DSC-based Parallel Control for Consensus Maneuvering of Multi-agent Systems subject to Unmatched Uncertainties based on Distributed Nash Equilibrium Seeking

1st Yibo Zhang
Department of Automation
Shanghai Jiao Tong University
Shanghai, China
zhang297@sjtu.edu.cn

2nd Wentao Wu
Department of Automation
Shanghai Jiao Tong University
Shanghai, China
wentao-wu@sjtu.edu.cn

3rd Di Wu*
Department of Automation
Shanghai Jiao Tong University
Shanghai, China
hainuwudi@163.com

4th Yuanhui Wang
College of Intelligent Systems Science and Engineering
Harbin Engineering University
Harbin, China
wangyuanhui@hrbeu.edu.cn

5th Weidong Zhang*
Department of Automation
Shanghai Jiao Tong University
Shanghai, China
wdzhang@sjtu.edu.cn

2022 China Automation Congress (CAC) | 978-1-6654-6533-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/CAC57257.2022.10056005

***Abstract*—We consider parameterized virtual leader-guided distributed consensus maneuvering of uncertain multi-agent systems herein. The model of the agent is in the strict-feedback form and subject to unmatched uncertainties. In comparison with existing results, we try to research distributed consensus maneuvering from a distributed game perspective. Inspired by distributed Nash equilibrium seeking approach and cooperative learning strategy, a parallel control method is developed for consensus maneuvering, including cooperative learning-based neural predictors, dynamic surface control(DSC)-based parallel control law, and parameter update law for virtual leader. It is proved that the total closed-loop system is input-to-state stable, and the outputs of all agents converge to the Nash equilibrium as far as possible. The main results are demonstrated by theoretical analysis and a simulation example.**

***Index Terms*—Dynamic surface control, parallel control, ACP methodology, MIMO strict-feedback systems, neural predictor**

I. Introduction

Leader-following consensus control is widely applied in coordinated operations of multi-agent systems during the past years, such as unmanned surface/underwater vehicles [1], [2]. Leader-following consensus control provides a powerful method for achieving a coordinated motion under distributed communication [3]–[5]. The existing leader-following consensus control methods are further separated into consensus tracking control methods [3] and consensus maneuvering control methods [4], [5]. In particular, consensus maneuvering control methods aim to achieve coordination guided by a parameterized virtual leader, including ordinary distributed tracking control laws for controlled objectives and a special update law for the leader's parameter. Consensus maneuvering control is also extended to containment maneuvering guided by multiple parameterized leaders, see [6]–[8].

On one hand, these cooperative maneuvering methods in [4]–[8] mainly focus on how to achieve the desired coordinated motion under the distributed communication among agents. However, the relationship among agents may exist the cooperation and competition in practical applications. Distributed game theory is one of the effective analysis tools for multi-agent systems in competition and cooperation. In the distributed game, each agent can make appropriate decisions through the distributed Nash equilibrium seeking [9]. Some distributed control methods by using the Nash equilibrium seeking have been proposed (see [9], [10]), but distributed consensus maneuvering based on distributed Nash equilibrium seeking is still open. On the other hand, in the existing cooperative maneuvering methods in [4]–[8], neural/fuzzy predictors are designed to recover the uncertain nonlinearities based on the local learning strategy. The local learning-based neural predictor may be hard to utilize existing information of neural networks for other similar tasks and need to be constructed repeatedly [11].

Parallel control is an effective design tool to develop controllers for complex systems [12]. Different from the existing control methods in [4]–[8], [11] related to states passively, the real controllers in parallel control extend to a virtual space and are produced by a parallel system relative to practical models. To some extent, parallel control can realize virtual-

This work was supported in part by the Key R&D Program of Guangdong under Grant 2020B1111010002, in part by the China Post-Doctoral Science Foundation under Grant 2022M722053, and in part by the National Natural Science Foundation of China under Grant U2141234 and 52201369.

*Di Wu and Weidong Zhang are also with School of Information and Communication Engineering, Hainan University, Haikou , China.

reality interaction. In recent years, some parallel control methods have been developed, such as LMI-based parallel control method [12], output-regulation-based parallel control method [13], slide-mode-based parallel control method [14], and backstepping-based parallel control method [15].

A parallel control method is devised by utilizing a distributed Nash equilibrium seeking approach for consensus maneuvering of uncertain multi-agent systems guided by a parameterized virtual leader herein. The model of the agent is in the strict-feedback form and subject to unmatched uncertain nonlinearities. At first, each cost function is constructed via the control objective of consensus maneuvering, the Nash equilibrium is obtained by calculating the partial derivative of the cost function. Then, a neural predictor is developed based on a distributed cooperative learning approach. Next, virtual control laws and a parallel control law are designed by using a dynamic-surface-control (DSC)-based parallel control design, where a linear tracking differentiator (LTD) is constructed to estimate the derivative of virtual control laws. At last, we design the update law by using the distributed consensus maneuvering error feedback. The resulting consensus maneuvering closed-loop is analyzed to be input-to-state stable (ISS). Outputs of agents converge to Nash equilibrium as far as possible.

II. Problem Formulation

The agents can be regarded as the players in the distributed game. The index of these players is $1, ..., M$. For the ith player, consider the dynamics in the following strict-feedback form

$$\begin{cases} \dot{x}_{i,k} = x_{i,k+1} + f_k\left(\bar{x}_{i,k}\right), \quad k = 1, ..., n-1 \\ \dot{x}_{i,n} = u_i + f_n\left(\bar{x}_{i,n}\right) \\ y_i = x_{i,1} \end{cases} \tag{1}$$

where $x_{i,l} \in \mathbb{R}$ with $l = 1, ...n$, $x_{i,n} \in \mathbb{R}$, $u_i \in \mathbb{R}$, $y_i \in \mathbb{R}$ represent the state, the input, and the output, respectively, $\bar{x}_{i,l} = [x_{i,1}, ..., x_{i,l}]^T \in \mathbb{R}^l$ with $l = 1, ...n$, and $f_{i,l}(\bar{x}_{i,l})$ with $l = n$ denotes the uncertain nonlinearities.

In the distributed game, we define $\mathbb{N}$ as the set of players, $\tau_i \in \mathbb{R}^m$ as the action of the ith player, and $J_i(\bar{\tau}) : \mathbb{R}^{Mm} \to \mathcal{R}$ as the cost function of the ith player with $\bar{\tau} = [\tau_1^T, ..., \tau_M^T]^T$. Obviously, $J_i(\bar{\tau})$ is related to the action of the ith player and its neighboring players during distributed gaming. By regulating τ_i, we can minimize $J_i(\bar{\tau})$. We need minimize the cost functions of all players spontaneously as

$$J_i(\tau_i, \tau_{-i}^*) \geq J_i(\tau_i^*, \tau_{-i}^*) \tag{2}$$

where τ_i^* is a Nash equilibrium solution, and $\tau_{-i}^* = [\tau_1^{T*}, ..., \tau_{i-1}^{T*}, \tau_{i+1}^{T*}, ..., \tau_M^{T*}]^T$. When $J_i(\bar{\tau})$ is convex and continuous, there always exists a Nash equilibrium solution. The Nash equilibrium τ_i^* satisfies $\partial J_i(\tau^*)/\partial \tau_i = 0_m$.

A directed graph $\mathcal{G} = (\mathcal{V}, \varepsilon, \Lambda)$ is introduced as the mathematical tool to describe the property of communication topology with a vertex set $\mathcal{V} = (n_1, ..., n_M)$, an edge set $\varepsilon = \{(n_i, n_j) \in \mathcal{V} \times \mathcal{V}\}$, and an adjacency matrix $\Lambda = [a_{i,j}] \in \mathbb{R}^{M \times M}$, where n_i is a node, and $(n_i, n_j) \in \varepsilon$ is a directed information access from agent j to agent i. Note that if $(n_i, n_j) \in \varepsilon$, then $a_{i,j} > 0$, otherwise if $(n_i, n_j) \notin \varepsilon$. In this work, there is no self connection such that $a_{i,i} = 0$.

The Laplacian matrix of the directed graph $\mathcal{G}$ is defined as $L \in \mathbb{R}^{M \times M} := D - \Lambda$, where $D = \text{diag}(d_1, ..., d_M)$ with $d_i = \sum_{j \in \mathcal{N}_i} a_{i,j}$ is a degree matrix and $\mathcal{N}_i = \{j | (n_i, n_j) \in \varepsilon\}$ is a neighbor set of node i. Similar to the definition of the followers' adjacency matrix, the adjacency matrix between players and the virtual leader is defined as $B = \text{diag}(b_1, ..., b_M)$, where $b_i > 0$ means that the node can be access to the leader, otherwise $b_i = 0$.

Assumption 1. There exists a directed spanning tree in the topology.

The control objectives of consensus maneuvering include the following two aspects

- *Geometric objective:* Define $\mathcal{N}_i$ as the set of neighbors. Players are to minimize the consensus maneuvering error with their neighbors and the desired parameterized virtual leader as

$$\min \Sigma_{j \in \mathcal{N}_i} a_{i,j} \frac{1}{2} |y_i - y_j|^2 + b_i \frac{1}{2} |y_i - y_r(\theta)|^2. \tag{3}$$

- *Dynamic objective:* The path variable θ satisfies a speed assignment, which $\dot{\theta}$ nearly converges to a desired velocity signal v_s:

$$\lim_{t \to \infty} |\dot{\theta} - v_s| < \delta \in \mathbb{R}^+. \tag{4}$$

Assumption 2. Assume that $y_r(\theta)$ and $y_r^\theta(\theta)$ are bounded.

III. Consensus Maneuvering Controllers Based on Distributed Nash Equilibrium Seeking

A. Distributed game model for consensus maneuvering of multi-agent systems

For the ith player, we can define the following cost function

$$J_i(\mathbf{y}) = \Sigma_{j \in \mathcal{N}_i} a_{i,j} \frac{1}{2} |y_i - y_j|^2 + b_i \frac{1}{2} |y_i - y_r(\theta)|^2 \tag{5}$$

where $\mathbf{y} = [y_1, ..., y_M]^T$. Due to $J_i(\mathbf{y})$ being continuous and convex, there exists a Nash equilibrium.

Then, we can take the partial derivative of $J_i(\mathbf{y})$ related to y_i, one has

$$\frac{\partial J_i(\mathbf{y})}{\partial y_i} = \Sigma_{j \in \mathcal{N}_i} a_{i,j}(y_i - y_j) + b_i(y_i - y_r(\theta)) \tag{6}$$

where $i \in \mathcal{N}$.

Defining $\mathbf{J}^d(\mathbf{y}) = [(\partial J_1(\mathbf{y})/\partial y_1), ..., (\partial J_M(\mathbf{y})/\partial y_M)]^T$, Eq. (6) can be further transformed into a vector form

$$\mathbf{J}^d(\mathbf{y}) = (\mathcal{L} + B)y - b \otimes y_r(\theta) \tag{7}$$

where $b = [b_1, ..., b_M]^T$.

Letting $\mathbf{J}^d(\mathbf{y}) = \mathbf{0}_M$, we can calculate the unique Nash equilibrium $\mathbf{y}^*$ as

$$\mathbf{y}^* = (\mathcal{L} + B)^{-1} b \otimes y_r(\theta) \tag{8}$$

where $\mathbf{y}^* = [y_1^*, ..., y_M^*]^T$.

B. Neural predictor based on cooperative learning

Step 1. Defining $J_i^d = \partial J_i(\mathbf{y})/\partial y_i$, $\dot{J}_i^d$ along (1) satisfies

$$\begin{aligned}\dot{J}_i^d = (d_i + b_i)\, x_{i,2} - \Sigma_{j\in\mathcal{N}_i} a_{i,j} x_{j,2} \\ - b_i y_r^\theta(\theta)\dot{\theta} + g_{i,1}(x_{i,1}, x_{j,1})\end{aligned} \tag{9}$$

where $d_i = \Sigma_{j\in\mathcal{N}_i} a_{i,j}$, and $g_{i,1}(x_{i,1}, x_{j,1}) = (d_i + b_i) f_{i,1}(x_{i,1}) - b_i f_{j,1}(x_{j,1})$. Then, a neural network is utilized to approximate $g_{i,1}(x_{i,1}, x_{j,1})$:

$$g_{i,1}(x_{i,1}, x_{j,1}) = W_{i,1}^T \varphi_{i,1}(\xi_{i,1}) + \varepsilon_{i,1} \tag{10}$$

where $W_{i,1} \in \mathbb{R}^m$ denotes a weight value matrix of the output layer satisfying $\|W_{i,1}\| \le W_{i,1}^*$ with $W_{i,1}^* \in \mathbb{R}^+$, $\varphi_{i,1}(\cdot) \in \mathbb{R}^m$ denotes the output of the hidden layer consisting of Sigmoid-like activation functions, $\xi_{i,1} = [x_{i,1},\ x_{j,1}]^T$ denotes an input vector of neural network with $j \in \mathcal{N}_i$, and $\varepsilon_{i,1} \in \mathbb{R}$ is an error with $|\varepsilon_{i,1}| \le \varepsilon_{i,1}^*$ and $\varepsilon_{i,1}^* \in \mathbb{R}^+$.

Then, a neural predictor for (9) is designed as:

$$\begin{aligned}\dot{\hat{J}}_i^d = (d_i + b_i) x_{i,2} - \Sigma_{j\in\mathcal{N}_i}^M a_{i,j} x_{j,2} - b_i y_r^\theta(\theta)\dot{\theta} \\ + \hat{W}_{i,1}^T \varphi_{i,1}(\xi_{i,1}) - (\zeta_{i,1} + \rho_{i,1})(\hat{J}_i^d - J_i^d)\end{aligned} \tag{11}$$

where $\zeta_{i,1} \in \mathbb{R}^+$ denotes a tuning parameter, $\rho_{i,1} \in \mathbb{R}^+$ represents a control gain, and $\hat{W}_{i,1} \in \mathbb{R}$ is an estimation of $W_{i,1}$.

We develop the following update law of $\hat{W}_{i,1}$ by using a cooperative learning approach

$$\begin{aligned}\dot{\hat{W}}_{i,1} = & -\Gamma_{i,1}[\varphi_{i,1}(\xi_{i,1})\tilde{J}_i^d + \lambda_{i,1}\hat{W}_{i,1} \\ & + k_{W_{i,1}} \Sigma_{j\in\mathcal{N}_i} a_{i,j}(\hat{W}_{i,1} - \hat{W}_{j,1})]\end{aligned} \tag{12}$$

where $\Gamma_{i,1} \in \mathbb{R}^+$, $\lambda_{i,1} \in \mathbb{R}^+$, $k_{W_{i,1}} \in \mathbb{R}^+$ are tuning parameters, and $\tilde{J}_i^d = \hat{J}_i^d - J_i^d$.

Step k. According to (1), recall the dynamics of $x_{i,k}$ as follows

$$\dot{x}_{i,k} = x_{i,k+1} + f_k(\bar{x}_{i,k}). \tag{13}$$

By using a neural network, $f_k(\bar{x}_{i,k})$ can be recovered as

$$f_k(\bar{x}_{i,k}) = W_{i,k}^T \varphi_{i,k}(\xi_{i,k}) + \varepsilon_{i,k} \tag{14}$$

where $W_{i,k} \in \mathbb{R}^m$ denotes a weight value matrix satisfying $\|W_{i,k}\| \le W_{i,k}^*$ with $W_{i,k}^* \in \mathbb{R}^+$, $\varphi_{i,k}(\cdot) \in \mathbb{R}^m$ denotes the neurons in the hidden layer consisting of Sigmoid-like activation functions, $\xi_{i,k} = \bar{x}_{i,k}$ denotes an input vector of neural network, and $\varepsilon_{i,k} \in \mathbb{R}$ satisfies $|\varepsilon_{i,k}| \le \varepsilon_{i,k}^*$ with $\varepsilon_{i,k}^* \in \mathbb{R}^+$ being the upper bound of $\varepsilon_{i,k}$.

Then, we can design a neural predictor for (13) as:

$$\begin{aligned}\dot{\hat{x}}_{i,k} = x_{i,k+1} + \hat{W}_{i,k}^T \varphi_{i,k}(\xi_{i,k}) \\ - (\zeta_{i,k} + \rho_{i,k})(\hat{x}_{i,k} - x_{i,k})\end{aligned} \tag{15}$$

where $\zeta_{i,k} \in \mathbb{R}^+$ denotes a tuning parameter, $\rho_{i,k} \in \mathbb{R}^+$ is a control gain, and $\hat{W}_{i,k} \in \mathbb{R}^m$ is an estimation of $W_{i,1}$.

Based on a cooperative learning approach, we can develop the following update law for $\hat{W}_{i,k}$ as

$$\begin{aligned}\dot{\hat{W}}_{i,k} = & -\Gamma_{i,k}[\varphi_{i,k}(\xi_{i,k})\tilde{x}_{i,k} + \lambda_{i,k}\hat{W}_{i,k} \\ & + k_{W_{i,k}} \Sigma_{j\in\mathcal{N}_i} a_{i,j}(\hat{W}_{i,k} - \hat{W}_{j,k})]\end{aligned} \tag{16}$$

where $\Gamma_{i,k} \in \mathbb{R}^+$, $\lambda_{i,k} \in \mathbb{R}^+$, and $k_{W_{i,k}} \in \mathbb{R}^+$, and $\tilde{x}_{i,k} = \hat{x}_{i,k} - x_{i,k}$.

Step n. Recall the dynamics of $x_{i,n}$ as follows

$$\dot{x}_{i,n} = u_i + f_{i,n}(\bar{x}_{i,n}). \tag{17}$$

Letting $x_{i,n+1} = u_i + f_{i,n}(\bar{x}_{i,n})$, the dynamics of $x_{i,n+1}$ is yielded by

$$\dot{x}_{i,n+1} = \dot{u}_i + F_i(\bar{x}_{i,n}) \tag{18}$$

where $F_i(\bar{x}_{i,n}) = \Sigma_{l=1}^n \frac{\partial f_{i,n}(\bar{x}_{i,n})}{\partial x_{i,l}} \dot{x}_{i,l}$.

Similarly, we can use the following neural network to recover $F_i(\bar{x}_{i,n})$ as

$$F_i(\bar{x}_{i,n}) = W_{i,n}^T \varphi_{i,n}(\xi_{i,n}) + \varepsilon_{i,n} \tag{19}$$

where $W_{i,n} \in \mathbb{R}^m$ denotes a weight value matrix of the output layer satisfying $\|W_{i,n}\| \le W_{i,n}^*$ with $W_{i,n}^* \in \mathbb{R}^+$, $\varphi_{i,n}(\cdot) \in \mathbb{R}^m$ denotes the neurons in the hidden layer consisting of Sigmoid-like activation functions, $\xi_{i,n} = \bar{x}_{i,n}$ denotes an input vector of neural network, and $\varepsilon_{i,n} \in \mathbb{R}$ satisfies $|\varepsilon_{i,n}| \le \varepsilon_{i,n}^*$ with $\varepsilon_{i,n}^* \in \mathbb{R}^+$ being the upper bound of $\varepsilon_{i,n}$.

A neural predictor for (17) is devised as:

$$\begin{aligned}\dot{\hat{x}}_{i,n+1} = \dot{u}_i + \hat{W}_{i,n}^T \varphi_{i,n}(\xi_{i,n}) \\ - (\rho_{i,n+1} + \zeta_{i,n+1})(\hat{x}_{i,n+1} - x_{i,n+1})\end{aligned} \tag{20}$$

where $\zeta_{i,n+1} \in \mathbb{R}$ denotes a positive tuning parameters, $\rho_{i,n+1} \in \mathbb{R}$ denotes a positive control gain, and $\hat{W}_{i,n} \in \mathbb{R}^m$ is an estimation of $W_{i,n}$.

By using the cooperative learning approach, we can develop the following update law for $\hat{W}_{i,n}$ is updated as

$$\begin{aligned}\dot{\hat{W}}_{i,n} = & -\Gamma_{i,n}[\varphi_{i,n}(\xi_{i,n})\tilde{x}_{i,n+1} + \lambda_{i,n}\hat{W}_{i,n} \\ & + k_{W_{i,n}} \Sigma_{j\in\mathcal{N}_i} a_{i,j}(\hat{W}_{i,n} - \hat{W}_{j,n})]\end{aligned} \tag{21}$$

where $\Gamma_{i,n} \in \mathbb{R}^+$, $\lambda_{i,n} \in \mathbb{R}^+$, and $k_{W_{i,n}} \in \mathbb{R}^+$ are tuning parameters, and $\tilde{x}_{i,n} = \hat{x}_{i,n} - x_{i,n}$.

Defining $\tilde{W}_{i,1} = \hat{W}_{i,1} - W_{i,1}$, $\tilde{W}_{i,k} = \hat{W}_{i,k} - W_{i,k}$, and $\tilde{W}_{i,n} = \hat{W}_{i,n} - W_{i,n}$, and the subsystem consisting of $\tilde{J}_i^d$, $\tilde{x}_{i,l}$, $\tilde{x}_{i,n+1}$, $\tilde{W}_{i,1}$, $\tilde{W}_{i,l}$, and $\tilde{W}_{i,n}$ is given by

$$\left\{\begin{aligned}
\dot{\tilde{J}}_i^d &= -(\rho_{i,1} + \zeta_{i,1})\tilde{J}_i^d + \hat{W}_{i,1}^T \varphi(\xi_{i,1}) - \varepsilon_{i,1} \\
\dot{\tilde{x}}_{i,l} &= -(\rho_{i,l} + \zeta_{i,l})\tilde{x}_{i,l} + \hat{W}_{i,l}^T \varphi(\xi_{i,l}) - \varepsilon_{i,l} \\
\dot{\tilde{x}}_{i,n+1} &= -(\rho_{i,n+1} + \zeta_{i,n+1})\tilde{x}_{i,n+1} \\
&\quad + \hat{W}_{i,n}^T \varphi(\xi_{i,n}) - \varepsilon_{i,n} \\
\dot{\tilde{W}}_{i,1} &= -\Gamma_{i,1}[\varphi_{i,1}(\xi_{i,1})\tilde{J}_i^d + \lambda_{i,1}\hat{W}_{i,1} \\
&\quad + k_{W_{i,1}} \Sigma_{j\in\mathcal{N}_i} a_{i,j}(\hat{W}_{i,1} - \hat{W}_{j,1})] \\
\dot{\tilde{W}}_{i,l} &= -\Gamma_{i,l}[\varphi_{i,l}(\xi_{i,l})\tilde{x}_{i,l} + \lambda_{i,l}\hat{W}_{i,l} \\
&\quad + k_{W_{i,l}} \Sigma_{j\in\mathcal{N}_i} a_{i,j}(\hat{W}_{i,l} - \hat{W}_{j,l})] \\
\dot{\tilde{W}}_{i,n} &= -\Gamma_{i,n}[\varphi_{i,n}(\xi_{i,n})\tilde{x}_{i,n+1} + \lambda_{i,n}\hat{W}_{i,n} \\
&\quad + k_{W_{i,n}} \Sigma_{j\in\mathcal{N}_i} a_{i,j}(\hat{W}_{i,n} - \hat{W}_{j,n})]
\end{aligned}\right. \tag{22}$$

where $l = 2, ..., n-1$. Consider the following lemma:

Lemma 1. The subsystem (22) is ISS.

Proof. Consider the following Lyapunov function candidate for the subsystem (22) as $V_p = \frac{1}{2}\Sigma_{i=1}^M \Sigma_{l=1}^{n+1} \tilde{x}_{i,l}^2 +$

$\frac{1}{2}\Sigma_{i=1}^{M}\Sigma_{l=1}^{n}\tilde{W}_{i,l}^{T}\Gamma_{i,l}^{-1}\tilde{W}_{i,l}$, where $\tilde{x}_{i,n} = 0$. The time derivative of V_p satisfies $\dot{V}_p \leq -\lambda_{\min}(\rho+\zeta)\|\tilde{X}\|^2 - (\lambda_{\min}(\lambda)+\lambda_{\min}(k_W)\lambda_{\min}(\mathcal{L}))\|\tilde{W}\|^2 + \|\tilde{X}\|\|\varepsilon\| + (\lambda_{\max}(\lambda)+\lambda_{\max}(k_W)\lambda_{\max}(\mathcal{L}))\|\tilde{W}\|\|W\| \leq -c_1\|E\|^2 + \|E\|\|h\|$, where $\tilde{X} = [\tilde{J}_1^d, \tilde{x}_{1,2}, ..., \tilde{x}_{1,n+1}, ..., \tilde{J}_M^d, \tilde{x}_{M,2}, ..., \tilde{x}_{M,n+1}]^T$, $\tilde{W} = [\tilde{W}_{1,1}^T, ..., \tilde{W}_{1,n}^T, ..., \tilde{W}_{M,1}^T, ..., \tilde{W}_{M,n}^T]^T$, $\varepsilon = [\varepsilon_{1,1}, ..., \varepsilon_{1,n}, ..., \varepsilon_{M,1}, ..., \varepsilon_{M,n}]^T$, $W_i = [W_{1,1}^T, ..., W_{1,n}^T, ..., W_{M,1}^T, ..., W_{M,n}^T]^T$, $\rho = \text{diag}\{\rho_{1,1}, ..., \rho_{1,n}, ..., \rho_{M,1}, ..., \rho_{M,n}\}$, $\zeta = \text{diag}\{\zeta_{1,1}, ..., \zeta_{1,n}, ..., \zeta_{M,1}, ..., \zeta_{M,n}\}$, $\lambda = \text{diag}\{\lambda_{1,1}, ..., \lambda_{1,n}, ..., \lambda_{M,1}, ..., \lambda_{M,n}\}$, and $k_W = \text{diag}\{k_{W_{1,1}}, ..., k_{W_{1,n}}, ..., k_{W_{M,1}}, ..., k_{W_{M,n}}\}$. $E = [\|\tilde{X}\|, \|\tilde{W}\|]^T$, $h = [\|\varepsilon\|, (\lambda_{\max}(\lambda)+\lambda_{\max}(\mathcal{L}))\|W\|]^T$, and $c_1 = \min\{\lambda_{\min}(\rho_i+\zeta_i), (\lambda_{\min}(\lambda_i)+\lambda_{\min}(\mathcal{L}))\}$. Since $\|E\| \geq (\|\varepsilon\| + (\lambda_{\max}(\lambda_i)+\lambda_{\max}(\mathcal{L}))\|W\|)/\eta_1 c_1 \geq \|h\|/\eta_1 c_1$ makes $\dot{V}_p \leq -(1-\eta_1)c_1\|E\|^2$, where $0<\eta_1<1$, we can prove the system (22) to be ISS. Consider $\kappa_{p1}(s) = \lambda_{\min}(P)s^2/2$ and $\kappa_{p2}(s) = \lambda_{\max}(P)s^2/2$ with $P = \text{diag}(1, \Gamma_{1,1}^{-1}, ...\Gamma_{1,n}^{-1}, ..., \Gamma_{M,1}^{-1}, ...\Gamma_{M,n}^{-1})$, and there exists a $\mathcal{KL}$ function $\alpha_1(\cdot)$ and $\mathcal{K}_\infty$ functions $\kappa^{\varepsilon}(\cdot)$ and $\kappa^{W}(\cdot)$ satisfying $\|E(t)\| \leq \alpha_1(\|E(t_0)\|, t-t_0) + \kappa^{\varepsilon}(\|\varepsilon\|) + \kappa^{W}(\|W\|)$, where $\kappa^{\varepsilon}(s) = (s\sqrt{\lambda_{\max}(P)})/(\eta_1 c_1\sqrt{\lambda_{\min}(P)})$ and $\kappa^{W}(s) = (s(\lambda_{\max}(\lambda)+\lambda_{\max}(\mathcal{L})\sqrt{\lambda_{\max}(P)})/(\eta_1 c_1\sqrt{\lambda_{\min}(P)})$.

Though we prove the the proposed cooperative learning-based neural predictor is ISS. But it should be that the proposed predictor is only suitable for consensus maneuvering. Because containment maneuvering has multiple parameterized virtual leaders, the how to design a similar cooperative learning-based neural predictor for containment maneuvering is still open.

C. DSC-based parallel control law

Step 1. We consider θ satisfying $\dot{\theta}=v_s-\omega$. The dynamics of $\hat{J}_i^d$ is rewritten as:

$$\begin{aligned}\dot{\hat{J}}_i^d =& (d_i+b_i)x_{i,2} - \Sigma_{j\in\mathcal{N}_i}a_{i,j}x_{j,2} - b_i y_r^{\theta}(\theta)(v_s-\omega) \\ &+ \hat{W}_{i,1}^T\varphi_{i,1}(\xi_{i,1}) - (\zeta_{i,1}+\rho_{i,1})\tilde{J}_i^d\end{aligned} \tag{23}$$

where the definitions of v_s and ω are same with *Section II.*

From (23), we develop the first virtual control law $\alpha_{i,1}$ as

$$\begin{aligned}\alpha_{i,1} = \frac{1}{d_i+b_i}\{&-\rho_{i,1}J_i^d + \Sigma_{j\in\mathcal{N}_i}a_{i,j}x_{j,2} \\ &+ b_i y_r^{\theta}(\theta)v_s - \hat{W}_{i,1}^T\varphi_{i,1}(\xi_{i,1})\}.\end{aligned} \tag{24}$$

Without utilizing the first-order low-pass filter in the traditional DSC method, a second-order LTD is employed in DSC method as follows:

$$\begin{cases}\dot{v}_{i,1} = v_{i,1}^d, \\ \dot{v}_{i,1}^d = -\gamma_{i,1}^2[(v_{i,1}-\alpha_{i,1}) + 2(v_{i,1}^d/\gamma_{i,1})]\end{cases} \tag{25}$$

where $\gamma_{i,1} \in \mathbb{R}^+$ denotes the time constant of TD.

We substitute (24) into (23), and it follows that:

$$\begin{aligned}\dot{\hat{J}}_i^d =& -\rho_{i,1}\hat{J}_i^d - \zeta_{i,1}\tilde{J}_i^d + b_i y_r^{\theta}(\theta)\omega \\ &- (d_i+b_i)\tilde{x}_{i,2} + (d_i+b_i)\hat{z}_{i,2} + \iota_{i,1}\end{aligned} \tag{26}$$

where $\hat{z}_{i,2} = \hat{x}_{i,2} - \iota_{i,2}$, $\iota_{i,1} = v_{i,1} - \alpha_{i,1}$, and $|v_{i,1}-\alpha_{i,1}| \leq \iota_{i,1}^*$ with $\iota_{i,1}^* \in \mathbb{R}^+$.

Step k. Construct a new error surface $\hat{z}_{i,k} = \hat{x}_{i,k} - v_{i,k-1}$, and take its dynamics as follows

$$\begin{aligned}\dot{\hat{z}}_{i,k} =& x_{i,k+1} + \hat{W}_{i,k}^T\varphi_{i,k}(\xi_{i,k}) - v_{i,k-1}^d \\ &- (\zeta_{i,k}+\rho_{i,k})(\hat{x}_{i,k} - x_{i,k})\end{aligned} \tag{27}$$

Then, we construct the kth virtual control law $\alpha_{i,k}$ to stabilize (27)

$$\alpha_{i,k} = -\rho_{i,k}z_{i,k} + v_{i,k-1}^d - \hat{W}_{i,k}^T\varphi_{i,k}(\xi_{i,k}) - e_{i,k-1}. \tag{28}$$

where $e_{i,1} = (d_i+b_i)\hat{J}_i^d$ and $e_{i,k-1} = \hat{z}_{i,k-1}$.

Similarly, an estimated derivative $v_{i,k}^d$ related to $\alpha_{i,k}$ can be obtained by the following second-order linear TD as

$$\begin{cases}\dot{v}_{i,k} = v_{i,k}^d, \\ \dot{v}_{i,k}^d = -\gamma_{i,k}^2[(v_{i,k}-a_{i,k}) + 2(v_{i,k}^d/\gamma_{i,k})]\end{cases} \tag{29}$$

where $\gamma_{i,k} \in \mathbb{R}^+$ denotes the time constant.

Then, it follows that by taking (28) into (27)

$$\begin{aligned}\dot{\hat{z}}_{i,k} =& -\rho_{i,k}\hat{z}_{i,k} - \zeta_{i,k}\tilde{x}_{i,k} - \tilde{x}_{i,k+1} \\ &+ \hat{z}_{i,k+1} - e_{i,k-1} + \iota_{i,k},\end{aligned} \tag{30}$$

where $\hat{z}_{i,k+1} = \hat{x}_{k+1} - v_{i,k}$, $\iota_{i,k} = v_{i,k} - \alpha_{i,k}$, and and $|\iota_{i,k}| \leq \iota_{i,k}^*$ with $\iota_{i,k}^* \in \mathbb{R}^+$.

Step n. Letting $\hat{z}_{i,n} = \hat{x}_{i,n} - v_{i,n-1}$, consider the model of $\hat{z}_{i,n}$ as follows

$$\dot{z}_{i,n} = x_{i,n+1} - v_{i,n-1}^d. \tag{31}$$

Then, we can obtain the following virtual control law

$$\alpha_{i,n} = -\rho_{i,n}z_{i,n} - \hat{z}_{i,n-1} + v_{i,n-1}^d \tag{32}$$

where $\rho_{i,n} \in \mathbb{R}^+$.

Similarly, an estimated derivative $v_{i,n}^d$ related to $\alpha_{i,n}$ can be obtained by the following TD as

$$\begin{cases}\dot{v}_{i,n} = v_{i,n}^d, \\ \dot{v}_{i,n}^d = -\gamma_{i,n}^2[(v_{i,n}-a_{i,n}) + 2(v_{i,n}^d/\gamma_{i,n})],\end{cases} \tag{33}$$

where $\gamma_{i,n} \in \mathbb{R}^+$.

The dynamics of $\hat{z}_{i,n}$ is further transformed via substituting (28) into (31) into

$$\dot{\hat{z}}_{i,n} = -\rho_{i,n}z_{i,n} + z_{i,n+1} - \hat{z}_{i,n-1} + \iota_{i,n} \tag{34}$$

where $\iota_{i,n} = v_{i,n} - \alpha_{i,n}$, and $|\iota_{i,n}| \leq \iota_{i,n}^*$ with $\iota_{i,n}^* \in \mathbb{R}^+$.

Step n+1. Define the following error surfaces $z_{i,n+1}$ and $\hat{z}_{i,n+1}$

$$\begin{cases}z_{i,n+1} = x_{i,n+1} - v_{i,n} \\ \hat{z}_{i,n+1} = \hat{z}_{i,n+1} - v_{i,n}.\end{cases} \tag{35}$$

$\dot{\hat{z}}_{i,n+1}$ is given by

$$\begin{aligned}\dot{\hat{z}}_{i,n+1} =& \dot{u}_i + \hat{W}_{i,n}^T\varphi_{i,n}(\xi_{i,n}) - (\rho_{i,n+1}+\zeta_{i,n+1}) \\ &\times \tilde{x}_{i,n+1} - v_{i,n}^d.\end{aligned} \tag{36}$$

A parallel control law u_i is designed for consensus maneuvering as follows:

$$\begin{aligned}\dot{u}_i =& -\rho_{i,n+1}z_{i,n+1} + v_{i,n-1}^d - \hat{W}_{i,n}^T\varphi_{i,n}(\xi_{i,n}) \\ &+ \zeta_{i,n+1}\tilde{x}_{i,n} - \hat{z}_{i,n}.\end{aligned} \tag{37}$$

Substituting (37) into (36), one has:

$$\dot{\hat{z}}_{i,n+1} = -\rho_{i,n+1}\hat{z}_{i,n+1} - \rho_{i,n+1}\tilde{x}_{i,n+1} - \hat{z}_{i,n}. \tag{38}$$

D. Update law for the virtual leader

The update of virtual leader is determined by its neighboring followers. Therefore, the following update law is developed with the information of its neighboring followers

$$\begin{cases} \dot{\theta} = v_s - \omega, \\ \omega = -\mu \sum_{i=1}^{M} b_i y_r^{\theta}(\theta)\hat{J}_i^d \end{cases} \tag{39}$$

with $\mu \in \mathbb{R}$ being a positive tuning parameter.

The error subsystem consisting of $\hat{J}_i^d$, $\hat{z}_{i,k}$, $z_{i,n}$, and $\hat{z}_{i,n+1}$ is summarized as:

$$\begin{cases} \dot{\hat{J}}_i^d = -\rho_{i,1}\hat{J}_i^d - \zeta_{i,1}\tilde{J}_i^d + b_i y_r^{\theta}(\theta)\omega \\ \qquad - (d_i+b_i)\tilde{x}_{i,2} + (d_i+b_i)\hat{z}_{i,2} \\ \dot{\hat{z}}_{i,k} = -\rho_{i,k}\hat{z}_{i,k} - \zeta_{i,k}\tilde{x}_{i,k} - \tilde{x}_{i,k+1} \\ \qquad + \hat{z}_{i,k+1} - e_{i,k-1} + \iota_{i,k} \\ \dot{z}_{i,n} = -\rho_{i,n}z_{i,n} + z_{i,n+1} - \hat{z}_{i,n-1} + \iota_{i,n}, \\ \dot{\hat{z}}_{i,n+1} = -\rho_{i,n+1}\hat{z}_{i,n+1} - \rho_{i,n+1}\tilde{x}_{i,n+1} - \hat{z}_{i,n} \\ \omega = -\mu \sum_{i=1}^{M} b_i y_r^{\theta}(\theta)\hat{J}_i^d \end{cases} \tag{40}$$

where $k = 2, ..., n-1$.

Lemma 2. The subsystem (40) is ISS.

Proof. Define $\hat{Z}_i = [\hat{J}_i^d, \hat{z}_{i,2}, ..., \hat{z}_{i,n-1}, z_{i,n}, \hat{z}_{i,n+1}]^T$. Consider a Lyapunov function $V_d = \frac{1}{2}\Sigma_{i=1}^{M}\hat{Z}_i^T\hat{Z}_i$. Its time derivative satisfies $\dot{V}_d \le -\lambda_{\min}(\rho)\|\hat{Z}\|^2 + c_3\|\hat{Z}\|\|\tilde{X}\| + \|\hat{Z}\|\|\iota\| \le -\lambda_{\min}(\rho)\|\hat{Z}\|^2 + \|\hat{Z}\|\|U_2\|$, where $\hat{Z} = [\hat{Z}_1^T, ..., \hat{Z}_M^T]^T$, $\iota = [\iota_{1,1}, .., \iota_{M,1}, .., \iota_{1,n-1}, .., \iota_{M,n-1}]^T$ $c_3 = \lambda_{\max}(\zeta) + d_i + b_i$, and $U_2 = [c_3\|\tilde{X}\|, \|\iota\|]^T$. Note that $\|\hat{Z}\| \ge (c_3\|\tilde{X} + \|\iota\|)\|/\eta_2\lambda_{\min}(\rho) \ge (\|U_2\|\|/\eta_2\lambda_{\min}(\rho))$ makes $\dot{V}_d \le -\lambda_{\min}(\rho)(1-\eta_2)\|\hat{Z}\|^2$, where $0 < \eta_2 < 1$. we can prove the system (40) to be ISS with a $\mathcal{KL}$ class function $\beta_2(\cdot)$ and two $\mathcal{K}_\infty$ class functions $\kappa^{\tilde{X}}(\cdot)$ and $\kappa^{\iota}(\cdot)$ as $\|\hat{Z}(t)\| \le \beta_2(\|\hat{Z}(t_0)\|, t-t_0) + (\kappa^{\tilde{X}}(\|\tilde{X}\|) + \kappa^{\iota}(\|\iota\|))$, where $\kappa^{\tilde{Z}}(s) = (c_3 s)/(\eta_2\lambda_{\min}(\rho))$ and $\kappa^{\iota}(s) = s/(\eta_2\lambda_{\min}(\rho))$.

Theorem 1. For the multi-agent system (1), the resulting closed-loop system is ISS based on the cooperative learning-based neural predictors (11), (12), (15), (16), (20) and (21), DSC-based parallel control laws (24), (28), (32), and (37), and update law for the virtual leader (39) under *Assumptions 1-2*. Moreover, the outputs of all agents converge to the Nash equilibrium as far as possible.

Proof. The detailed proof process can be refered to [4], and a biref analysis is given herein for saving space. The resulting consensus maneuvering closed-loop is cascaded by subsystems (22) and (40). These subsystems (22) and (40) have been proved to be ISS, separately. The state of the subsystem (22) $\tilde{X}$ is an input of the subsystem (40). According to *Lemma 4.7* in [16], we can prove the closed-loop to be ISS. Since $\tilde{J}_d^i$ and $\hat{J}_d^i$ have been proved to be bounded, J_d^i can be proved to be bounded, too. Letting $y = [y_1, \ ..., \ y_M]^T$, we obtain $\mathbf{J}^d(\mathbf{y}) = (\mathcal{L}+B)y - b\otimes y_r$. One has $\|y - (\mathcal{L}+B)^{-1}b\otimes y_r(\theta)\| \le \|\mathbf{J}^d(\mathbf{y})\|/\lambda_{\min}(\mathcal{L}+B)$. Therefore, the outputs of all agents can be proved to converge to the Nash equilibrium as far as possible.

According to properties of input-to-state stable systems, all error signals in the resulting closed-loop is ultimate bounded. Therefore, two objectives of distributed consensus maneuvering is also satisfied.

IV. A Numerical Simulation Example

Consider an uncertain multi-agent systems shown in Fig. 1. Two followers labeled as $1, 2$, and a virtual leaders labeled as 0. The virtual leader moves along $y_r(\theta) = \sin(\theta)$ with $v_s = 1$. The dynamics of the follower is given by

$$\begin{cases} \dot{x}_{i,1} = x_{i,2} + 0.15x_{i,1}^2, \\ \dot{x}_{i,2} = u_i + 0.15x_{i,1}^2 + \cos(x_{i,2}), \\ y_i = x_{i,1}. \end{cases} \tag{41}$$

The parameters used in the simulation are $\rho_{i,1} = 10$, $\rho_{i,2} = \rho_{i,3} = 1$, $\gamma_{i,1} = \gamma_{i,2} = 200$, $\Gamma_{i,1} = 10000$, $\Gamma_{i,2} = 100$, $\lambda_{i,1} = \lambda_{i,2} = 5\times10^{-4}$, $k_{W_{i,1}} = 1\times10^{-5}$, $k_{W_{i,2}} = 1\times10^{-3}$, $\zeta_{i,1} = 90$, $\zeta_{i,2} = 19$, $\mu = 0.5$. Fig. 2 shows that the output trajectories of multi-agent systems can track the virtual leader based on the proposed method. Fig. 3 demonstrates that the unmatched uncertain nonlinearities can be estimated by the designed NN. Fig. 4 plots the control input u_1.

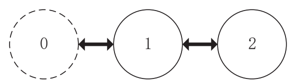

Fig. 1. The communication topology.

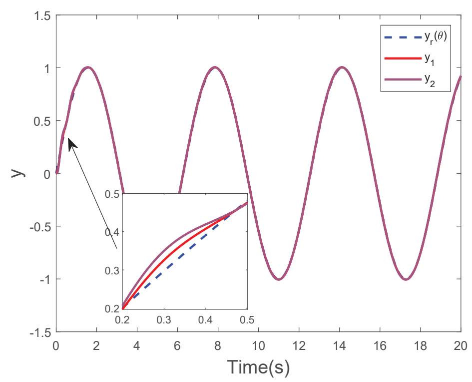

Fig. 2. Output trajectories.

V. Conclusion

In this paper, parameterized virtual leader-guided consensus maneuvering was investigated based on the distributed game for strict-feedback multi-agent systems in the presence of unmatched uncertainties. A distributed consensus maneuvering method was proposed by using the distributed Nash equilibrium seeking approach and the DSC-based parallel control method. The proposed distributed consensus maneuvering method could minimize the cost function of consensus maneuvering in the distributed game, and enabled the stability of the resulting closed-loop system. At last, a simulation example

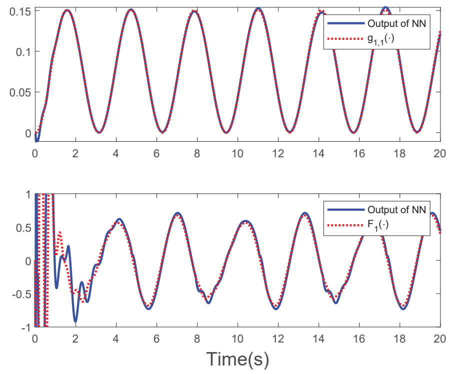

Fig. 3. Learning profiles of NN based on the proposed method.

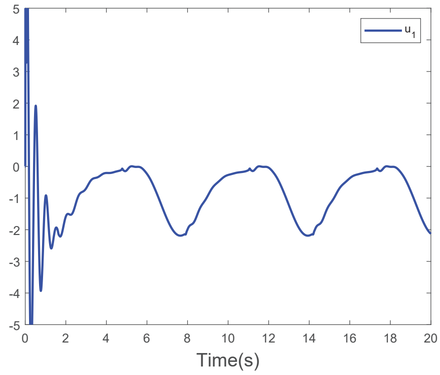

Fig. 4. Evolution of control input u_1.

was carried out to verify the efficacy of the proposed consensus maneuvering controllers based on distributed game and DSC-based parallel control.

In future works, we consider extend this work to the following aspects. Firstly, we will further consider the agent subject to faults and DoS attacks. Secondly, we will consider data corruption during the cooperative learning process. Thirdly, safety of the parallel system is also our one of research interests.

REFERENCES

[1] L. Qiao and W. Zhang, "Double-loop integral terminal sliding mode tracking control for uuvs with adaptive dynamic compensation of uncertainties and disturbances," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 1, pp. 29–53, 2019.

[2] Y. Zhang, D. Wang, Y. Yin, and Z. Peng, "Event-triggered distributed coordinated control of networked autonomous surface vehicles subject to fully unknown kinetics via concurrent-learning-based neural predictor," *Ocean Engineering*, vol. 234, p. 108966, 2021.

[3] W. Wang, Y. Li, and S. Tong, "Neural-network-based adaptive event-triggered consensus control of nonstrict-feedback nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, 2020, in press.

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

[5] Z. Peng, J. Wang, and D. Wang, "Distributed maneuvering of autonomous surface vehicles based on neurodynamic optimization and fuzzy approximation," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 1083–1090, May 2018.

[6] Y. Zhang, D. Wang, Z. Peng, and T. Li, "Distributed containment maneuvering of uncertain multiagent systems in mimo strict-feedback form," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 1354–1364, 2021.

[7] Y. Zhang, D. Wang, Z. Peng, T. Li, and L. Liu, "Event-triggered ISS-modular neural network control for containment maneuvering of nonlinear strict-feedback multi-agent systems," *Neurocomputing*, vol. 377, pp. 314–324, 2020.

[8] Z. Peng, D. Wang, T. Li, and M. Han, "Output-feedback cooperative formation maneuvering of autonomous surface vehicles with connectivity preservation and collision avoidance," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2527–2535, Jun. 2020.

[9] M. Ye and G. Hu, "Distributed nash equilibrium seeking by a consensus based approach," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4811–4818, 2017.

[10] Y. Zhang, W. Wu, and W. Zhang, "Noncooperative game-based cooperative maneuvering of intelligent surface vehicles via accelerated learning-based neural predictors," *IEEE Transactions on Intelligent Vehicles*, pp. 1–10, 2022.

[11] S.-L. Dai, S. He, Y. Ma, and C. Yuan, "Distributed cooperative learning control of uncertain multiagent systems with prescribed performance and preserved connectivity," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 7, pp. 3217–3229, 2021.

[12] Q. Wei, H. Li, and F.-Y. Wang, "Parallel control for continuous-time linear systems: A case study," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 919–928, 2020.

[13] ——, "A novel parallel control method for continuous-time linear output regulation with disturbances," *IEEE Transactions on Cybernetics*, pp. 1–11, 2021.

[14] Z. Chunyang, G. Qing, L. Jinhu, D. Yue, and X. Zhiyu, "An integral sliding-mode parallel control approach for general nonlinear systems via piecewise affine linear models," *arXiv preprint arXiv:2101.05039*, 2021.

[15] J. Lu, Q. Wei, T. Zhou, L. Han, and F.-Y. Wang, "Backstepping-based parallel control for cascaded nonlinear systems," in *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, 2021, pp. 70–73.

[16] H. K. Khalil, "Nonlinear systems," *Prentice Hall*, 2002.