Tunnel Prescribed Performance Control for Distributed Path Maneuvering of Multi-UAV Swarms via Distributed Neural Predictor

Di Wu, *Member, IEEE*, Yibo Zhang, *Member, IEEE*, Wentao Wu, *Student Member, IEEE*, Edmond Q. Wu, *Senior Member, IEEE*, Weidong Zhang, *Senior Member, IEEE*

***Abstract*—In this brief, we investigate the multiple parameterized paths-guided distributed maneuvering problem of a swarm of simplified unmanned aerial vehicles (UAVs) using a tunnel prescribed performance (TPP) strategy under directed communication. The primary focus of this paper lies in establishing a distributed TPP-based path maneuvering controller to achieve the desired cooperative performance. Firstly, a kinematic control law is designed by using the TPP strategy to limit the overshoot of the distributed path maneuvering error in transient and steady process. Secondly, an update law is developed for each path variable based on a control effort minimization method. Next, a total control law is designed by using a distributed neural predictor (DNP) at the kinetic level, where the DNP is constructed based on the information of neighbors to estimate uncertainties in the kinetics of UAVs. Then, via the Lyapunov analysis, practical distributed path maneuvering of multiple UAVs is achieved, ensuring the uniform ultimate bounded stability of the total closed-loop system through the proposed method. Finally, the effectiveness of the proposed approach for UAV swarms is validated via simulation results.**

***Index Terms*—Unmanned aerial vehicles (UAVs), path maneuvering, tunnel prescribed performance (TPP), distributed neural predictor (DNP).**

I. Introduction

RECENTLY, cooperative control has emerged as one of the most prominent research areas, given its wide applications in various engineering fields [1]–[3]. According to characteristics of guidance, cooperative control primarily focuses on target tracking [4], [5], trajectory tracking [6], [7], and path maneuvering/following [8]–[10]. Specially, path maneuvering-based cooperative control aims to coordinate the movement of all vehicles along several parameterized paths [8]–[10]. In [8], a path maneuvering control method is proposed and a control effort reducing scheme is designed for the path update. In [9], a single path-guided consensus maneuvering control method is designed for nonlinear systems based on the leader-following scheme. In [10], a modular distributed maneuvering controller is designed for nonlinear multi-agent systems based on neural predictors. The above results in [8]–[10] focus on numerical models and are naturally expanded to UAV swarms [11]. However, the above path maneuvering control methods consider ideal system performances. In practical applications, performance constraints cannot be neglected easily. These constraints may include overshoot and limitations of tracking error [11].

Funnel control is commonly employed to address the challenge of initial error convergence [1], [12]–[14]. As one of the important branches within funnel control, prescribed performance control (PPC) provides an efficient tool via combining error transformation and prescribed performance to enhance both stability and performance constraints [13]. Many PPC methods have been reported [13], [14], but these methods may face the loose of performance constraints due to the presence of distributed funnel boundaries and symmetric global performance functions. To overcome this issue, an improved tunnel prescribed performance (TPP) control strategy is proposed, which is introduced to not only have a concise form but also achieve smaller overshoot performance [15], [16]. Although some TPP-based control problems have been studied gradually, it should be noted that the TPP-based path maneuvering problem is still open.

Besides, in many path maneuvering methods, neural predictors are introduced as adaptive estimators, such as adaptive neural predictors [10], data-driven neural predictors [17], and high-order tuner-based neural predictors [18]. These neural predictors employ a local learning strategy, which may encounter limitations in generalization ability. Thus, there is a need to develop a new distributed learning-based neural predictor to enhance the performance of estimators.

This brief investigates the distributed path maneuvering of UAV swarms under the directed communication. Distributed path maneuvering finds applications in various scenarios. For example, envision a scenario where multiple UAVs are involved in a search within a predefined area delineated by several parameterized paths. Some of UAVs act as leaders, traversing along these parameterized paths, while the remaining UAVs follow the guidance of these leader UAVs. This challenge can be effectively tackled using distributed path maneuvering methods. In this paper, the kinetics of the UAV

This paper is partly supported by the National Key R&D Program of China (2022ZD0119903), the China Post-Doctoral Science Foundation (2022M722053), the Oceanic Interdisciplinary Program of Shanghai Jiao Tong University (SL2022PT112), the National Natural Science Foundation of China (52201369), the Shanghai Science and Technology Program (22015810300), "South China Sea Rising Star" Education Platform Foundation of Hainan Province (JYNHXX2023-17G). (*Corresponding Author: Yibo Zhang and Weidong Zhang*)

Di Wu, Yibo Zhang, Wentao Wu, Edmond Q. Wu and Weidong Zhang are with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China. Di Wu, Weidong Zhang are also with the School of Information and Communication Engineering, Hainan University, Haikou 570228, Hainan, China. Email: {sjtuwudi, zhang297, wentao-wu, edmondqwu, wdzhang}@sjtu.edu.cn.

is subject to uncertainties and external disturbances. To reduce the overshoot and cover uncertainties, we develop a distributed path maneuvering controller for the UAV swarm based on a tunnel prescribed performance (TPP) strategy and a distributed neural predictor (DNP). The key contributions of this concise study are outlined as follows.

1) In contrast to some of cooperative maneuvering control techniques proposed in [9], [10] under the ideal performance, we consider the prescribed performance of the distributed maneuvering error, and a distributed TPP-based path maneuvering controller is developed to ensure the prescribed performance.
2) In contrast to some of cooperative maneuvering control techniques in [10], [17]–[19] using local learning-based neural predictors, we construct a DNP in the proposed distributed TPP-based path maneuvering controller. The proposed DNP is under the distributed learning strategy. This implies that the UAV can update the weight using the information from its neighbors, thereby enhancing its generalization ability.

Notations: $\mathrm{sign}(\cdot)$ denotes the signum function; $|\cdot|$, $||\cdot||$, and $||\cdot||_{\mathrm{F}}$ correspond to the magnitude of a real number, the 2-norm of a vector, and the F-norm of a matrix, respectively; $\exp(\cdot)$ denotes the exponential function; $f^{\mathrm{inv}}(\cdot) \in \mathbb{R}^{n\times n}$ is the inverse of the function $f(\cdot)$; $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ represent the minimum eigenvalue and the maximum eigenvalue of matrix.

II. Preliminaries and Problem Formulation

A. Graph Theory

To describe the communication among multiple virtual leaders and UAVs, a directed graph is described as $\mathcal{G} = \{\mathcal{O}, \mathcal{E}\}$. $\mathcal{O} = \{\mathcal{O}^L, \mathcal{O}^F\}$ is the nodes set with $\mathcal{O}^L$ being the subset consisting of virtual leaders and $\mathcal{O}^F$ being the subset of UAVs. $\mathcal{E} = \{(i,j) \in \mathcal{O}\times\mathcal{O}\}$ denotes the set of edge. For the ith UAV, define its neighbor set as $\mathcal{N}_i = \{\mathcal{N}_i^L, \mathcal{N}_i^F\}$, where $\mathcal{N}_i^L = \{j \in \mathcal{O}^L | (i,j) \in \mathcal{E}\}$ and $\mathcal{N}_i^F = \{j \in \mathcal{O}^F | (i,j) \in \mathcal{E}\}$. An adjacency matrix associated with $\mathcal{G}$ is defined as $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$, where $a_{ij} = 1$ for $(j,i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Correspondingly, a degree matrix $\mathcal{D}$ connected with $\mathcal{G}$ is characterized as $\mathcal{D} = \mathrm{diag}\{d_i\} \in \mathbb{R}^{N\times N}$ with $d_i = \sum_{j=1}^{N} a_{i,j}$. Additionally, a Laplacian matrix associated with $\mathcal{G}$ is defined as $\mathcal{L} = \mathcal{D} - \mathcal{A}$.

B. Problem formulation

This paper investigates distributed path maneuvering of a swarm consisting of M UAVs and N-M virtual leaders, depicted in Fig. 1 . Inspired by [20], the model of the ith UAV in the swarm can be divided into two distinct subsystems: one is the outer-loop path maneuvering subsystem and the other one is the inner-loop kinetic control subsystem. We consider a simplified UAV model, such that attitude dynamics is ignored. The dynamics of the ith UAV can be represented by

$$\begin{cases} \dot{\boldsymbol{p}}_i(t) = \boldsymbol{v}_i(t), & i = 1,2,...,M \\ \dot{\boldsymbol{v}}_i(t) = \boldsymbol{f}_i(\boldsymbol{p}_i, \boldsymbol{v}_i, t) + \boldsymbol{u}_i(t) \end{cases} \tag{1}$$

where $\boldsymbol{f}_i(\boldsymbol{p}_i, \boldsymbol{v}_i, t) \in \mathbb{R}^3 := \boldsymbol{\beta}_{p_i}\boldsymbol{p}_i(t) + \boldsymbol{\beta}_{v_i}\boldsymbol{v}_i(t) + \boldsymbol{w}_{i,d}(t)$ denotes uncertain nonlinearities, $\beta_{p_i} \in \mathbb{R}^{3\times3}$ and $\beta_{v_i} \in \mathbb{R}^{3\times3}$ are two unknown parameter matrices, $\boldsymbol{p}_i(t) = [p_{i,x}, p_{i,y}, p_{i,z}]^{\mathrm{T}} \in \mathbb{R}^3$, $\boldsymbol{v}_i(t) = [v_{i,x}, v_{i,y}, v_{i,z}]^{\mathrm{T}} \in \mathbb{R}^3$ and $\boldsymbol{w}_{i,d}(t) \in \mathbb{R}^3$ represent the position, velocity and environmental disturbance, respectively. Additionally, $\boldsymbol{u}_i(t) \in \mathbb{R}^3$ represents the control input vector. For the convenience, $\boldsymbol{p}_i(t)$, $\boldsymbol{v}_i(t)$, $\boldsymbol{w}_{i,d}(t)$, $\boldsymbol{u}_i(t)$ and $\boldsymbol{f}_i(\boldsymbol{p}_i, \boldsymbol{v}_i, t)$ are abbreviated as $\boldsymbol{p}_i$, $\boldsymbol{v}_i$, $\boldsymbol{w}_{i,d}$, $\boldsymbol{u}_i$ and $\boldsymbol{f}_i(\cdot)$.

Virtual leaders move along parameterized paths $\boldsymbol{p}_{r,j}(\theta_j) \in \mathbb{R}^3$ with $j = M+1, ..., N$, where $\theta_j \in \mathbb{R}$ is the path variable for the jth virtual leader.

In this article, we design a distributed path maneuvering method of multi-UAV swarms, with primary objectives being as follows. 1) Geometric objective: Drive the position $\boldsymbol{p}_i$ of the ith UAV converges to a convex hull as $\|\boldsymbol{p}_i - Co(\boldsymbol{p}_{r,j}(\theta_j))\| \le \iota_g$ with $\iota_g \in \mathbb{R}^+$ being a residual error. 2) Dynamic objective: Drive all path variables synchronization as $\sum_{l=M+1}^{N} a_{j,l}(\theta_j - \theta_l) + a_{j,0}(\theta_j - \theta_0) = 0$ where $\dot{\theta}_0 = v_s$ denotes the speed assignment and can only be accessed to a few of leaders.

Assumption 1: The derivative of $\boldsymbol{p}_{r,j}(\theta_j)$ is bounded and adheres to the condition: $\|\dot{\boldsymbol{p}}_{r,j}(\theta_j)\| \le \bar{p}_r \in \mathbb{R}^+$.

Assumption 2: There is at least one virtual leader which has a path leading to each follower.

Remark 1: Assumption 1 ensures that the desired path is smooth bounded and derivable. The exact boundaries of $\boldsymbol{p}_{r,j}(\theta_j)$ do not need to be known because we do not use them in the controller. Assumption 2 is common for the design of the distributed controller. According to Lemma 1 in [20], if Assumption 2 holds true, it implies that within $\mathcal{G}$, there exists a spanning tree where the virtual leader serves as the root node.

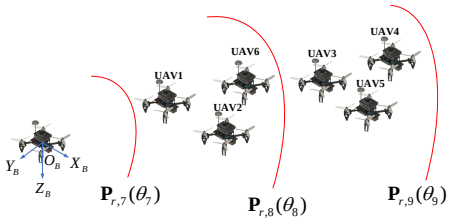

Fig. 1. Distributed path maneuvering of the swarm with multiple UAVs and virtual leaders.

III. Maneuvering Controller Design

A. TPP-based Kinematic Control Law

To achieve the distributed cooperation of the UAV swarm guided by multiple parameterized paths, we define the following distributed path maneuvering error $\boldsymbol{E}_{i,1} = [E_{i,x}, E_{i,y}, E_{i,z}]^{\mathrm{T}}$ for the ith UAV

$$\boldsymbol{E}_{i,1} = \sum_{j=1}^{M} a_{i,j}(\boldsymbol{p}_i - \boldsymbol{p}_j) + \sum_{j=M+1}^{N} a_{i,j}(\boldsymbol{p}_i - \boldsymbol{p}_{r,j}(\theta_j)). \tag{2}$$

Differentiating (2) along (1) , we have

$$\dot{\boldsymbol{E}}_{i,1} = d_i\boldsymbol{v}_i - \sum_{j=1}^{M} a_{i,j}\boldsymbol{v}_j - \sum_{j=M+1}^{N} a_{i,j}\boldsymbol{p}_{r,j}^{\theta_j}(\theta_j)\dot{\theta}_j \tag{3}$$

where $d_i = \sum_{j=1}^{N} a_{i,j}$. To enhance both transient and steady-state performance of UAVs, the distributed path maneuvering error is designed to meet the following TPP constraints

$$-\underline{E}_{i,k} \leq E_{i,k} \leq \overline{E}_{i,k} \quad k = x, y, z \tag{4}$$

where $\underline{E}_{i,k}$ and $\overline{E}_{i,k}$ are time-varying functions

$$\begin{cases} \overline{E}_{i,k} = (\overline{\xi}_{i,k} + \text{sign}(E_{ik}(0)))\rho_{i,k} - \rho_{ik,\infty}\text{sign}(E_{ik}(0)) \\ \underline{E}_{i,k} = (\underline{\xi}_{i,k} - \text{sign}(E_{ik}(0)))\rho_{i,k} + \rho_{ik,\infty}\text{sign}(E_{ik}(0)) \end{cases} \tag{5}$$

with $\overline{\xi}_{i,k}$, $\underline{\xi}_{i,k} \in [0, 1]$, $\rho_{i,k} = (\rho_{ik,0} - \rho_{ik,\infty})\exp(-\mathcal{U}_{ik}t) + \rho_{ik,\infty}$, $\rho_{ik,0} > \rho_{ik,\infty} \in \mathbb{R}^+$ and $\mathcal{U}_{ik} \in \mathbb{R}^+$.

Then, (4) is further put into

$$E_{i,k} = \frac{\overline{E}_{i,k} + \underline{E}_{i,k}}{2} G_{i,k}(\delta_{i,k}) + \frac{\overline{E}_{i,k} - \underline{E}_{i,k}}{2} \tag{6}$$

where $G_{i,k}(\delta_{i,k}) : (-\infty, \infty) \to (-1, 1)$ represents a strictly increasing transformation function, while $\delta_{i,k}$ corresponds to an unconstrained transformed error equivalent to $E_{i,k}$. In this context, the chosen transformation function $G_{i,k}(\delta_{i,k})$ is expressed as $G_{i,k}(\delta_{i,k}) = (\exp(\delta_{i,k}) - \exp(-\delta_{i,k}))/(\exp(\delta_{i,k}) + \exp(-\delta_{i,k}))$. Through the inverse function of $G_{i,k}(\delta_{i,k})$, the transformed error $\delta_{i,k}$ can be obtained as follows

$$\delta_{i,k} = G_{i,k}^{-\text{inv}}(E_{i,k}, \overline{E}_{i,k}, \underline{E}_{i,k}) = \frac{1}{2} \ln \frac{1 + G_{i,k}}{1 - G_{i,k}} \tag{7}$$

where $G_{i,k} = (2E_{i,k} - \overline{E}_{i,k} + \underline{E}_{i,k})/(\overline{E}_{i,k} + \underline{E}_{i,k})$. The time derivative of $\delta_{i,k}$ is provided as follows

$$\dot{\delta}_{i,k} = \frac{\partial \delta_{i,k}}{\partial G_{i,k}} \left(\frac{\partial G_{i,k}}{\partial E_{i,k}} \dot{E}_{i,k} + \frac{\partial G_{i,k}}{\partial \overline{E}_{i,k}} \dot{\overline{E}}_{i,k} + \frac{\partial G_{i,k}}{\partial \underline{E}_{i,k}} \dot{\underline{E}}_{i,k} \right), \tag{8}$$

where $\partial \delta_{i,k}/\partial G_{i,k} = 1/(1 - G_{i,k}^2)$, $\partial G_{i,k}/\partial E_{i,k} = 2/(\overline{E}_{i,k} + \underline{E}_{i,k})$, $\partial G_{i,k}/\partial \overline{E}_{i,k} = -2(\underline{E}_{i,k} + E_{i,k})/(\overline{E}_{i,k} + \underline{E}_{i,k})^2$, $\partial G_{i,k}/\partial \underline{E}_{i,k} = 2(\overline{E}_{i,k} - E_{i,k})/(\overline{E}_{i,k} + \underline{E}_{i,k})^2$.

Define $\boldsymbol{\delta}_i = [\delta_{i,x}, \delta_{i,y}, \delta_{i,z}]^{\mathrm{T}}$, $\overline{\boldsymbol{E}}_{i,1} = [\overline{E}_{i,x}, \overline{E}_{i,y}, \overline{E}_{i,z}]^{\mathrm{T}}$, and $\underline{\boldsymbol{E}}_{i,1} = [\underline{E}_{i,x}, \underline{E}_{i,y}, \underline{E}_{i,z}]^{\mathrm{T}}$. Then, (8) can be transformed into the following form

$$\dot{\boldsymbol{\delta}}_i = \boldsymbol{\mathcal{H}}_{E_{i,1}} \dot{\boldsymbol{E}}_{i,1} + \boldsymbol{\mathcal{H}}_{\overline{E}_{i,1}} \dot{\overline{\boldsymbol{E}}}_{i,1} + \boldsymbol{\mathcal{H}}_{\underline{E}_{i,1}} \dot{\underline{\boldsymbol{E}}}_{i,1} \tag{9}$$

where $\boldsymbol{\mathcal{H}}_{E_{i,1}} = \text{diag}\{\mathcal{H}_{E_{i,k}}\} \in \mathbb{R}^{3\times3}$ with $\mathcal{H}_{E_{i,k}} = (\partial \delta_{i,k}/\partial G_{i,k}) \times (\partial G_{i,k}/\partial E_{i,k})$; $\boldsymbol{\mathcal{H}}_{\overline{E}_{i,1}} = \text{diag}\{\mathcal{H}_{\overline{E}_{i,k}}\} \in \mathbb{R}^{3\times3}$ with $\mathcal{H}_{\overline{E}_{i,k}} = (\partial \delta_{i,k}/\partial G_{i,k}) \times (\partial G_{i,k}/\partial \overline{E}_{i,k})$; $\boldsymbol{\mathcal{H}}_{\underline{E}_{i,1}} = \text{diag}\{\mathcal{H}_{\underline{E}_{i,k}}\} \in \mathbb{R}^{3\times3}$ with $\mathcal{H}_{\underline{E}_{i,k}} = (\partial \delta_{i,k}/\partial G_{i,k}) \times (\partial G_{i,k}/\partial \underline{E}_{i,k})$.

Substituting (3) into (9) yields

$$\begin{aligned} \dot{\boldsymbol{\delta}}_i = \boldsymbol{\mathcal{H}}_{E_{i,1}} \Big(d_i \boldsymbol{v}_i - \sum_{j=1}^{M} a_{i,j} \boldsymbol{v}_j - \sum_{j=M+1}^{N} a_{i,j} \boldsymbol{p}_{r,j}^{\theta_j}(\theta_j) \dot{\theta}_j \Big) \\ + \boldsymbol{\mathcal{H}}_{\overline{E}_{i,1}} \dot{\overline{\boldsymbol{E}}}_{i,1} + \boldsymbol{\mathcal{H}}_{\underline{E}_{i,1}} \dot{\underline{\boldsymbol{E}}}_{i,1}. \end{aligned} \tag{10}$$

Define $\tilde{\boldsymbol{v}}_j = \hat{\boldsymbol{v}}_j - \boldsymbol{v}_j$. To stabilize error dynamics (10), we construct a TPP-based kinematic control law $\boldsymbol{\alpha}_i$ as follows

$$\begin{aligned} \boldsymbol{\alpha}_i = \frac{1}{d_i} \Big(-\boldsymbol{\mathcal{H}}_{E_{i,1}}^{-1} (\boldsymbol{K}_{i,1} \boldsymbol{\delta}_i + \boldsymbol{\mathcal{H}}_{\overline{E}_{i,1}} \dot{\overline{\boldsymbol{E}}}_{i,1} + \boldsymbol{\mathcal{H}}_{\underline{E}_{i,1}} \dot{\underline{\boldsymbol{E}}}_{i,1}) \\ + \sum_{j=1}^{M} a_{i,j} \hat{\boldsymbol{v}}_j + \sum_{j=M+1}^{N} a_{i,j} \boldsymbol{p}_{r,j}^{\theta_j}(\theta_j) \dot{\theta}_j \Big) \end{aligned} \tag{11}$$

where $\boldsymbol{K}_{i,1} \in \mathbb{R}^{3\times3}$ is a positive definite control gain matrix.

Substituting (11) into (10), $\dot{\boldsymbol{\delta}}_i$ is transformed into the following form: $\dot{\boldsymbol{\delta}}_i = \boldsymbol{\mathcal{H}}_{E_{i,1}} \sum_{j=1}^{M} a_{i,j} \tilde{\boldsymbol{v}}_j - \boldsymbol{K}_{i,1} \boldsymbol{\delta}_i$.

B. Path update law

Unlike trajectory tracking, path maneuvering involves an additional control degree of freedom. Hence, a feedback law is introduced for each path variable to achieve the update and synchronization among multiple virtual leaders.

At first, the dynamics of the jth path variable is defined as $\ddot{\theta}_j = \omega_j$. A distributed parameter error is designed as $z_j = \sum_{j=M+1}^{N} a_{j,l}(\dot{\theta}_j - \dot{\theta}_l) + a_{j,0}(\dot{\theta}_j - v_s)$.

Then, a path update law can be designed with the ability to reduce the control effort as follows [8]

$$\omega_j = -\frac{(c k_d z_j + \boldsymbol{E}_{i,1}^{\mathrm{T}} \boldsymbol{E}_{i,1} (\eta \boldsymbol{p}_{r,j}^{\theta_j \mathrm{T}}(\theta_j) \boldsymbol{E}_{i,1}))}{c + ||\boldsymbol{p}_{r,j}^{\theta_j}(\theta_j)||^2 \boldsymbol{E}_{i,1}^{\mathrm{T}} \boldsymbol{E}_{i,1}} \tag{12}$$

where $k_d \in \mathbb{R}^+$ is a gain, and $c \in \mathbb{R}^+$ and $\eta \in \mathbb{R}^+$ are two tuning parameters.

Remark 2: In order to ensure synchronization among the virtual leaders, the path update law ω_j, as a to-be designed tuning term, is designed to fulfill the dynamic objective.

C. DNP-based Kinetic Control Law

Define a velocity tracking error at the kinetic level: $\boldsymbol{E}_{i,2} = \boldsymbol{v}_i - \boldsymbol{\alpha}_i$. And the dynamics of $\boldsymbol{E}_{i,2}$ can be obtained along (1) as: $\dot{\boldsymbol{E}}_{i,2} = \boldsymbol{u}_i + \boldsymbol{f}_i(\cdot) - \dot{\boldsymbol{\alpha}}_i$. Then, an ideal kinetic control law $\boldsymbol{u}_i$ is proposed as: $\boldsymbol{u}_i = -\boldsymbol{K}_{i,2} \boldsymbol{E}_{i,2} - \boldsymbol{f}_i(\cdot) + \dot{\boldsymbol{\alpha}}_i$. Where $\boldsymbol{K}_{i,2} \in \mathbb{R}^{3\times3}$ is a positive definite control gain matrix. Some adaptive approximators can be introduced to identify $\boldsymbol{f}_i(\cdot)$. We can use the following neural network

$$\boldsymbol{f}_i(\cdot) = \boldsymbol{W}_i^{\mathrm{T}} \boldsymbol{\zeta}_i(\boldsymbol{\chi}_i) + \boldsymbol{\varepsilon}_i(\boldsymbol{\chi}_i) \tag{13}$$

where $\boldsymbol{W} \in \mathbb{R}^{p\times3}$ represents the optional weight vector with p being the number of neurons, $\boldsymbol{\zeta}_i(\boldsymbol{\chi}_i) \in \mathbb{R}^p$ is a known smooth activation function, $\boldsymbol{\chi}_i = [\boldsymbol{E}_{i,2}^{\mathrm{T}}(t), \boldsymbol{E}_{i,2}^{\mathrm{T}}(t - t_d), \boldsymbol{u}_i^{\mathrm{T}}(t), \dot{\boldsymbol{\alpha}}_i^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{12}$ denotes the input vector, and $\boldsymbol{\varepsilon}_i \in \mathbb{R}^3$ denotes an time-varying approximation error satisfying $\|\boldsymbol{\varepsilon}_i(\boldsymbol{\chi}_i)\| \leq \varepsilon_i^*$ with $\varepsilon_i^* \in \mathbb{R}^+$.

An actual kinetic control law $\boldsymbol{u}_i$ is proposed as follows

$$\boldsymbol{u}_i = -\boldsymbol{K}_{i,2} \boldsymbol{E}_{i,2} - \hat{\boldsymbol{W}}_i^{\mathrm{T}} \boldsymbol{\zeta}_i(\boldsymbol{\chi}_i) + \dot{\boldsymbol{\alpha}}_i. \tag{14}$$

In the existing path maneuvering methods [10], [17], [18], the neural predictors are utilized to tune the weights of neural network. These neural predictors rely on the local learning strategy. To further enhance the ability of the neural predictor [19], we design the following DNP law by using a distributed learning strategy [21]

$$\begin{cases} \dot{\hat{\boldsymbol{v}}}_i = \hat{\boldsymbol{W}}_i^{\mathrm{T}} \boldsymbol{\zeta}_i(\boldsymbol{\chi}_i) + \boldsymbol{u}_i - (\boldsymbol{K}_{i,2} + \boldsymbol{\kappa}_i) \tilde{\boldsymbol{v}}_i \\ \dot{\hat{\boldsymbol{W}}}_i = \Gamma_i (-\boldsymbol{\zeta}_i(\boldsymbol{\chi}_i) \tilde{\boldsymbol{v}}_i^{\mathrm{T}} - \sigma_i \hat{\boldsymbol{W}}_i) - K_{W_i} \sum_{j=1}^{M} a_{i,j} (\hat{\boldsymbol{W}}_i - \hat{\boldsymbol{W}}_j) \end{cases} \tag{15}$$

where $\Gamma_i, K_{W_i} \in \mathbb{R}^+$ are adaptation gains, $\boldsymbol{\kappa}_i \in \mathbb{R}^{3\times3}$ is an additional positive definite tuning parameter matrix to shape the transient learning behavior of neural network, and $\sigma_i \in (0, 1)$ is a modification factor.

IV. Main Result

Recall the error dynamics of the resulting distributed path maneuvering closed-loop system as follows:

$$\begin{cases}\dot{\boldsymbol{\delta}}_i=\boldsymbol{\mathcal{H}}_{E_{i,1}}\sum_{j=1}^{M}a_{i,j}\tilde{\boldsymbol{v}}_j-\boldsymbol{K}_{i,1}\boldsymbol{\delta}_i\\ \dot{\hat{\boldsymbol{E}}}_{i,2}=-\boldsymbol{K}_{i,2}\hat{\boldsymbol{E}}_{i,2}-\boldsymbol{\kappa}_i\tilde{\boldsymbol{v}}_i\\ \dot{\tilde{\boldsymbol{v}}}_i=-(\boldsymbol{K}_{i,2}+\boldsymbol{\kappa}_i)\tilde{\boldsymbol{v}}_i+\hat{\mathbf{W}}_i^{\mathrm{T}}\boldsymbol{\zeta}_i(\boldsymbol{\chi}_i)-\boldsymbol{\varepsilon}_i\\ \dot{\tilde{\boldsymbol{W}}}_i=\Gamma_i(-\boldsymbol{\zeta}_i(\boldsymbol{\chi}_i)\tilde{\boldsymbol{v}}_i^{\mathrm{T}}-\sigma_i\hat{\boldsymbol{W}}_i)-K_{W_i}\sum_{j=1}^{M}a_{i,j}(\hat{\boldsymbol{W}}_i-\hat{\boldsymbol{W}}_j).\end{cases}\tag{16}$$

The main result can be summarized as a theorem as below.

Theorem 1: Under Assumptions 1 & 2, when the distributed path maneuvering controller is chosen as the transformed error (7), the kinematic control law (11), the path update law (12), the kinetic control law (14), and the DNP (15), all error signals within the total closed-loop system are uniformly ultimately bounded (UUB), and practical distributed path maneuvering of the multi-UAV swarm can be achieved.

Proof: Define a Lyapunov function as follows

$$V=\frac{1}{2}\sum_{i=1}^{M}\left(\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{\delta}_i+\hat{\boldsymbol{E}}_{i,2}^{\mathrm{T}}\hat{\boldsymbol{E}}_{i,2}+\tilde{\boldsymbol{v}}_i^{\mathrm{T}}\tilde{\boldsymbol{v}}_i+\tilde{\boldsymbol{W}}_i^{\mathrm{T}}\Gamma_i^{-1}\tilde{\boldsymbol{W}}_i\right).$$

By taking the time derivatives of V using (16), we obtain the following expression

$$\begin{aligned}\dot{V}&=\sum_{i=1}^{M}\left(\boldsymbol{\delta}_i^{\mathrm{T}}\dot{\boldsymbol{\delta}}_i+\hat{\boldsymbol{E}}_{i,2}^{\mathrm{T}}\dot{\hat{\boldsymbol{E}}}_{i,2}+\tilde{\boldsymbol{v}}_i^{\mathrm{T}}\dot{\tilde{\boldsymbol{v}}}_i+\tilde{\boldsymbol{W}}_i^{\mathrm{T}}\Gamma_i^{-1}\dot{\tilde{\boldsymbol{W}}}_i\right)\\ &\le\sum_{i=1}^{M}\Big(-(\boldsymbol{K}_{i,2}+\boldsymbol{\kappa}_i)\tilde{\boldsymbol{v}}_i^{\mathrm{T}}\tilde{\boldsymbol{v}}_i-\sigma_i\tilde{\boldsymbol{W}}_i^{\mathrm{T}}\hat{\boldsymbol{W}}_i-\tilde{\boldsymbol{v}}_i^{\mathrm{T}}\boldsymbol{\varepsilon}_i\\ &\quad-\boldsymbol{K}_{i,1}\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{\delta}_i-\boldsymbol{K}_{i,2}\hat{\boldsymbol{E}}_{i,2}^{\mathrm{T}}\hat{\boldsymbol{E}}_{i,2}-\hat{\boldsymbol{E}}_{i,2}^{\mathrm{T}}\boldsymbol{\kappa}_i\tilde{\boldsymbol{v}}_i+\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{\mathcal{H}}_{E_{i,1}}\sum_{j=1}^{M}a_{i,j}\tilde{\boldsymbol{v}}_j\Big)\end{aligned}\tag{17}$$

By employing Young's inequality, the subsequent inequalities are satisfied.

$$\begin{cases}-\sum_{i=1}^{M}\sigma_i\tilde{\boldsymbol{W}}_i^{\mathrm{T}}\hat{\boldsymbol{W}}_i\le-\frac{\underline{\sigma}}{2}||\tilde{\boldsymbol{W}}||_{\mathrm{F}}^2+\frac{\overline{\sigma}}{2}||\boldsymbol{W}||_{\mathrm{F}}^2\\ -\sum_{i=1}^{M}\tilde{\boldsymbol{v}}_i^{\mathrm{T}}\boldsymbol{\varepsilon}_i\le\frac{||\tilde{\boldsymbol{v}}||^2}{2}+\frac{||\boldsymbol{\varepsilon}||^2}{2}\\ -\sum_{i=1}^{M}\hat{\boldsymbol{E}}_{i,2}^{\mathrm{T}}\boldsymbol{\kappa}_i\tilde{\boldsymbol{v}}_i\le\frac{\lambda_{\max}(\boldsymbol{\kappa})}{2}(||\hat{\boldsymbol{E}}_2||^2+||\tilde{\boldsymbol{v}}||^2)\\ \sum_{i=1}^{M}\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{\mathcal{H}}_{E_{i,1}}\sum_{j=1}^{M}a_{i,j}\tilde{\boldsymbol{v}}_j=\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\mathcal{H}}_E(\boldsymbol{\mathcal{A}}\otimes\boldsymbol{I}_3)\tilde{\boldsymbol{v}}\\ \qquad\le\frac{\lambda_{\max}(\boldsymbol{\mathcal{H}}_E)\lambda_{\max}(\boldsymbol{\mathcal{A}})}{2}(||\boldsymbol{\delta}||^2+||\tilde{\boldsymbol{v}}||^2)\end{cases}\tag{18}$$

where $\tilde{\boldsymbol{v}}=[\tilde{\boldsymbol{v}}_1^{\mathrm{T}},\tilde{\boldsymbol{v}}_2^{\mathrm{T}},\cdots,\tilde{\boldsymbol{v}}_M^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{\varepsilon}=[\boldsymbol{\varepsilon}_1^{\mathrm{T}},\boldsymbol{\varepsilon}_2^{\mathrm{T}},\cdots,\boldsymbol{\varepsilon}_M^{\mathrm{T}}]^{\mathrm{T}}$, $\tilde{\boldsymbol{W}}=[\tilde{\boldsymbol{W}}_1^{\mathrm{T}},\tilde{\boldsymbol{W}}_2^{\mathrm{T}},\cdots,\tilde{\boldsymbol{W}}_M^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{W}=[\boldsymbol{W}_1^{\mathrm{T}},\boldsymbol{W}_2^{\mathrm{T}},\cdots,\boldsymbol{W}_M^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{\delta}=[\boldsymbol{\delta}_1^{\mathrm{T}},\boldsymbol{\delta}_2^{\mathrm{T}},\cdots,\boldsymbol{\delta}_M^{\mathrm{T}}]^{\mathrm{T}}$, $\hat{\boldsymbol{E}}_2=[\hat{\boldsymbol{E}}_{1,2}^{\mathrm{T}},\hat{\boldsymbol{E}}_{2,2}^{\mathrm{T}},\cdots,\hat{\boldsymbol{E}}_{M,2}^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{K}_1=\mathrm{diag}\{\boldsymbol{K}_{1,1},\boldsymbol{K}_{2,1},\cdots,\boldsymbol{K}_{M,1}\}$, $\boldsymbol{\kappa}=\mathrm{diag}\{\boldsymbol{\kappa}_1,\boldsymbol{\kappa}_2,\cdots,\boldsymbol{\kappa}_M\}$, $\boldsymbol{\mathcal{H}}_E=\mathrm{diag}\{\boldsymbol{\mathcal{H}}_{E_{i,1}},\boldsymbol{\mathcal{H}}_{E_{2,1}},\cdots,\boldsymbol{\mathcal{H}}_{E_{M,1}}\}$, $\boldsymbol{K}_2=\mathrm{diag}\{\boldsymbol{K}_{1,2},\boldsymbol{K}_{2,2},\cdots,\boldsymbol{K}_{M,2}\}$, $\underline{\sigma}=\min\{\sigma_1,\sigma_2,\cdots,\sigma_m\}$, and $\overline{\sigma}=\max\{\sigma_1,\sigma_2,\cdots,\sigma_m\}$. We rewrite $\dot{V}$ as follows

$$\begin{aligned}\dot{V}\le&-\left(\lambda_{\min}(\boldsymbol{K}_2)-\lambda_{\max}(\boldsymbol{\kappa})-\frac{\lambda_{\max}(\boldsymbol{\mathcal{H}}_E)\lambda_{\max}(\boldsymbol{\mathcal{A}})}{2}\right)||\tilde{\boldsymbol{v}}||^2\\ &-\frac{\underline{\sigma}}{2}||\tilde{\boldsymbol{W}}||_{\mathrm{F}}^2-\left(\lambda_{\min}(\boldsymbol{K}_1)-\frac{\lambda_{\max}(\boldsymbol{\mathcal{H}}_E)\lambda_{\max}(\boldsymbol{\mathcal{A}})}{2}\right)||\boldsymbol{\delta}||^2\\ &-\left(\lambda_{\min}(\boldsymbol{K}_2)-\frac{\lambda_{\max}(\boldsymbol{\kappa})}{2}\right)||\hat{\boldsymbol{E}}_2||^2+\frac{\overline{\sigma}}{2}||\boldsymbol{W}||_{\mathrm{F}}^2+||\boldsymbol{\varepsilon}||^2.\end{aligned}$$

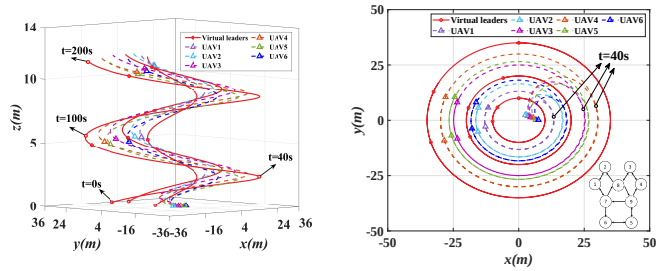

Fig. 2. Trajectories of 6 UAVs based on TPP and DNP.

Letting

$$\begin{cases}h_1=\lambda_{\min}(\boldsymbol{K}_2)-\lambda_{\max}(\boldsymbol{\kappa})-\frac{\lambda_{\max}(\boldsymbol{\mathcal{H}}_E)\lambda_{\max}(\boldsymbol{\mathcal{A}})}{2}>0\\ h_2=\lambda_{\min}(\boldsymbol{K}_1)-\frac{\lambda_{\max}(\boldsymbol{\mathcal{H}}_E)\lambda_{\max}(\boldsymbol{\mathcal{A}})}{2}>0\\ h_3=\lambda_{\min}(\boldsymbol{K}_2)-\frac{\lambda_{\max}(\boldsymbol{\kappa})}{2}>0\\ \varpi=\frac{\overline{\sigma}}{2}||\boldsymbol{W}||_{\mathrm{F}}^2+||\boldsymbol{\varepsilon}||^2,\end{cases}\tag{19}$$

it follows that $\dot{V}\le-hV+\varpi$ with $h=\min\{2h_1,\underline{\sigma}\Gamma_i,2h_2,2h_3\}$. Consequently, we can deduce that all error signals within the closed-loop system exhibit UUB.

Define $\boldsymbol{p}_r=[\boldsymbol{p}_{r,M+1}^{\mathrm{T}}(\theta_{M+1}),...,\boldsymbol{p}_{r,N}^{\mathrm{T}}(\theta_N)]^{\mathrm{T}}$ and $\boldsymbol{p}=[\boldsymbol{p}_1^{\mathrm{T}},...,\boldsymbol{p}_M^{\mathrm{T}}]^{\mathrm{T}}$. It follows that $L_1\boldsymbol{p}-L_2\boldsymbol{p}_r=\boldsymbol{\delta}$. According to *Lemma 2.3* in [22], we have $\|\boldsymbol{p}-L_1^{-1}L_2\boldsymbol{p}_r\|\le\|\boldsymbol{\delta}\|/\lambda_{\min}(L_1)$. Therefore, the practical distributed path maneuvering is achieved. ■

Remark 3. Another way to improve control performance is accelerating convergence rates. Using (2) in [23], we can design a prescribed-time TPP strategy via modifying the distributed path maneuvering error $\boldsymbol{E}_{i,1}$. Based on the results in [24], the proposed controller can be extended to an observer-based prescribed-time control case.

V. Simulation Results

In this section, we select the simulation parameters as follows: $\mathcal{U}_{ik}=[0.1,0.1,0.6]^{\mathrm{T}}$, $k_d=15$, $c=5.5\times10^3$, $\eta=5$, $\Gamma_i=4.0\times10^4$, $K_{W_i}=0.2$, $\boldsymbol{\kappa}_i=198$, $\sigma=0.0001$, $\boldsymbol{K}_{i,1}=\mathrm{diag}\{0.4,0.4,0.2\}$, $\boldsymbol{K}_{i,2}=\mathrm{diag}\{2,2,2\}$, $\boldsymbol{p}_{r,1}(\theta_1)=[10\sin(0.05\theta_1),10\cos(0.05\theta_1),0.05\theta_1]^{\mathrm{T}}$, $\boldsymbol{p}_{r,2}(\theta_2)=[20\sin(0.05\theta_2-0.3),20\cos(0.05\theta_2-0.3),0.05\theta_2+0.3]^{\mathrm{T}}$, $\boldsymbol{p}_{r,3}(\theta_3)=[35\sin(0.05\theta_3),35\cos(0.05\theta_3),0.05\theta_3]^{\mathrm{T}}$, and $\rho_0=[5,8,3;8,15,3;8,15,3;5,30,3;5,30,3;8,10,3]$.

Fig. 2 illustrates how the followers are able to track the virtual leaders. Fig. 3 demonstrates the updating rate of path parameters for the virtual leaders. Fig. 4 shows the control inputs of 6 UAVs. Taking UAV1 of the swarm as an example, as shown in Fig. 5, compared with PPC constraints in [13], the proposed TPP constraints have a more tight space and the tracking errors of the followers converge within a prescribed tight set with minimal overshoot using TPP. Furthermore, Fig. 6 presents a comparison between DNP and neural predictors in [19], clearly highlighting the superior performance of D-NP. Subsequently, the enhanced performance of the proposed method is evident from simulation results.

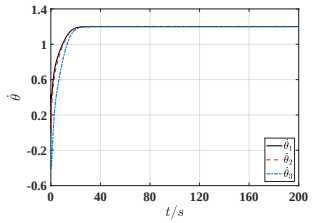

Fig. 3. The updating rate.

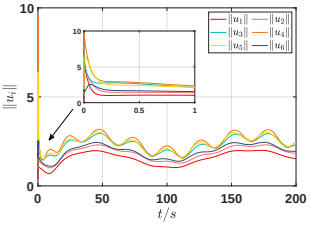

Fig. 4. Control inputs of 6 UAVs

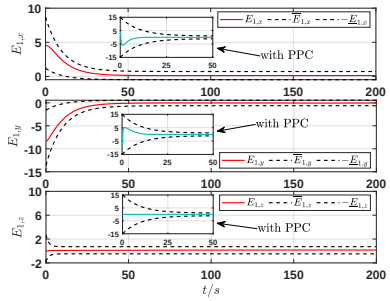

Fig. 5. The comparison between TPP and PPC in [13] for UAV1.

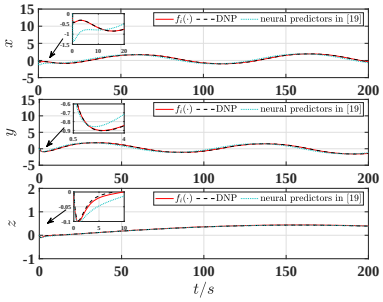

Fig. 6. The comparison between DNP and neural predictors in [19] for UAV1.

VI. Conclusion

This paper tackles the problem of TPP-based distributed path maneuvering for multi-UAV swarms in the presence of uncertain nonlinearities and external disturbances. Firstly, in the proposed kinematic control law, a TPP strategy is utilized to achieve prescribed performances by driving path maneuvering errors into a prescribed tight set. Secondly, a path update law is designed to achieve path updates and synchronization among path variables. Then, in the kinetic control law, the DNP is developed for facilitating the identification of unknown nonlinearities in individual UAVs through the distributed learning strategy. At last, error signals are demonstrated to be UUB and practical distributed path maneuvering is achieved. The simulation results demonstrate the effectiveness of the proposed distributed path maneuvering method for multi-UAV swarms. In upcoming researches, we intend to incorporate the attitude of UAV into our modeling and conduct experiments on its rigid body.

References

[1] X. Min, S. Baldi, W. Yu, and J. Cao, "Low-complexity control with funnel performance for uncertain nonlinear multi-agent systems," *IEEE Trans. Automat. Contr.*, pp. 1–8, 2023.

[2] H. Liu, Y. Wang, and F. L. Lewis, "Robust distributed formation controller design for a group of unmanned underwater vehicles," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 2, pp. 1215–1223, 2021.

[3] Z. Feng, M. Huang, D. Wu, E. Q. Wu, and C. Yuen, "Multi-agent reinforcement learning with policy clipping and average evaluation for UAV-assisted communication markov game," *IEEE trans. Intell. Transp. Syst.*, pp. 1–13, 2023.

[4] S. Wang, F. Jiang, B. Zhang, R. Ma, and Q. Hao, "Development of UAV-based target tracking and recognition systems," *IEEE trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3409–3422, 2020.

[5] Z. Xia, J. Du, J. Wang, C. Jiang, Y. Ren, G. Li, and Z. Han, "Multi-agent reinforcement learning aided intelligent UAV swarm for target tracking," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 931–945, 2022.

[6] Z. Wang, X. Zhou, C. Xu, J. Chu, and F. Gao, "Alternating minimization based trajectory generation for quadrotor aggressive flight," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4836–4843, 2020.

[7] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1992–2009, 2021.

[8] D. B. Dacic, M. V. Subbotin, and P. V. Kokotovic, "Control effort reduction in tracking feedback laws," *IEEE Trans. Automat. Contr.*, vol. 51, no. 11, pp. 1831–1837, 2006.

[9] Y. Zhang, D. Wang, and Z. Peng, "Consensus maneuvering for a class of nonlinear multivehicle systems in strict-feedback form," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1759–1767, 2019.

[10] Y. Zhang, D. Wang, Z. Peng, and T. Li, "Distributed containment maneuvering of uncertain multiagent systems in MIMO strict-feedback form," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 2, pp. 1354–1364, 2021.

[11] X. Shao, H. Liu, W. Zhang, J. Zhao, and Q. Zhang, "Path driven formation-containment control of multiple UAVs: A path-following framework," *Aerosp. Sci. Technol.*, vol. 135, p. 108168, 2023.

[12] X. Min, S. Baldi, and W. Yu, "Funnel-based asymptotic control of leader-follower nonholonomic robots subject to formation constraints," *IEEE Trans. Control. Netw. Syst.*, vol. 10, no. 3, pp. 1313–1325, 2023.

[13] Z. Jia, Z. Hu, and W. Zhang, "Adaptive output-feedback control with prescribed performance for trajectory tracking of underactuated surface vessels," *ISA Trans.*, vol. 95, pp. 18–26, 2019.

[14] H. Wang, M. Li, C. Zhang, and X. Shao, "Event-based prescribed performance control for dynamic positioning vessels," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 68, no. 7, pp. 2548–2552, 2021.

[15] W. Wu, Y. Zhang, W. Zhang, and W. Xie, "Distributed finite-time performance-prescribed time-varying formation control of autonomous surface vehicles with saturated inputs," *Ocean Eng.*, vol. 266, p. 112866, 2022.

[16] R. Ji, B. Yang, J. Ma, and S. S. Ge, "Saturation-tolerant prescribed control for a class of MIMO nonlinear systems," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13 012–13 026, 2022.

[17] L. Liu, D. Wang, Z. Peng, and Q. Han, "Distributed path following of multiple under-actuated autonomous surface vehicles based on data-driven neural predictors via integral concurrent learning," *IEEE Trans. Neural Netw. Learn Syst.*, vol. 32, no. 12, pp. 5334–5344, 2021.

[18] Y. Zhang, W. Wu, and W. Zhang, "Noncooperative game-based cooperative maneuvering of intelligent surface vehicles via accelerated learning-based neural predictors," *IEEE Trans. Veh. Technol.*, vol. 8, no. 3, pp. 2212–2221, 2023.

[19] Z. Peng, D. Wang, and J. Wang, "Predictor-based neural dynamic surface control for uncertain nonlinear systems in strict-feedback form," *IEEE Trans. Neural Netw. Learn Syst.*, vol. 28, no. 9, pp. 2156–2167, 2017.

[20] X. Dong, Y. Li, C. Lu, G. Hu, Q. Li, and Z. Ren, "Time-varying formation tracking for UAV swarm systems with switching directed topologies," *IEEE Trans. Neural Netw. Learn Syst.*, vol. 30, no. 12, pp. 3674–3685, 2019.

[21] W. Chen, S. Hua, and H. Zhang, "Consensus-based distributed cooperative learning from closed-loop neural control systems," *IEEE Trans. Neural Netw. Learn Syst.*, vol. 26, no. 2, pp. 331–345, 2015.

[22] J. Mei, W. Ren, and G. Ma, "Distributed containment control for lagrangian networks with parametric uncertainties under a directed graph," *Automatica*, vol. 48, no. 4, pp. 653–659, 2012.

[23] Y. Zhou, Y. Liu, Y. Zhao, M. Cao, and G. Chen, "Fully distributed prescribed-time bipartite synchronization of general linear systems: An adaptive gain scheduling strategy," *Automatica*, vol. 161, p. 111459, 2024.

[24] B. Zhou, Z. Zhang, and W. Michiels, "Functional and dual observer based prescribed-time control of linear systems by periodic delayed feedback," *Automatica*, vol. 159, p. 111406, 2024.