Automatica 156 (2023) 111207

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

# Single parameter adaptive neural network control for multi-agent deployment with prescribed tracking performance ☆

Zhijie Liu [a], Zhongqi Lu [a], Zhijia Zhao [b,*], Mehmet Önder Efe [c], Keum-Shik Hong [d,e]

[a] School of Intelligence Science and Technology, University of Science and Technology Beijing, Beijing 100083, China
[b] School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China
[c] Department of Computer Engineering, Hacettepe University, 37515 Ankara, Turkey
[d] Institute For Future, School of Automation, Qingdao University, Qingdao, Shandong 266071, China
[e] School of Mechanical Engineering, Pusan National University, Busan 46241, Republic of Korea

ARTICLE INFO

*Article history:*
Received 11 October 2021
Received in revised form 5 April 2023
Accepted 20 June 2023
Available online 28 July 2023

*Keywords:*
Adaptive control
RBF neural network control
Multi-agent
Performance constraint

ABSTRACT

This study addresses the problem of deploying multi-agent systems using single-parameter adaptive neural network control in time and space where the system is modeled by a parabolic partial differential equation (PDE). We investigate the agent model, simplify the pointwise dynamics using a PDE model, and consider the deployment problem when the number of agents is relatively large. In order for the deployed agents to follow the desired trajectory, we augment the agent dynamics with individual control inputs, accounting for the unknown interference faced by each agent during the deployment process. In the proposed approach, a radial basis function neural network structure is introduced to enhance the systems' adaptivity under unknown interference. The unknown parameter is estimated via the single-parameter idea for reducing the computation of the entire process and increasing the calculation speed. Asymmetric performance constraints are imposed on the tracking error of the system to ensure that each agent is deployed in the required position. The results of numerical simulation prove the effectiveness of the method.



## 1. Introduction

Multi-Agent Systems (MAS) and associated algorithms for their behaviors, tasks, stability and performance issues have been a prime research area during the past few decades. Recent studies have proposed many control methods for multi-agent systems described by ordinary differential equations (ODEs) (Chen et al., 2019; Khalili et al., 2018; Rezaee et al., 2021), to realize the control of each agent by obtaining global or local information. However, during the analysis of the actual situation, the multi-agent system model should be described with a high-dimensional ODE system. When the number of agents is large enough, the high-dimensional ODE system can be converted to partial differential equations (PDEs) (Chen et al., 2021; Freudenthaler & Meurer, 2020; Pilloni et al., 2015). PDEs can be used to describe flexible structures (He et al., 2020; Liu et al., 2021; Wang & Krstic, 2021), thermal and fluid dynamics (Espitia et al., 2021; Qi et al., 2019), multi-agent systems (Frihauf & Krstic, 2011; Qi et al., 2015) and so on. The PDE-based approach for multi-agent systems is especially powerful when the number of agents is large, further the PDE-based schemes have the advantages that not only can they reduce a high-dimensional ODE system to a single PDE but also can generate more diversified desirable formation manifolds. In Wei et al. (2019), researchers studied the deployment of a first-order multi-agent system over a desired smooth curve in the 3D space. However, this method neither considers the weight of each agent to control nor simultaneously achieves high-precision control of the formation effect.

In the actual deployment process of multi-agent systems, agents also face interference from external factors or uncertainties. In many cases, these uncertainties or disturbances are unknown and nonlinear (He et al., 2018; Liu et al., 2018; Zhao et al., 2021). Many intelligent control algorithms, such as neural networks (NNs) (Zhao, He et al., 2023; Zhao, Zhang et al., 2023) and fuzzy methods, have also been applied without knowing

☆ This work was supported in part by the National Natural Science Foundation of China under Grant 62073030 and Grant 62273112; in part by the Science and Technology Planning Project of Guangzhou City under Grant 202201010758 and Grant 2023A03J0120; and in part by the National Research Foundation of Korea under the Ministry of Science and ICT, Korea under Grant RS-2023-00207954. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Angelo Alessandri under the direction of Editor Thomas Parisini.

* Corresponding author.
*E-mail addresses:* liuzhijie2012@gmail.com (Z. Liu), 812747204@qq.com (Z. Lu), zhjzhaoscut@163.com (Z. Zhao), onderefe@gmail.com (M.Ö. Efe), kshong@pusan.ac.kr (K.-S. Hong).

https://doi.org/10.1016/j.automatica.2023.111207
0005-1098/

the upper bound of the disturbance. To compensate the deficiencies caused by the uncertainty of the interference term, NN methods (Kong et al., 2019; Krstic et al., 1995) were used to approximate the unknown interference, and an appropriate adaptation law was designed. To reduce the computational burden, we used the single-parameter processing idea (Kam et al., 1986) to estimate the ideal weight vector of each layer of the NN. By estimating the square of the norm of the weight vector, we only need to design an adaptive law, which considerably reduces the amount of calculation.

In modern industrial applications, the designed control scheme is required to ensure not only the stability of the controlled system, but also proper control performance. The barrier Lyapunov function (BLF) is widely used to address these performance constraints (Liu et al., 2022, 2020; Zhao, Liu et al., 2023; Zhao et al., 2022). Various BLFs can be constructed to solve different constraint problems. Moreover, we can construct the corresponding BLF to solve the constraint problems of the PDE. The constraint is constant and symmetrical, but cannot achieve precise performance constraints. In the present study, we addressed this shortcoming. By appropriately designing the BLF, we can also change the upper and lower bounds and the symmetry of the performance constraints to improve accuracy.

In this paper, we consider a type of deployment control problem for multi-agent systems. For this, we do not need to have the information about the leader. Each agent can realize its corresponding function within a limited time (Ren & Beard, 2008), but the individual agents are not unrelated. Each must determine its location through other respective locations, as proposed in Li and Duan (2014). We propose a distributed control law for multi-agent systems that are modeled as a parabolic type PDE. Because an individual control input is added to each agent, the deployment trajectory of each agent is different, and the control coefficient of the agent at each location is different. Through this design method, the functions realized by the system can be highly diverse, and the system can be deployed in different ways according to different actual situations. In the proposed approach, the BLF method is used to analyze the tracking performance of the system, and the control law and adaptive laws of the system are designed. We can apply this multi-agent closed-loop control system to the deployment process of unmanned aerial vehicles (UAVs), where each drone needs not be shackled to the leader. The UAV linkage system designed in this way has three obvious advantages. First, it is decentralized; that is, no individual is in a dominant position, and the disappearance or loss of kinetic energy of any individual does not affect the function of the group. The second is autonomous control, that is, all individuals control only individual actions and observe the location of neighboring individuals in real-time autonomous coordination. The third is cluster recovery; that is, when the cluster is disturbed by unknown external forces to change the structure and position of the group, a new cluster organization is quickly and automatically formed, and the system remains stable.

In summary, this study deals with the problem of deployment for MASs in corresponding time and space, in which the system is modeled by a parabolic PDE. Then we design an adaptive formation control scheme to render the deployed agents following an ideal trajectory. A radial basis function neural network structure is used to estimate parametric and nonparametric uncertainties. The unknown parameter is estimated with the single-parameter framework at the same time for reducing the calculation amount of the entire process and increasing the calculation speed. Asymmetric performance constraints are imposed on the tracking error of the system to ensure that each agent is deployed in the required position.

The current paper differentiates from the existing body of literature from the points of (i) generalizing pointwise agent dynamics using a PDE and postulating a PDE control law for the individual agents and introducing the advantage of theoretically unlimited agent number, which is a significant problem in swarm systems; (ii) using a single-parameter based RBF neural network to reduce the amount of calculation and improve the system's efficiency, and neural network support in the control law that is explained in the Lyapunov sense.

The rest of this paper is organized as follows. The system modeling and equivalent description are presented in Section 2. Section 3 describes the design process of the adaptive NN control scheme and stability analysis using the BLF. Section 4 discusses the simulation results, and the concluding remarks are given in Section 5.

**Notations:** For clarity, the notations of the partial derivatives $\partial(*)/\partial t$, $\partial(*)/\partial x$ and $\partial(*)^2/\partial^2 x$ are replaced by $(*)_t$, $(*)_x$ and $(*)_{xx}$, respectively. The superscript 'T' is used for the transpose of a vector. $R$ denotes the set of all real numbers. We denote $\|\cdot\|$ the norm in the Euclidean space, and use the notation $L^2(0,l)$ for the space of measurable squared integrable functions with the norm $\|f\|_2^2 = \int_0^l f^2 dx$. $L^\infty(0,l)$ represents the space of essentially bounded functions and is endowed with the $L^\infty$ norm $\|f\|_\infty^2 = sup_{x\in[0,l]} f^2$.

## 2. System description

Multi-agent systems have a wide range of applications in UAV formation control. In military operations, UAV systems can complete combat tasks autonomously and intelligently. By approximating the model of the system and building the model on the basis of the PDE, a distributed control input is applied to the whole. By using this processing method, the entire UAV group can operate in accordance with a predetermined formation trajectory.

The purpose of this study is to seek for a simple but effective control law to maintain $N$ agents in the required formation. In this study, we consider a typical agent motion equation (Ferrari-Trecate et al., 2006), where the motion of the agent is controlled by the position information of adjacent agents, defined by

$$\dot{z}_i(t) = \frac{z_{i+1}(t) - 2z_i(t) + z_{i-1}(t)}{h^2} + b_i u_i(t) + f(\mathbf{z}_i(t)) \tag{1}$$

where $h$ is the distance between adjacent agents, defined by $h = \frac{l}{N-1}$, $l$ is the distance occupied by all agents, and $z_i(t)$ $i = 1, 2, \ldots, N$ is the position of each agent at different time instants. $u_i(t)$ is the control input of each agent, and has different control coefficients $b_i$. $f(\mathbf{z}_i(t))$ is the unknown interference encountered by each agent in the deployment process, and is related to the position and speed of the agent with $\mathbf{z}_i(t) = [z_i(t)\ \dot{z}_i(t)]^T$. Through mutual communication between neighboring agents, a variety of deployment actions can be realized.

**Remark 1.** $b_i$ in (1) is peculiar to a certain agent and can be used to solve the problem of agent heterogeneity. Different agents have different qualities owing to their different functions, and intuitively, their control weights are different. This unknown parameter can also be used to handle the problem of partial actuator failure. When a part of the actuator fails, it does not affect the function of the entire MAS.

When the number of agents is large, the system (1) is approximated by a parabolic equation:

$$z_t(x,t) = z_{xx}(x,t) + b(x)u(x,t) + f(\mathbf{z}(x,t)) \tag{2}$$

where $x$ denotes the spatial location, and $0 < x < l$. $z(x,t)$ is the position of the agent measured by the sensor, $u(x,t)$ is the control input of the system, and $b(x)$ represents the control coefficient, that is related to position $x$. $f(\mathbf{z}(x,t))$ is the unknown smooth nonlinear function which contains both parametric and nonparametric uncertainties. Boundary condition of the system is the Dirichlet case $z(0,t)=z(l,t)=0$. This boundary condition does not mean that the agents at both ends do not move. However, there are no actual agents at the boundaries, only virtual agents. The system parameter $b(x)$ is not available, therefore, we can employ an adaptive control method to estimate the value of $b(x)$. $\hat{\eta}(x,t)$ is the dynamic compensation for $\frac{1}{b(x)}$, and $\tilde{\eta}(x,t)=\hat{\eta}(x,t)-\frac{1}{b(x)}$ is the error of the estimation.

**Assumption 1.** For the multi-agent system (2), the communication topology between agents is undirected and connected.

**Assumption 2.** We assume that $\|f(\mathbf{z}(x,t))\|_2$ is a continuous function defined on a compact set $U\subset R$.

**Remark 2.** The parabolic equations are used to model multi-agents in some studies (He, 2018; Wei et al., 2019) and can achieve good control performance. However, the unknown system parameter b(x)and model uncertainties are not taken into account. From the model in (2), it is observed that the controllers are imposed on every agent, which means that multiple actuators and senors are needed.

To realize the compensation of the unknown function $f(\mathbf{z}(x,t))$, we use the separation of variables method to write it as a product of a function of space and time,

$$f(\mathbf{z}(x,t))=g_1(x)\cdot g_2(\|\mathbf{z}(x,t)\|_2)$$

with $\mathbf{z}(x,t)=[z(x,t)\ z_t(x,t)]^T$, and an RBF-based NN framework is employed. Because the RBF NN has good approximation ability, it can be used to approximate unknown nonlinear functions. The input/output relation of the RBF NN is given as

$$\begin{aligned}\|f(\mathbf{z}(x,t))\|_2&=\|g_1(x)\|_2\cdot|g_2(\|\mathbf{z}(x,t)\|_2)|\\&=\mathbf{G}^{*\mathrm{T}}\mathbf{h}\left(\|\mathbf{z}(x,t)\|_2\right)+\delta(t)\end{aligned}\tag{3}$$

where $\|\mathbf{z}(x,t)\|_2$ is the input of the network, $m$ is the number of network nodes, $\mathbf{G}^*\in R^m$ is the ideal constant weight vector, and $\delta(t)$ is the approximation error of the network, with a bound $\bar{\delta}=\sup_{t\geq 0}|\delta(t)|$. $\mathbf{h}(\|\mathbf{z}(x,t)\|_2)=[h_1,h_2,\ldots,h_m]^{\mathrm{T}}$, which is in the form of Gaussian functions defined by

$$h_i=\exp\left(-|\ \|\mathbf{z}(x,t)\|_2-c_i|^2/2d_i^2\right),\ i=1\ldots m$$

where $c_i\in R$ is the center of the receptive field and $d_i>0$ is the width of the Gaussian function.

**Remark 3.** The proposed method in this study can increase the computational burden when the number of network nodes is large. Here, we do not estimate the unknown function, but estimate its norm; therefore, we need to only introduce an unknown parameter. Designing the controller according to this single-parameter processing idea can reduce the computational burden of the system and the communication burden of the controller.

The control goal of the system is to allow each agent to be in its ideal deployment position at different times. The ideal tracking signal is considered as $\alpha(x,t)$, which is bounded by $\alpha(0,t)=\alpha(l,t)=0$. Therefore, the tracking error of the system output signal is $e(x,t)=z(x,t)-\alpha(x,t)$, and $e(0,t)=e(l,t)=0$. By constraining the tracking error of the system, the deployment error of the multi-agent system can be reduced. Here, the following performance function is introduced:

$$\lambda(x,t)=(\lambda_0-\lambda_\infty)e^{-kxt}+\lambda_\infty$$

where $k>0$ and $\lambda_0>\lambda_\infty>0$ are constants. The tracking error satisfies the specified transient and steady-state performance when the following constraints are satisfied:

$$-\theta_2\lambda(x,t)\leq e(x,t)\leq\theta_1\lambda(x,t)$$

where $\theta_1,\theta_2>0$ are adjustable constants. To facilitate the subsequent analysis, we denote $\lambda_a(x,t)=\theta_2\lambda(x,t)$, $\lambda_b(x,t)=\theta_1\lambda(x,t)$.

The following lemmas are used in the subsequent control design and stability analysis.

**Lemma 1** (*Young's Inequality*)**.** *For any scalars $w_1$ and $w_2\in R$, the following inequality holds*

$$w_1w_2\leq\gamma w_1^2+\frac{1}{\gamma}w_2^2\tag{4}$$

*where $\gamma$ is a positive constant.*

**Lemma 2** (*Cauchy–Schwarz Inequality (Rahn, 2001)*)**.** *For any vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ in the Euclidean space $R^n$ with the standard inner product, the following inequality holds*

$$|\boldsymbol{u}^T\boldsymbol{v}|\leq\|\boldsymbol{u}\|\|\boldsymbol{v}\|.\tag{5}$$

## 3. Control design and stability analysis

The specific formation of the multi-agent system (2) is governed by a PDE that manages the space and time relations. By designating control inputs in the domain for individual agents, the position of the agent at $x\in[0,l]$ is forced to the desired formation objective $\alpha(x,t)$, and the tracking error $e(x,t)$ remains within the predefined performance function for all $(x,t)\in[0,l]\times[0,\infty)$, even with unknown perturbations.

To achieve asymmetric constraints, we construct the following BLF

$$\begin{aligned}\Xi=&(1-q(e))\ln\frac{2\lambda_a^2(x,t)}{\lambda_a^2(x,t)-e^2(x,t)}\\&+q(e)\ln\frac{2\lambda_b^2(x,t)}{\lambda_b^2(x,t)-e^2(x,t)}\end{aligned}\tag{6}$$

where

$$q(e)=\begin{cases}1, e(x,t)>0\\0, e(x,t)\leq 0\end{cases}$$

Let $\varepsilon_a(x,t)=e(x,t)/\lambda_a(x,t)$, $\varepsilon_b(x,t)=e(x,t)/\lambda_b(x,t)$, $\varepsilon(x,t)=(1-q)\varepsilon_a(x,t)+q\varepsilon_b(x,t)$. For the convenience of the subsequent analysis, (6) is simplified to

$$\Xi=\ln\frac{2}{1-\varepsilon^2(x,t)}\tag{7}$$

To achieve the aforementioned formation control, we propose the following NN-based adaptive control algorithm:

$$\begin{aligned}u(x,t)=&-\frac{\mathcal{E}}{\Xi}\hat{\eta}(x,t)e(x,t)-\hat{\eta}(x,t)[z_{xx}(x,t)-\alpha_t(x,t)]\\&-\Xi e(x,t)\hat{\eta}(x,t)\frac{\hat{\mathbf{G}}^{\mathrm{T}}\mathbf{h}\left(\|\mathbf{z}(x,t)\|_2\right)}{\|\Xi e(x,t)\|_2}-\sigma_1\hat{\eta}(x,t)\Xi e(x,t)\\&-k_1\hat{\eta}(x,t)e(x,t)\end{aligned}\tag{8}$$

where

$$\|\Xi e(x,t)\|_2=\begin{cases}\|\Xi e(x,t)\|_2, & \|e(x,t)\|_\infty>\rho_0\\ \|\Xi e(x,t)\|_2\big|_{t=t_0}, & \|e(x,t)\|_\infty\leq\rho_0\end{cases}$$

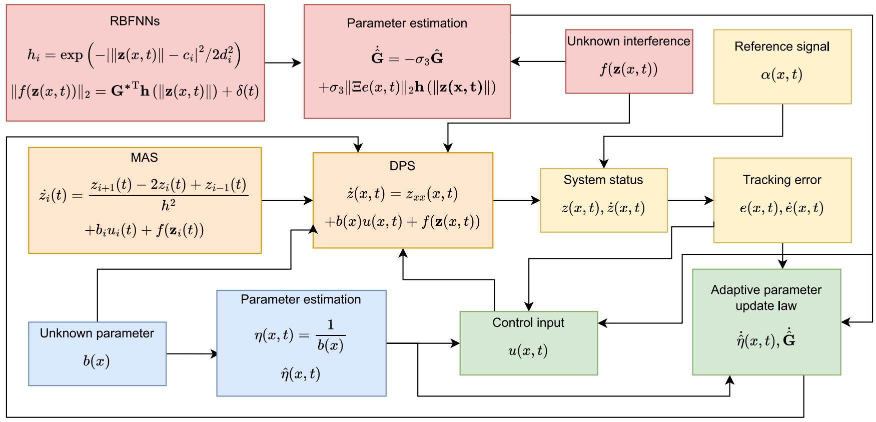



**Fig. 1.** The flowchart of the proposed NN-based adaptive formation control scheme.

with parameter updating laws

$$\begin{aligned}\hat{\eta}_t(x,t) =& -\sigma_2\hat{\eta}(x,t) - \sigma_2\Xi e(x,t)(z_{xx}(x,t)\\ &-\alpha_t(x,t) + \mathcal{E}e(x,t)) - \sigma_2\Xi e(x,t)^2(k_1\\ &+\sigma_1\Xi + \Xi\frac{\hat{\mathbf{G}}^{\mathrm{T}}\mathbf{h}(\|\mathbf{z}(\mathbf{x},\mathbf{t})\|_2)}{\|\Xi e(x,t)\|_2})\end{aligned} \tag{9}$$

$$\dot{\hat{\mathbf{G}}} = -\sigma_3\hat{\mathbf{G}} + \sigma_3\|\Xi e(x,t)\|_2\mathbf{h}(\|\mathbf{z}(x,t)\|_2) \tag{10}$$

where $t_0 = \arg\{\|e(x,t)\|_\infty = \rho_0\}$, $\rho_0 < \min\{\theta_1\lambda_0, \theta_2\lambda_0\}$, $k_1$, $\sigma_1$, $\sigma_2$, and $\sigma_3$ are designed constants. $\mathcal{E}$ is defined by $\mathcal{E} = \varepsilon(x,t)\varepsilon_t(x,t)/(1-\varepsilon^2(x,t))$, $\hat{\mathbf{G}}$ is the estimates of the ideal weight $\mathbf{G}^*$ (see Fig. 1).

**Remark 4.** All the signals in the controller can be measured by sensors or obtained by a backward difference algorithm. $z(x,t)$ represents the state of each agent obtained from optitrack motion system capture and $z_{xx}(x,t)$ can be obtained by differencing the values of $z_{i-1}(x,t)$, $z_i(x,t)$, and $z_{i+1}(x,t)$. It can be seen that the topology between the agents is a chain-like topology.

The following theorem is used to analyze the deployment performance of the multi-agent system.

**Theorem 1.** *For the multi-agent system (2) with unknown interference and control coefficients, the NN-based control (8) and parameter updating laws (9) and (10) allow each agent to move according to a predefined deployment plan and maintain the deployment error within a pre-defined open-loop bounds $(-\theta_2\lambda_0, \theta_1\lambda_0)$ in the whole time domain.*

**Proof.** To evaluate the stability of the closed-loop system with the NN-based adaptive control law, we define the final Lyapunov function as

$$V(t) = V_1(t) + V_2(t) \tag{11}$$

where

$$\begin{aligned}V_1(t) =& \frac{1}{2}\int_0^l \Xi e^2(x,t)dx,\\ V_2(t) =& \int_0^l \frac{b(x)}{2\sigma_2}\tilde{\eta}^2(x,t)dx + \frac{1}{2\sigma_3}\tilde{\mathbf{G}}^{\mathrm{T}}\tilde{\mathbf{G}}\end{aligned}$$

with $\tilde{\eta}(x,t) = \hat{\eta}(x,t) - \frac{1}{b(x)}$ and $\tilde{\mathbf{G}} = \hat{\mathbf{G}} - \mathbf{G}^*$.

Differentiating $V_1(t)$ along the solutions of (2), then combining (8), (3), and Lemma 2, we can get

$$\begin{aligned}\dot{V}_1(t) \le& \int_0^l \Xi e(x,t)(1 - \hat{\eta}(x,t)b(x))(z_{xx}(x,t) - \alpha_t(x,t)\\ &+ \mathcal{E}e(x,t))dx + \|\Xi e(x,t)\|_2(\mathbf{G}^{*\mathrm{T}}\mathbf{h}(\|\mathbf{z}(x,t)\|_2)\\ &+ \delta(x,t)) + \int_0^1 \Xi e(x,t)^2\hat{\eta}(x,t)b(x)(k_1 + \Xi\sigma_1\\ &+ \Xi\frac{\hat{\mathbf{G}}^{\mathrm{T}}\mathbf{h}(\|\mathbf{z}(\mathbf{x},\mathbf{t})\|_2)}{\|\Xi e(x,t)\|_2})dx\end{aligned} \tag{12}$$

Taking the derivative of $V_2(t)$ and combining the parameter updating laws (9) and (10) yields

$$\begin{aligned}\dot{V}_2(t) =& -\int_0^l \frac{b(x)}{\sigma_2}\tilde{\eta}(x,t)\hat{\eta}_t(x,t)dx - \frac{1}{\sigma_3}\tilde{\mathbf{G}}^{\mathrm{T}}\dot{\hat{\mathbf{G}}}\\ =& \int_0^l b(x)\Xi\tilde{\eta}(x,t)e(x,t)(z_{xx}(x,t) - \alpha_t(x,t)\\ &+ \mathcal{E}e(x,t))dx + \int_0^l b(x)\tilde{\eta}(x,t)\hat{\eta}(x,t)dx\\ &+ \int_0^l b(x)\Xi\tilde{\eta}(x,t)e^2(x,t)[(\sigma_1\Xi + k_1\\ &+ \Xi\frac{\hat{\mathbf{G}}^{\mathrm{T}}\mathbf{h}(\|\mathbf{z}(\mathbf{x},\mathbf{t})\|_2)}{\|\Xi e(x,t)\|_2})]dx + \tilde{\mathbf{G}}^{\mathrm{T}}\hat{\mathbf{G}}\\ &- \|\Xi e(x,t)\|_2\tilde{\mathbf{G}}^{\mathrm{T}}\mathbf{h}(\|\mathbf{z}(x,t)\|_2)\end{aligned} \tag{13}$$

Further, combining (12) and (13) and Lemma 1, one obtains

$$\begin{aligned}\dot{V}(t) \le& -k_1\int_0^l \Xi e^2(x,t)dx - \frac{1}{2}\tilde{\mathbf{G}}^{\mathrm{T}}\tilde{\mathbf{G}}\\ &- \frac{1}{2}\int_0^l b(x)\tilde{\eta}^2(x,t)dx + \frac{1}{2}\int_0^l \frac{1}{b(x)}dx\\ &+ \frac{1}{2}\mathbf{G}^{*\mathrm{T}}\mathbf{G}^* + \frac{1}{\sigma_1}\bar{\delta}^2(t)\end{aligned} \tag{14}$$

Let $\chi = \min\{2k_1, \sigma_2, \sigma_3\}$ and consider (11) and (14), we obtain

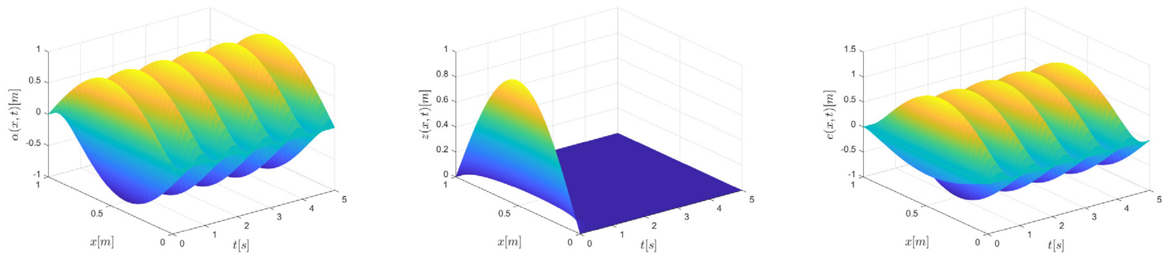
$$\dot{V}(t) \le -\chi V(t) + \Delta \tag{15}$$

where

$$\Delta = \frac{1}{2}\int_0^l \frac{1}{b(x)}dx + \frac{1}{2}\mathbf{G}^{*\mathrm{T}}\mathbf{G}^* + \frac{1}{\sigma_1}\bar{\delta}^2(t)$$

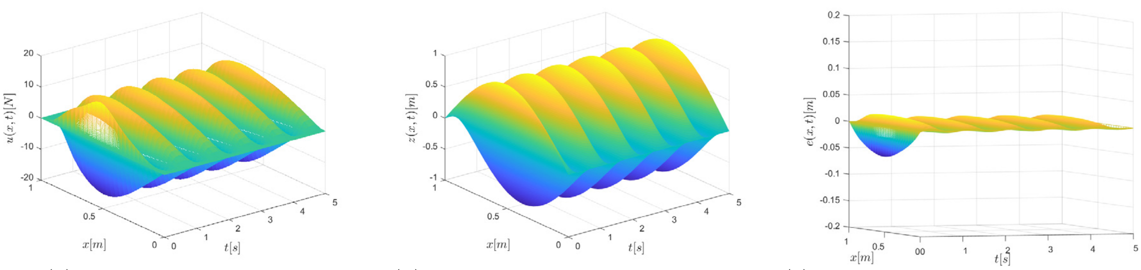
Multiplying both sides of (15) by $e^{\chi t}$ yields

$$\dot{V}(t)e^{\chi t} + \chi e^{\chi t}V(t) \le \Delta e^{\chi t} \tag{16}$$




(a) Desired reference signal (b) System state without control (c) Tracking error without control

**Fig. 2.** Simulated responses of the UAV system without control.




(a) System control input (b) System state with control input (c) The tracking error with control input

**Fig. 3.** Simulated responses of the UAV system with the proposed control.

Then, integrating both sides of (16), we get

$$V(t) \leq e^{-\chi t}(V(0) - \frac{\Delta}{\chi}) + \frac{\Delta}{\chi} \leq V(0)e^{-\chi t} + \frac{\Delta}{\chi} \tag{17}$$

Thus, we can conclude that $V(t)$ is bounded, and further the BLF (6) is bounded. The tracking error $e(x, t)$ remains within the predefined performance function for all $(x, t) \in [0, l] \times [0, \infty)$. The proof is finished.

**Remark 5.** The center $\mathbf{c}_i$ and the width $d_i$ of RBF are fixed arbitrarily. However, the above parameters can be chosen by adding an off-line learning step to find different clusters of their centers and widths such that the unknown smooth nonlinear function that contains both parametric and nonparametric uncertainties is better estimated. Another useful method to specify the center is to randomly select a subset of the input patterns based on the input range, and then the width can be determined according to the data distribution in the region of the corresponding center.

**Remark 6.** In contrast to the constraint problem of ODE-governed systems, the proposed PDE-based approach can constrain the distributed tracking error in a time-varying bound the constructed BLF $V_1(t)$ in (11), that is, every agent can track a preset deployment trajectory and maintain the deployment error within a prescribed bound. Moreover, based on the PDE-based scheme, the corresponding performance and the associated control scheme for the individual agents can be obtained. This control design is independent of the number of agents, provided this number is large enough.

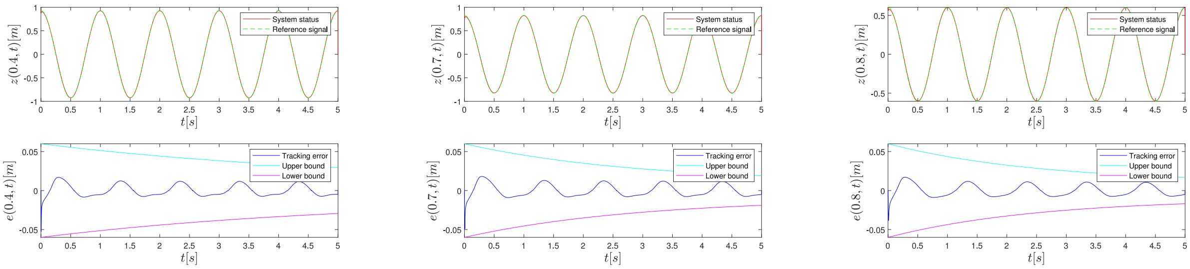
## 4. Simulation results

In this study, we use the finite difference method to simulate the system performance with distributed control. The space and time scales are selected by 1 m and 5 s, respectively. Dividing the domain by a mesh of discrete points of $x$ and $t$ allows the finite difference method to solve numerical solutions to the PDE-based multi agent system.

Consider a multi-agent system with reference signal $\alpha(x, t) = \sin(\pi x)\cos(2\pi t)$. We assume that the unknown interference is set as $f(\mathbf{z}(x, t)) = 0.1 \sin(x)(\int_0^1 z^2(x, t) + z_t^2(x, t))dx$. The control coefficient $b(x)$ is chosen as $\exp(-x)$. The initial values of parameter estimation scheme are set as $\hat{\eta}(x, 0) = 0$. The initial condition of the system is $z(x, 0) = \sin(\pi x)$. The constraint boundaries are given by $\lambda_a = \lambda_b = 0.05e^{-0.5xt} + 0.01$. The adjustable control gains are set as $k_1 = 12$, $\rho_0 = 0.5$, $\sigma_1 = 0.5$, $\sigma_2 = 0.5$, and $\sigma_3 = 0.5$. The utilized neural network has 5 nodes. Spreads and the centers associated to the neural network are given as $d_i = 1$ and

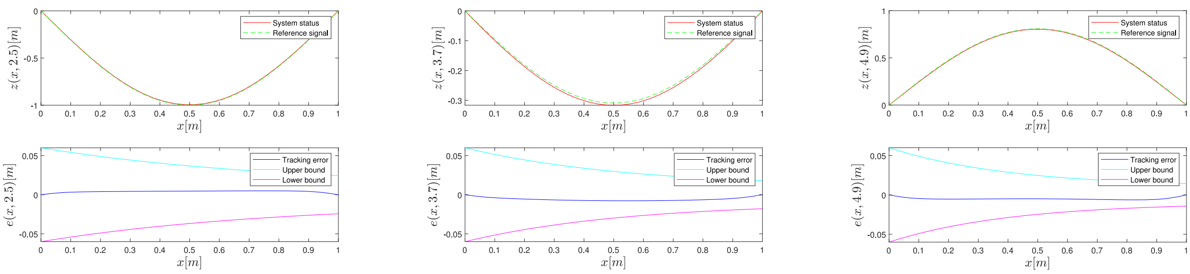
$$\mathbf{c} = \begin{pmatrix} -6 & -3 & 0 & 3 & 6 \end{pmatrix}$$

The simulation results are shown in Figs. 2–5. As depicted by Fig. 2(a), the reference trajectory evolves spatiotemporally. Fig. 2(b) shows the trajectory of the multi-agent system when there is no control input. The state of the system finally tends to a stable solution but cannot achieve the desired formation. To deploy the multi-agent system according to the desired spatiotemporal reference, we synthesize the control signal shown in Fig. 3(a). Fig. 3(b) shows the deployment state of the MAS after adding the control inputs. As seen from the figure, the deployment state of the multi-agent system is quite close to the ideal deployment state. Fig. 3(c) shows the tracking error of each agent in the deployment process. After a short adjustment time, the tracking error of the system is maintained within a certain range that is uniformly bounded. The effect of the constraint is different along the formation. To show the simulation results more clearly, some numerical examples are shown in Figs. 4 and 5 at different positions. In Fig. 4, the positions at $x = 0.4$, 0.7, and 0.8 are selected and the corresponding agent states and tracking errors are given. In Fig. 5, the states and tracking errors of all agents at $t = 2.5$, 3.7, and 4.9 are selected and given, respectively. It can be seen from these figures that the tracking error of the system is constrained within a small bound. From the simulation results, the trajectory tracking can be achieved without violating the constraints for all individual agents and over the entire simulation time scale.



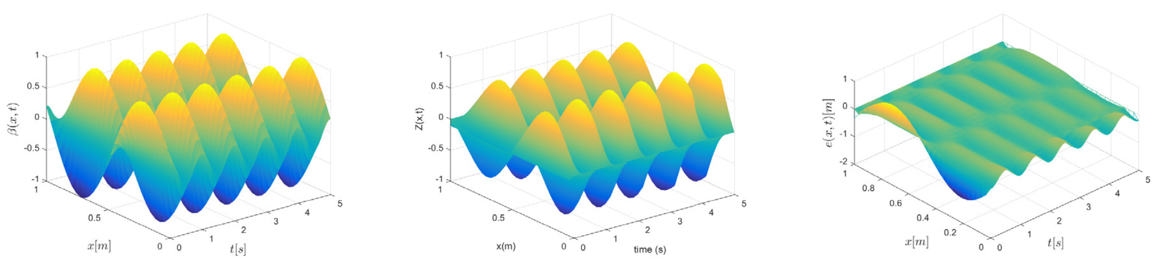

(a) (b) (c)

**Fig. 4.** Displacement and tracking error with reference signal $\alpha(x,t)$ at (a) $x = 0.4$ m, (b) $x = 0.7$ m, (c) $x = 0.8$ m.



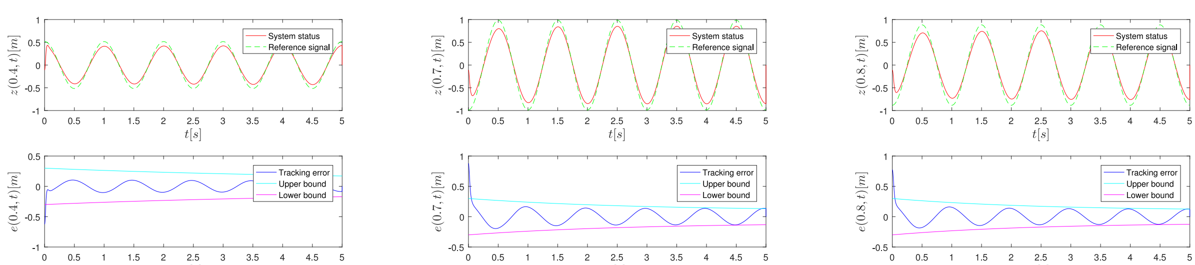

(a) (b) (c)

**Fig. 5.** Displacement and tracking error with reference signal $\alpha(x,t)$ at (a) $t = 2.5$ s, (b) $t = 3.7$ s, and (c) $t = 4.9$ s.




(a) Desired reference signal (b) System state with control input (c) The tracking error with control input

**Fig. 6.** Simulated responses of the UAV system with reference signal $\beta(x,t)$.




(a) (b) (c)

**Fig. 7.** Displacement and tracking error with reference signal $\beta(x,t)$ at (a) $x = 0.4$ m, (b) $x = 0.7$ m, (c) $x = 0.8$ m.

To verify the validity in the 2D plane, we consider another multi-agent system with reference signal $\beta(x,t) = \sin(2\pi x)\cos(2\pi t)$ and assume that the unknown interference is also set as $f(\mathbf{z}(x,t)) = 0.1\sin(x)(\int_0^1 z^2(x,t) + z_t^2(x,t))dx$. The rest of the system parameters is the same as the above system with reference signal $\alpha(x,t)$. The simulation results are shown in Figs. 6–8. By applying the method in Frihauf and Krstic (2011), we combine two uncoupled systems to obtain a deployment on a 2D plane. Each agent is represented by a hollow circle in figures, and the total number of agents is 30. Fig. 9 shows the numerical example of a 2-D multi-agent deployment the from initial curve to final target curve.

## 5. Conclusion

This study mainly considers the deployment of multi-agent systems with unknown perturbation and control factors. To

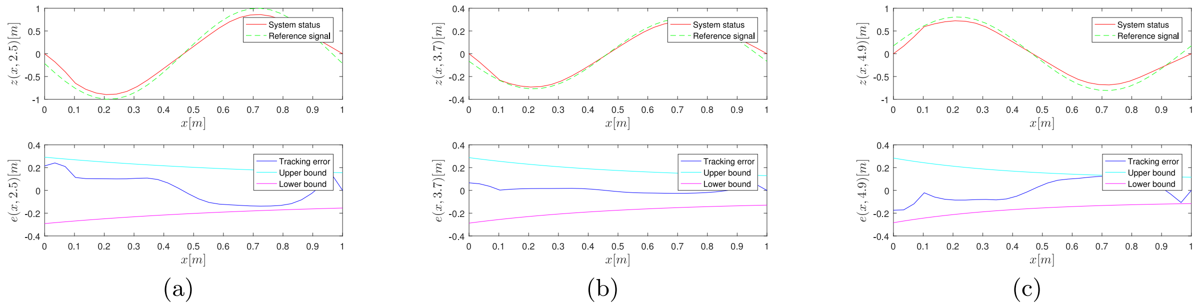



**Fig. 8.** Displacement and tracking error with reference signal $\beta(x, t)$ at (a) $t = 2.5$ s, (b) $t = 3.7$ s, and (c) $t = 4.9$ s.

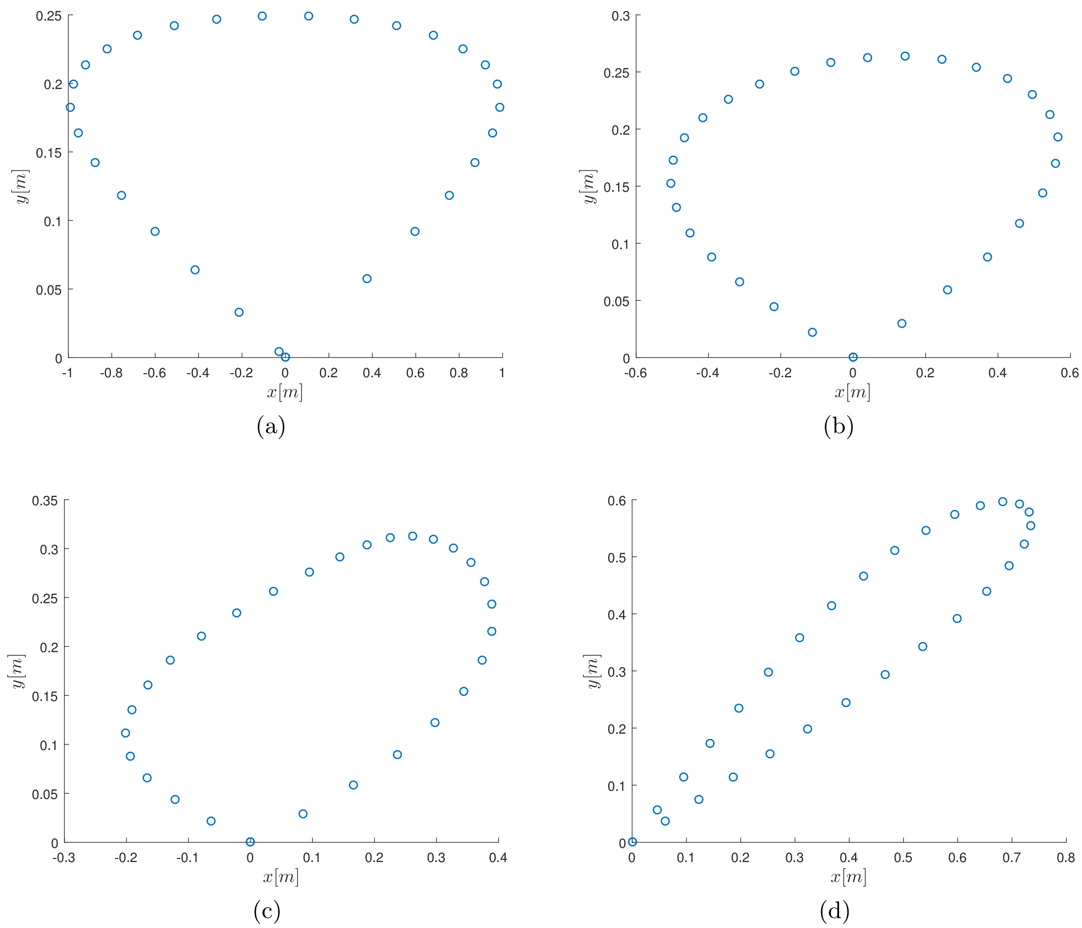



**Fig. 9.** The evolution of the formation.

strengthen deployment capabilities of the entire multi-agent system and achieve more extensive functions, we apply the control inputs to each agent. The control coefficients of each agent vary according to contributions to the formation of the entire agent system. For an individual agent in the control loop, the reported formation algorithm requires communication with its nearest neighbors. In real circumstances, designers must consider unknown interference and constraints, that are limited in the tracking error of the system, to achieve the smallest possible tracking effect. By adopting NN-based adaptive control design and BLF construction, we can solve this multi-agent deployment problem. The proposed adaptive method also applies to possible actuator failures. However, this method can only constrain the tracking error to a small adjustable boundary, and the asymptotic regulation of the motion of the agent system will be investigated. The problem of tracking control and constraints for high-dimensional multi-agent systems is also a future work.

## References


Chen, B., Li, P., Pin, G., Fedele, G., & Parisini, T. (2019). Finite-time estimation of multiple exponentially-damped sinusoidal signals: A kernel-based approach. *Automatica*, *106*, 1–7.

Chen, Y., Zuo, Z., & Wang, Y. (2021). Bipartite consensus for a network of wave PDEs over a signed directed graph. *Automatica*, *129*, Article 109640.

Espitia, N., Karafyllis, I., & Krstic, M. (2021). Event-triggered boundary control of constant-parameter reaction–diffusion PDEs: A small-gain approach. *Automatica*, *128*, Article 109562.

Ferrari-Trecate, G., Buffa, A., & Gati, M. (2006). Analysis of coordination in multi-agent systems through partial difference equations. *IEEE Transactions on Automatic Control*, *51*(6), 1058–1063. http://dx.doi.org/10.1109/TAC.2006.876805.

Freudenthaler, G., & Meurer, T. (2020). PDE-based multi-agent formation control using flatness and backstepping: Analysis, design and robot experiments. *Automatica*, *115*(108897), 1–13.

Frihauf, P., & Krstic, M. (2011). Leader-enabled deployment onto planar curves: A PDE-based approach. *IEEE Transactions on Automatic Control*, *56*(8), 1791–1806. http://dx.doi.org/10.1109/TAC.2010.2092210.

He, P. (2018). Consensus of uncertain parabolic PDE agents via adaptive unit-vector control scheme. *IET Control Theory & Applications*, *12*(18), 2488–2494.

He, W., Meng, T., He, X., & Sun, C. (2018). Iterative learning control for a flapping wing micro aerial vehicle under distributed disturbances. *IEEE Transactions on Cybernetics*, *49*(4), 1524–1535.

He, W., Mu, X., Zhang, L., & Zou, Y. (2020). Modeling and trajectory tracking control for flapping-wing micro aerial vehicles. *IEEE/CAA Journal of Automatica Sinica*, *8*(1), 148–156.

Kam, W. Y., Morris, A. J., Tham, M. T., & Jones, R. W. (1986). A comparison of several single variable parameter adaptive controllers. In *Proceedings of the 7th IFAC/IFIP/IMACS conference* (pp. 379–386). Vienna, Austria: Elsevier.

Khalili, M., Zhang, X., Polycarpou, M. M., Parisini, T., & Cao, Y. (2018). Distributed adaptive fault-tolerant control of uncertain multi-agent systems. *Automatica*, *87*, 142–151.

Kong, L., He, W., Yang, C., Li, Z., & Sun, C. (2019). Adaptive fuzzy control for coordinated multiple robots with constraint using impedance learning. *IEEE Transactions on Cybernetics*, *49*(8), 3052–3063. http://dx.doi.org/10.1109/TCYB.2018.2838573.

Krstic, M., Kokotovic, P. V., & Kanellakopoulos, I. (1995). *Nonlinear and adaptive control design*. New York, NY, USA: Wiley.

Li, Z., & Duan, Z. (2014). *Cooperative control of multi-agent systems: A consensus region approach*. Boca Raton, FL, USA: CRC Press.

Liu, Y., Chen, X., Mei, Y., & Wu, Y. (2022). Observer-based boundary control for an asymmetric output-constrained flexible robotic manipulator. *Science China. Information Sciences*, *65*(3), 1–3.

Liu, Y., Fu, Y., He, W., & Hui, Q. (2018). Modeling and observer-based vibration control of a flexible spacecraft with external disturbances. *IEEE Transactions on Industrial Electronics*, *66*(11), 8648–8658.

Liu, Z., Han, Z., Zhao, Z., & He, W. (2021). Modeling and adaptive control for a spatial flexible spacecraft with unknown actuator failures. *Science China. Information Sciences*, *64*(152208), 1–16.

Liu, L., Liu, Y.-J., Chen, A., Tong, S., & Chen, C. L. P. (2020). Integral barrier Lyapunov function-based adaptive control for switched nonlinear systems. *Science China. Information Sciences*, *63*(3), 1–14.

Pilloni, A., Pisano, A., Orlov, Y., & Usai, E. (2015). Consensus-based control for a network of diffusion PDEs with boundary local interaction. *IEEE Transactions on Automatic Control*, *61*(9), 2708–2713.

Qi, J., Krstic, M., & Wang, S. (2019). Stabilization of reaction–diffusions PDE with delayed distributed actuation. *Systems & Control Letters*, *133*, Article 104558.

Qi, J., Vazquez, R., & Krstic, M. (2015). Multi-agent deployment in 3D via PDE control. *IEEE Transactions on Automatic Control*, *60*(4), 891–906. http://dx.doi.org/10.1109/TAC.2014.2361197.

Rahn, C. D. (2001). *Mechatronic control of distributed noise and vibration: A lyapunov approach*. Berlin, Germany: Springer-Verlag.

Ren, W., & Beard, R. (2008). *Distributed consensus in multi-vehicle cooperative control, vol. 27*. Berlin, Germany: Springer.

Rezaee, H., Parisini, T., & Polycarpou, M. M. (2021). Resiliency in dynamic leader–follower multiagent systems. *Automatica*, *125*, Article 109384.

Wang, J., & Krstic, M. (2021). Adaptive event-triggered PDE control for load-moving cable systems. *Automatica*, *129*, Article 109637.

Wei, J., Fridman, E., Selivanov, A., & Johansson, K. H. (2019). Multi-agent deployment under the leader displacement measurement: a PDE-based approach. In *2019 18th European control conference (ECC)* (pp. 2424–2429). IEEE.

Zhao, Z., He, W., Zhang, F., Wang, C., & Hong, K.-S. (2023). Deterministic learning from adaptive neural network control for a 2-DOF helicopter system with unknown backlash and model uncertainty. *IEEE Transactions on Industrial Electronics*, *70*(9), 9379–9389.

Zhao, Z., Liu, Z., He, W., Hong, K.-S., & Li, H.-X. (2021). Boundary adaptive fault-tolerant control for a flexible Timoshenko arm with backlash-like hysteresis. *Automatica*, *130*, Article 109690.

Zhao, Z., Liu, Y., Zou, T., Hong, K.-S., & Li, H.-X. (2023). Robust adaptive fault-tolerant control for a riser-vessel system with input hysteresis and time-varying output constraints. *IEEE Transactions on Cybernetics*, *53*(6), 3939–3950.

Zhao, Z., Zhang, J., Liu, Z., He, W., & Hong, K.-S. (2023). Adaptive quantized fault-tolerant control of a 2-DOF helicopter system with actuator fault and unknown dead zone. *Automatica*, *148*, Article 110792.

Zhao, Z., Zhang, J., Liu, Z., Mu, C., & Hong, K.-S. (2022). Adaptive neural network control of an uncertain 2-DOF helicopter with unknown backlash-like hysteresis and output constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 1–10.

**Zhijie Liu** received B.Sc. degree from China University of Mining and Technology Beijing, Beijing, China in 2014, and the Ph.D degree from Beihang University, Beijing, China in 2019. In 2017, he was a Research Assistant with the Department of Electrical Engineering, University of Notre Dame, for twelve months. He is currently a Full Professor with School of Intelligence Science and Technology, University of Science and Technology Beijing, Beijing, China. His research interests include adaptive control, modeling and vibration control for flexible structures, and distributed parameter system.

**Zhongqi Lu** received the B.Eng. degree in control science and engineering in 2020 from the University of Science and Technology Beijing, Beijing, China, where he is currently working toward the Ph.D. degree in control science and engineering with the School of Intelligence Science and Technology. His current research interests include flapping wing robots, multi-agent, and neural network control.

**Zhijia Zhao** received his B. Eng. degree in automatic control from North China University of Water Resources and Electric Power, Zhengzhou, China, in 2010, and his M.Eng. and Ph.D. degree in automatic control from South China University of Technology, Guangzhou, China, in 2013 and 2017, respectively. He is currently an Associate Professor in the School of Mechanical and Electrical Engineering, Guangzhou University. His research interests include adaptive and learning control, flexible mechanical systems, and robotics.

**Mehmet Önder Efe** received the B.Sc. degree in electronics and communications engineering from Istanbul Technical University, Istanbul, Turkey, in 1993, the M.Sc. degree in systems and control engineering from Boğaziçi University, Istanbul, Turkey, in 1996, and the Ph.D. degree in electrical and electronics engineering from Boğaziçi University, in 2000 . He is currently a Professor with the Department of Computer Engineering, Hacettepe University, Ankara, Turkey. Dr. Efe serves and served as an Associate Editor of the IEEE Transactions on Industrial Electronics, IEEE Transactions on Industrial Informatics, IEEE/ASME Transactions on Mechatronics, and Editor of Transactions of the Institute of Measurement and Control and Measurement and Control.

**Keum-Shik Hong** received his B.S. degree in Mechanical Design from Seoul National University in 1979, his M.S. degree in Mechanical Engineering from Columbia University, New York, in 1987, and both an M.S. degree in Applied Mathematics and a Ph.D. in Mechanical Engineering from the University of Illinois at Urbana-Champaign in 1991. He joined the School of Mechanical Engineering, Pusan National University, in 1993. He served as Associate Editor of Automatica (2000–2006), as Editor-in-Chief of the Journal of Mechanical Science and Technology (2008–2011), and is serving as the Editor-in-Chief of the International Journal of Control, Automation, and Systems. He was a past President of the Institute of Control, Robotics and Systems (ICROS), Korea, and is President of the Asian Control Association. He is an IEEE Fellow, a Fellow of the Korean Academy of Science and Technology, an ICROS Fellow, and a Member of the National Academy of Engineering of Korea. He has received many awards, including the Best Paper Award from the KFSTS of Korea (1999) and the Presidential Award of Korea (2007). His research interests include brain–computer interface, nonlinear systems theory, adaptive control, and distributed parameter systems.