032.

Hu, B.-B., Zhang, H.-T., Yao, W., Ding, J. and Cao, M. (2023). Spontaneous-ordering platoon control for multirobot path navigation using guiding vector fields, *IEEE Transactions on Robotics* **39**(4): 2654–2668.

Hu, M., Li, C., Bian, Y., Zhang, H., Qin, Z. and Xu, B. (2022). Fuel economy-oriented vehicle platoon control using economic model predictive control, *IEEE Transactions on Intelligent Transportation Systems* **23**(11).

Huang, J., Chen, J., Yang, H. and Li, D. (2023). Vehicle platoon tracking control based on adaptive neural network algorithm, *International Journal of Control, Automation and Systems* **21**(10): 3405–3418.

Huang, Z., Chu, D., Wu, C. and He, Y. (2019). Path planning and cooperative control for automated vehicle platoon using hybrid automata, *IEEE Transactions on Intelligent Transportation Systems* **20**(3): 959–974.

Li, J., Zhang, A. and Peng, C. (2022). Neuro-adaptive cooperative control for a class of high-order nonlinear multi-agent systems, *Measurement and Control* **56**(5–6): 928–937.

Li, K., Li, S.E., Gao, F., Lin, Z., Li, J. and Sun, Q. (2020a). Robust distributed consensus control of uncertain multiagents interacted by eigenvalue-bounded topologies, *IEEE Internet of Things Journal* **7**(5): 3790–3798.

Li, M., Cao, Z. and Li, Z. (2021). A reinforcement learning-based vehicle platoon control strategy for reducing energy consumption in traffic oscillations, *IEEE Transactions on Neural Networks and Learning Systems* **32**(12): 5309–5322.

Li, S. E., Zheng, Y., Li, K. and Wang, J. (2015). An overview of vehicular platoon control under the four-component framework, *IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea*, pp. 286–291.

Li, Y., Chen, W., Peeta, S. and Wang, Y. (2020b). Platoon control of connected multi-vehicle systems

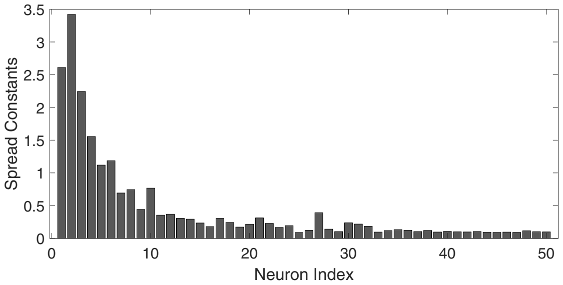

Fig. 14. Spread constants of hidden neurons of the DANN.

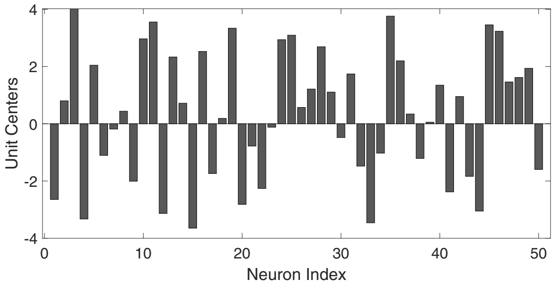

Fig. 15. Unit centers of hidden neurons of the DANN.

under v2x communications: Design and experiments, *IEEE Transactions on Intelligent Transportation Systems* **21**(5): 1891–1902.

Li, Y., Tang, C., Peeta, S. and Wang, Y. (2019). Nonlinear consensus-based connected vehicle platoon control incorporating car-following interactions and heterogeneous time delays, *IEEE Transactions on Intelligent Transportation Systems* **20**(6): 2209–2219.

Liang, X., Xu, C. and Wang, D. (2020). Adaptive neural network control for marine surface vehicles platoon with input saturation and output constraints, *AIMS Math* **5**(1): 587–602.

Liu, H., Zhuang, W., Yin, G., Tang, Z. and Xu, L. (2018). Strategy for heterogeneous vehicular platoons merging in automated highway system, *Chinese Control And Decision Conference (CCDC), Shenyang, China*, pp. 2736–2746.

Liu, Y., Yao, D., Li, H. and Lu, R. (2022). Distributed cooperative compound tracking control for a platoon of vehicles with adaptive NN, *IEEE Transactions on Cybernetics* **52**(7): 7039–7048.

Liu, Y., Zong, C. and Zhang, D. (2019). Lateral control system for vehicle platoon considering vehicle dynamic characteristics, *IET Intelligent Transport Systems* **13**(9): 1356–1364.

Mitrinovic, D.S., Pecaric, J.E. and Fink, A.M. (1993a). *Cauchy's and Related Inequalities*, Springer, Dordrecht.

Mitrinovic, D.S., Pecaric, J.E. and Fink, A.M. (1993b). *Young's Inequality*, Springer, Dordrecht.

Okamoto, A., Feeley, J., Edwards, D. and Wall, R. (2004). Robust control of a platoon of underwater autonomous vehicles, *Oceans'04, MTS/IEEE Techno-Ocean, Kobe, Japan*, pp. 505–510.

Peng, C., Zhang, A. and Li, J. (2021). Neuro-adaptive cooperative control for high-order nonlinear multi-agent systems with uncertainties, *International Journal of Applied Mathematics and Computer Science* **31**(4): 635–645, DOI: 10.34768/amcs-2021-0044.

Prayitno, A., Indrawati, V. and Nilkhamhang, I. (2023). Distributed model reference control for synchronization of a vehicle platoon with limited output information and subject to periodical intermittent information, *International Journal of Applied Mathematics and Computer Science* **33**(4): 537–551, DOI: 10.34768/amcs-2023-0039.

Wang, P., Deng, H., Zhang, J., Wang, L., Zhang, M. and Li, Y. (2022). Model predictive control for connected vehicle platoon under switching communication topology, *IEEE Transactions on Intelligent Transportation Systems* **23**(7): 7817–7830.

Wang, W., Gao, X., Li, T., Wang, Y. and Chen, C.L.P. (2023). Observer-based platoon formation control with prescribed performance guarantees for unmanned surface vehicles in presence of nonsmooth input characteristics, *IEEE Transactions on Circuits and Systems II: Express Briefs* **71**(3): 1226–1230.

Wu, Z., Sun, J. and Hong, S. (2022). RBFNN-based adaptive event-triggered control for heterogeneous vehicle platoon consensus, *IEEE Transactions on Intelligent Transportation Systems* **23**(10): 18761–18773.

Zhang, D., Shen, Y.-P., Zhou, S.-Q., Dong, X.-W. and Yu, L. (2021). Distributed secure platoon control of connected vehicles subject to DoS attack: Theory and application, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **51**(11): 7269–7278.

Zhao, X., Chen, Y.H. and Zhao, H. (2017). Robust approximate constraint-following control for autonomous vehicle platoon systems, *Asian Journal of Control* **20**(4): 1611–1623.

Zhou, H., Saigal, R., Dion, F. and Yang, L. (2012). Vehicle platoon control in high-latency wireless communications environment: Model predictive control method, *Journal of the Transportation Research Board* **2324**(1): 81–90.

Ping Wang is an associate professor at the School of Mechanical and Electrical Engineering, Hefei Technology College. She specializes in electrical control and automation devices. Born in 1981, she graduated in 2008 with a master's degree in detection technology and automation devices from the Anhui University of Engineering. Her research interests include control systems and technology, as well as artificial neural networks.

Min Gao is a distinguished professor and a senior dual-skilled teacher in the field of mechatronics. Recognized as a Hefei category D talent, she holds the title of a school-level distinguished teaching master. Professor Gao has successfully overseen a series of educational projects, and has established a robust teaching team. She has also administered two scholarly teaching and research initiatives, and has participated in a multitude of scientific research projects. Professor Gao has published numerous academic papers, and has been granted five national invention patents. For her coaching excellence she received the distinguished designation of Outstanding Judge at the 2019 National Competition for Intelligent Manufacturing Application Technology Skills.

Junyu Li graduated from the Wuhan Institute of Technology in 2013 with a major in mechanical engineering. She currently works as a lecturer at Hefei Technology College. Her main research interests include mechanical engineering, machine learning and artificial neural networks.

Anguo Zhang received his PhD degree from the College of Physics and Information Engineering, Fuzhuo University, in 2022 and his BSc and MS degrees from the School of Automation, Chongqing University, in 2012 and 2016, respectively. He is currently a research fellow with the Institute of Microelectronics, University of Macau. His research interests include brain-inspired computational intelligence, neuromorphic computing and deep learning.

Received: 6 November 2023
Revised: 5 December 2023
Re-revised: 6 March 2024
Accepted: 10 March 2024