A Two-Level Neural-RL-based Approach for Hierarchical Multiplayer Systems under Mismatched Uncertainties

Xiangnan Zhong, *Member, IEEE,* and Zhen Ni, *Senior Member, IEEE*

***Abstract*—AI and reinforcement learning (RL) have attracted great attention in the study of multiplayer systems over the past decade. Despite the advances, most of the studies are focused on synchronized decision-making to attain Nash equilibrium, where all the players take actions simultaneously. On the other hand, however, in complex applications, certain players may have an advantage in making sequential decisions and this situation introduces a hierarchical structure and influences how other players respond. The control design for such system is challenging since it relies on solving the coupled Hamilton-Jacobi equation. The situation becomes more difficult when the learning process is exposed to complex uncertainties with unreliable data being exchanged. Therefore, in this paper, we develop a new learning-based control approach for a class of nonlinear hierarchical multiplayer systems subject to mismatched uncertainties. Specifically, we first formulate this new problem as a multiplayer Stackelberg-Nash game in conjunction with the hierarchical robust-optimal transformation. Theoretical analysis confirms the equivalence of this transformation and ensures that the designed control policies can achieve stable equilibrium. Then, a two-level neural-RL-based approach is developed to automatically and adaptively learn the solutions. The stability of this online learning process is also provided. Finally, two numerical examples are presented to demonstrate the effectiveness of the developed learning-based robust control design.**

***Impact Statement*—The integration of AI into game theory has revolutionized the analysis and resolution of interactions among players. Particularly, the adoption of reinforcement learning (RL), a powerful AI learning paradigm, has attracted increasing attention in recent years. While RL has achieved many success in multiplayer games, existing studies primarily focus on synchronized decision-making to achieve Nash equilibrium, overlooking the existing of hierarchical optimization and asymmetric decision-making in some practical scenarios. The challenges of control design in such systems are illuminated in this research, characterized by coupled relationship among players and nonlinear system evolution. The complexity is further exacerbated by uncertain situations, which introduce additional hurdles to the learning process. To bridge this gap, this paper develops a two-level neural-RL-based approach for hierarchical multiplayer systems under mismatched uncertainties. This work facilitates the development of more sophisticated and adaptive control approaches and enables AI players to efficiently navigate hierarchical multiplayer environments.**

***Index Terms*—Reinforcement learning, artificial intelligence, neural networks, hierarchical multiplayer systems, mismatched uncertainties, Stackelberg-Nash game, and theoretical analysis.**

I. Introduction

Artificial intelligence (AI) plays a crucial role in multiplayer games, which revolutionize the way strategic interactions are analyzed, understood, and even played out in practice. The applications of multiplayer AI systems are in diverse fields, including autonomous vehicles [1], [2], robotics [3], [4], economics [5], computer gaming [6], [7], and more. Among the current successful AI-based stories, many of them are centered by reinforcement learning (RL) [8]–[15], a learning mechanism which mimics human learning through exploration and interaction with the environment.

In recent years, many efforts have been dedicated in the development of RL-based control and decision-making for multiplayer systems [16]–[23]. Despite the success, most of the studies were focused on synchronized decision-making to achieve Nash equilibrium, where all the players take actions simultaneously. However, this design limits the strategic depth since there is little room for hierarchical or sequential decision-making strategies. It is important to note that hierarchical optimization and asymmetric decision-making widely exist, and are usually the essential concepts in reflecting the complex interactions among players in real-world scenarios [24]–[26]. For example, in energy management and demand reduction applications, power utility companies typically issue a load shedding command during peak periods to avoid the excessive costs of purchasing electricity at high prices. Local manufacturers and residential communities then respond by adjusting their energy consumption to fulfill individual requirements while complying with the utility's directives [27]. Such hierarchical optimization is also commonly seen in smart grid [28] and transportation [29], where the low-level optimization is based on a consequence of high-level decision. A valuable mechanism to model such process is the Stackelberg game [30]–[32], which is a strategic interaction between the dominant player (leader) and the subordinates (followers). Specifically, the leader aims to optimize its control performance first, taking into account the anticipated responses of the followers, while the followers aim to react optimally later given the leader's decisions. Particularly, for the scenario with one leader hierarchy and multiple followers making decisions simultaneously, it can be formulated as a Stackelberg-Nash game [33].

It is noteworthy that the learning-based control design for such hierarchical multiplayer systems is more challenging than the conventional multiplayer systems with simultaneous decision-making process. This is because the equilibrium in

This work was supported in part by the National Science Foundation under Grant 2047010, 2047064, 1947419, and 2205205. Ni was also partially supported by the Department of Transportation under Grant 69A3552348304. (*Corresponding author: Xiangnan Zhong.*)

X. Zhong and Z. Ni are with the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA. (email: xzhong@fau.edu, zhenni@fau.edu)

the corresponding formulated multiplayer Stackelberg-Nash game relies on solving a coupled Hamilton-Jacobi (HJ) equation, which captures the interdependence among players' strategies and the system's dynamics. Obtaining the solution of such equation is difficult due to the high nonlinearity and couplings. In [34], RL algorithms were developed to address the Stackelberg game for a two-player linear-quadratic continuous-time system. The two-player nonlinear hierarchical control problem was investigated in [35] and a learning-based algorithm was developed with two parametric HJ equations and a costate equation to obtain the Stackelberg equilibrium. The authors in [36] considered the multiplayer Stackelberg-Nash game for a nonlinear hierarchical system. The value-iteration-based integral RL algorithm was developed to asymptotically converge the hierarchical system to the equilibrium strategies under the weak coupling conditions.

However, the above research results are built on the secure interaction experience. If the communication network is vulnerable to noise and/or malicious attacks, the learning systems will suffer from uncertainties or perturbations, which makes the received data unreliable [37]. Therefore, the RL methods can not be applied directly. In [38], the disturbance in the hierarchical system was considered as an additional player, and the relationship between the disturbance and other players in the game was formulated as the zero-sum game. RL method was designed for the game to achieve Stackelberg-Nash-Saddle equilibrium. The authors in [39] developed the robust approach through the corresponding optimal control for Stackelberg-Nash game with matched uncertainties in the event-triggered mechanism. The sliding mode control technique was integrated in the learning-based design to enhance the robustness for matched uncertain Stackelberg-Nash game in [40]. However, it is worth noting that in multiplayer systems and complex applications, there often exist mismatched uncertainties [41], which can be more general and widespread.

Motivated by the above observations and literature studies, this paper develops a learning-based control approach for a class of continuous-time nonlinear hierarchical multiplayer systems subject to mismatched uncertainties. By optimizing the strategies of the leader and followers, this approach can lead to more effective and stable decision-making in dynamic hierarchical multiplayer environments. The major contributions are summarized as follows:

(i) This paper considers a new class of problems which involve a strategic decision-making process with mismatched uncertainties. Particularly, the leader in this type of problem has a distinct advantage to make the decisions ahead of the followers, in contrast to the simultaneous-move multiplayer decision-making process [18], [20], [23], [41]. Furthermore, we formulate this robust hierarchical control problem as a Stackelberg-Nash game integrated with the hierarchical robust-optimal transformation to facilitate the learning-based design. The equivalence of the developed transformation is discussed explicitly. Compared to the two-player Stackelberg game problems [34], [35], this work is more complex since it involves one leader and multiple followers whose interactions lead to increased coupling relationships.

(ii) The robust controller is designed with assistance from the transformed hierarchical multiplayer system. The theoretical analysis ensures that the derived control policies can achieve Stackelberg-Nash equilibrium which also serves as the solution of the original robust problem. Comparing with the existing works [36], [39], [40], this paper focuses on the design with mismatched uncertainties, which is more challenging due to the increased difficulties in predicting system dynamics and the additional couplings among players introduced by the auxiliary inputs.

(iii) A two-level neural-RL-based method is developed to automatically learn the solution. Specifically, a critic network is established for each player, i.e., the leader at the high level and each follower at the low level, to estimate the performance index and assist in calculating the control policies. The designed critic networks are updated in a hierarchical fashion. The stability of the developed online learning process is also provided to ensure the learning performance.

The rest of this paper is organized as follows. In Section II, the problem of hierarchical multiplayer decision-making with mismatched uncertainties is formulated. Section III provides the development of the robust control process with theoretical proof. A two-level neural-RL-based design is presented in Section IV to update the performance index and calculate the control policy for each player. In Section V, the numerical examples are given to demonstrate the effectiveness of the developed approach. Finally, Section VI concludes this paper.

II. Mismatched Uncertainties in Hierarchical Multiplayer Systems

Consider the continuous-time nonlinear differential game with a group of $N+1$ players $\mathcal{P}=\{0,1,\cdots,N\}$, where player 0 is the leader and other players $\mathcal{F}=\{1,2,\cdots,N\}$ are the followers. We assume that the players take different hierarchical roles in the decision-making process. Specifically, the leader is in a dominant position and can determine the policy first in the hierarchy, while the followers have equal status and determine their responses simultaneously. The system function of this hierarchical multiplayer system can be described as

$$\dot{x}=f(x)+h_0(x)u_0+\varepsilon_0(x)+\sum_{i=1}^{N}h_i(x)u_i+\sum_{i=1}^{N}\varepsilon_i(x) \quad (1)$$

where $x\in\mathbb{R}^n$ is the state vector, $u_0\in\mathbb{R}^{p_0}$ and $u_i\in\mathbb{R}^{p_i}$ are the policies controlled by the leader and ith follower respectively, $f(x)\in\mathbb{R}^n$ is the system drift dynamics, $h_0(x)\in\mathbb{R}^{n\times p_0}$ and $h_i(x)\in\mathbb{R}^{n\times p_i}$ are the input dynamics of the leader and ith follower respectively, and $\varepsilon_0(x)\in\mathbb{R}^n$ and $\varepsilon_i(x)\in\mathbb{R}^n$ are the unknown uncertainties applied on the leader and ith follower respectively with $\varepsilon_0(x)=d_0(x)\xi_0(x)$ and $\varepsilon_i(x)=d_i(x)\xi_i(x)$. Assume that each unknown uncertainty is upper bounded by a known function, that is, $\|\varepsilon_0(x)\|\le\varepsilon_{M,0}(x)$ and $\|\varepsilon_i(x)\|\le\varepsilon_{M,i}(x)$. Therefore, considering all the uncertainties applied on the system, we define $\mathcal{A}_\varepsilon^2(x)\triangleq\sum_{j=0}^{N}\varepsilon_{M,j}^2(x)$. In this paper, we consider the mismatched uncertainties for all the players, i.e., $h_j(x)\ne d_j(x),\forall j\in\mathcal{P}$.

Note that, comparing with the multiplayer consensus problem [20], the hierarchical multiplayer system (1) is more complex and attains coupling information. This is because the state x in (1) is determined by the policies of all players u_j, $j \in \mathcal{P}$, which leads to interdependence between their decisions and the resulting state. In contrast, the player in the conventional consensus problem operates with its own state, and each player's state evolves independently based on its policy.

Therefore, the leader-follower relationship in this paper is substantially different from that in the consensus problem. In particular, we define the leader as the dominant player who has a distinct advantage of determining its policy first, and the follower as the subordinate player who responds to the leader's decisions. The dynamic system (1) considered in this paper is commonly employed in control theory and game theory to model the interactions between a leader and multiple followers. This framework is particularly relevant in engineering fields where hierarchical decision-making and strategic interactions are crucial. In this setup, the leader's control input directly affects the system dynamics, while each follower's control input contributes to the system's overall behavior. Hence, equation (1) encapsulates the natural dynamics of the system, the influence of the leader, and the collective impact of the followers in a mismatched uncertain environment.

Because of the existence of unknown uncertainties in system (1), the communication data can not be trusted. This makes the traditional RL method hard to be applied directly. Therefore, we transform this robust control problem into an equivalent optimal stabilization design. In order to achieve this goal, we decompose the uncertain term for each player $j \in \mathcal{P}$ into two parts as the matched and mismatched elements,

$$\begin{aligned} d_j(x)\xi_j(x) =& h_j(x)h_j^+(x)d_j(x)\xi_j(x) \\ &+ (I_n - h_j(x)h_j^+(x))d_j(x)\xi_j(x) \end{aligned} \quad (2)$$

where $h_j^+(x)$ is the Moore-Penrose pseudoinverse matrix of $h_j(x)$.

By constructing the auxiliary inputs $v_0 \in \mathbb{R}^{m_0}$ and $v_i \in \mathbb{R}^{m_i}$ for the leader and ith follower, respectively, we obtain the following nominal plant as

$$\dot{x} = f(x) + h_0(x)u_0 + \mathcal{G}_0(x)v_0 + \sum_{i=1}^{N} h_i(x)u_i + \sum_{i=1}^{N} \mathcal{G}_i(x)v_i \quad (3)$$

where $\mathcal{G}_0(x) = \big(I_n - h_0(x)h_0^+(x)\big)d_0(x)$ and $\mathcal{G}_i(x) = \big(I_n - h_i(x)h_i^+(x)\big)d_i(x)$. Comparing with (1), the transformed system (3) also involves multiple players in hierarchy, that is, the player 0 as the leader and the other players $i \in \mathcal{F}$ as the followers. In this paper, we will demonstrate that this transformation is equivalent when an appropriate cost function is established for each player (see Theorem 1).

Assume the system (3) is controllable. Construct the augment control policy as $\mathcal{U}_j = [u_j^T, v_j^T]^T, \forall j \in \mathcal{P}$. Then, we design the cost function associated to player j as

$$\begin{aligned} J_j(x,\mathcal{U}_j,\mathcal{U}_{-j}) = \int_0^{\infty} \Big\{ & \mathcal{T}_j^2(x(\tau)) + \mathcal{A}_\varepsilon^2(x(\tau)) \\ & + \Lambda_j(x(\tau),\mathcal{U}_j(\tau),\mathcal{U}_{-j}(\tau)) \Big\} d\tau \end{aligned} \quad (4)$$

where $\mathcal{T}_j(x)$ is the design parameter, $\mathcal{U}_{-j} = [u_{-j}^T, v_{-j}^T]^T$ with $u_{-j} = \{u_k | \ k \in \mathcal{P}, k \neq j\}$ and $v_{-j} = \{v_k | \ k \in \mathcal{P}, k \neq j\}$ are the sets of control policies from the neighbors of agent j, and $\Lambda_j(x,\mathcal{U}_j,\mathcal{U}_{-j})$ is the utility function. Note that the cost function (4) contains coupling information since the system state x is driven by the control policies of all the players.

Since the leader makes its decision with consideration of responses of all the followers, we define

$$\begin{aligned} &\Lambda_0(x,\mathcal{U}_0,\mathcal{U}_{-0}) \\ &\quad = x^T Q_0 x + \Big\| u_0 + \sum_{i=1}^{N} \alpha_{i1} u_i \Big\|_{R_0}^2 + \Big\| v_0 + \sum_{i=1}^{N} \alpha_{i2} v_i \Big\|_{Y_0}^2 \end{aligned} \quad (5)$$

where $Q_0 > 0$, $R_0 > 0$, and $Y_0 > 0$ are the symmetric matrices with appropriate dimensions, and $\alpha_{i1} \in \mathbb{R}^{p_0 \times p_i}$, $\alpha_{i2} \in \mathbb{R}^{m_0 \times m_i}$ denote the coupling coefficients of follower i to the leader.

For follower $i \in \mathcal{F}$, we define

$$\begin{aligned} &\Lambda_i(x,\mathcal{U}_i,\mathcal{U}_{-i}) \\ &\qquad = x^T Q_i x + \Big\| u_i + \beta_{i1} u_0 \Big\|_{R_i}^2 + \Big\| v_i + \beta_{i2} v_0 \Big\|_{Y_i}^2 \end{aligned} \quad (6)$$

where $Q_i > 0$, $R_i > 0$, and $Y_i > 0$ are the symmetric matrices with appropriate dimensions, and $\beta_{i1} \in \mathbb{R}^{p_i \times p_0}$, $\beta_{i2} \in \mathbb{R}^{m_i \times m_0}$ denote the coupling coefficients of the leader to the ith follower.

Definition 1 *(Stackelberg-Nash Equilibrium):* If there exists a mapping $\bar{\mathcal{M}}_i : \mathcal{U}_0 \to \mathcal{U}_i, i \in \mathcal{F}$, such that for a given control policy of the leader $\mathcal{U}_0$, $\bar{\mathcal{U}}_i = \bar{\mathcal{M}}_i(\mathcal{U}_0)$ is the optimal control policy for the ith follower. The set $\{\bar{\mathcal{U}}_0, \bar{\mathcal{U}}_1, \cdots, \bar{\mathcal{U}}_N\}$ is considered to constitute the Stackelberg-Nash equilibrium, if, for any $\mathcal{U}_0$ and $\mathcal{U}_i, i \in \mathcal{F}$,

$$J_i(x,\bar{\mathcal{M}}_i(\mathcal{U}_0),\bar{\mathcal{M}}_{-i}(\mathcal{U}_0)) \le J_i(x,\mathcal{U}_i,\bar{\mathcal{M}}_{-i}(\mathcal{U}_0)) \quad (7)$$

$$J_0(x,\bar{\mathcal{U}}_0,\bar{\mathcal{M}}_{-0}(\bar{\mathcal{U}}_0)) \le J_0(x,\mathcal{U}_0,\bar{\mathcal{M}}_{-0}(\mathcal{U}_0)) \quad (8)$$

where $\bar{\mathcal{M}}_{-i}(\cdot) = \{\bar{\mathcal{M}}_k(\cdot) | \ k \in \mathcal{F}, k \neq i\}$ and $\bar{\mathcal{M}}_{-0}(\cdot) = \{\bar{\mathcal{M}}_k(\cdot) | \ k \in \mathcal{F}\}$.

Condition (7) indicates that the ith follower observes the leader's policy $\mathcal{U}_0$ and reacts optimally to it, assuming that all other followers choose the control policies $\bar{\mathcal{M}}_{-i}(\mathcal{U}_0)$. This implies that the set $\{\bar{\mathcal{M}}_1(\mathcal{U}_0), \bar{\mathcal{M}}_2(\mathcal{U}_0), \cdots, \bar{\mathcal{M}}_N(\mathcal{U}_0)\}$ for all the followers achieve a Nash equilibrium. On the other hand, condition (8) characterizes the Stackelberg equilibrium for the leader who is desired to find a policy $\bar{\mathcal{U}}_0$ such that the followers' best responses to this given $\bar{\mathcal{U}}_0$ result in the minimal cost function J_0. Therefore, the Stackelberg equilibrium for the leader and the Nash equilibrium for the followers are interdependent.

In this way, by establishing an auxiliary input and developing the appropriate cost function for leader and followers respectively, we formulate the hierarchical multiplayer decision-making problem with mismatched uncertainties into a Stackelberg-Nash game in conjunction with a hierarchical robust-optimal transformation.

III. Robust Control Design

A. Equivalent Analysis of the Developed Hierarchical Robust-Optimal Transformation

Define the performance index for each player $j \in \mathcal{P}$ as

$$V_j(x) = \int_0^\infty \Big\{\mathcal{T}_j^2(x(\tau)) + \mathcal{A}_\varepsilon^2(x(\tau)) + \Lambda_j(x(\tau),\mathcal{U}_j(\tau),\mathcal{U}_{-j}(\tau))\Big\}d\tau. \quad (9)$$

Comparing with the cost function (4), we have $V_j(x) = J_j(x,\mathcal{U}_j,\mathcal{U}_{-j})$. Therefore, considering the coupling relationship between the leader and followers, the Hamiltonian for each player can be provided as

$$\mathcal{H}_j(x,\nabla V_j,\mathcal{U}_j,\mathcal{U}_{-j}) = \mathcal{T}_j^2(x) + \mathcal{A}_\varepsilon^2(x) + \Lambda_j(x,\mathcal{U}_j,\mathcal{U}_{-j}) + \nabla V_j^T(x)\left(f(x) + \sum_{k=0}^{N} h_k(x)u_k + \sum_{k=0}^{N}\mathcal{G}_k(x)v_k\right) \quad (10)$$

where $\nabla V_j(x) = \partial V_j(x)/\partial x$. Then, we have the optimal performance index as

$$V_j^*(x) = \min_{\mathcal{U}_j}\int_0^\infty \Big\{\mathcal{T}_j^2(x(\tau)) + \mathcal{A}_\varepsilon^2(x(\tau)) + \Lambda_j(x(\tau),\mathcal{U}_j(\tau),\mathcal{U}_{-j}(\tau))\Big\}d\tau \quad (11)$$

which satisfies the coupled HJ equation

$$\mathcal{H}_j(x,\nabla V_j^*,\mathcal{U}_j^*,\mathcal{U}_{-j}^*) = 0 \quad (12)$$

where $\nabla V_j^* = \partial V_j^*(x)/\partial x$.

Define $\mathcal{U}_i^{\mathcal{U}_0}$ as the optimal control action of follower i given $\mathcal{U}_0$, and $\mathcal{U}_{-i}^{\mathcal{U}_0} = \{\mathcal{U}_k^{\mathcal{U}_0} | k \in \mathcal{F}, k \neq i\}$. Assume $\nabla V_i^{\mathcal{U}_0}$ is the performance index of follower i given control actions $\mathcal{U}_0$ and $\mathcal{U}_{-i}^{\mathcal{U}_0}$. Hence, $\mathcal{U}_i^{\mathcal{U}_0}$ can be provided as

$$\mathcal{U}_i^{\mathcal{U}_0} = \arg\min_{\mathcal{U}_i}\mathcal{H}_i(x,\nabla V_i^{\mathcal{U}_0},\mathcal{U}_i,\mathcal{U}_{-i}). \quad (13)$$

Substitute (6) into (10) for follower $i \in \mathcal{F}$, and follow the first-order optimality condition. We have $\partial\mathcal{H}_i/\partial\mathcal{U}_i = 0_{(p_i+m_i)}$, that is,

$$\mathcal{U}_i^{\mathcal{U}_0} = \begin{bmatrix} u_i^{\mathcal{U}_0} \\ v_i^{\mathcal{U}_0}\end{bmatrix} = \begin{bmatrix} -\beta_{i1}u_0 - \frac{1}{2}R_i^{-1}h_i^T(x)\nabla V_i^{\mathcal{U}_0}(x) \\ -\beta_{i2}v_0 - \frac{1}{2}Y_i^{-1}\mathcal{G}_i^T(x)\nabla V_i^{\mathcal{U}_0}(x)\end{bmatrix}. \quad (14)$$

Note that if $\mathcal{U}_0 = [u_0^T, v_0^T]^T = [u_0^{*T}, v_0^{*T}]^T = \mathcal{U}_0^*$, we have $V_i^{\mathcal{U}_0^*} = V_i^*$ and $\mathcal{U}_i^{\mathcal{U}_0^*} = \mathcal{U}_i^*$.

In the Stackelberg-Nash game, the leader makes its decision first with the consideration of all the followers' responses. Therefore, the control policy of leader can be provided as

$$\mathcal{U}_0^* = \arg\min_{\mathcal{U}_0}\mathcal{H}_0(x,\nabla V_0^*,\mathcal{U}_0,\mathcal{U}_{-0}) \quad (15)$$

where $\mathcal{U}_{-0}$ represents the control policy from all the followers. Substitute (5) into (10), and notice the fact that $\mathcal{U}_i^{\mathcal{U}_0}$ is provided in (14). Then, the first-order optimality condition $\partial\mathcal{H}_0/\partial\mathcal{U}_0 = 0_{(p_0+m_0)}$ yields that

$$\mathcal{U}_0^* = \begin{bmatrix} u_0^* \\ v_0^*\end{bmatrix} = \begin{bmatrix} -\frac{1}{2}C_1\nabla V_0^*(x) + \frac{1}{2}\mathcal{F}_1\sum_{i=1}^N \alpha_{i1}R_i^{-1}h_i^T(x)\nabla V_i^*(x) \\ -\frac{1}{2}C_2\nabla V_0^*(x) + \frac{1}{2}\mathcal{F}_2\sum_{i=1}^N \alpha_{i2}Y_i^{-1}\mathcal{G}_i^T(x)\nabla V_i^*(x)\end{bmatrix} \quad (16)$$

where

$$\mathcal{F}_1 = I_{p_0} - \sum_{i=1}^N \alpha_{i1}\beta_{i1}, \quad \mathcal{F}_2 = I_{m_0} - \sum_{i=1}^N \alpha_{i2}\beta_{i2},$$

$$C_1 = \left(\mathcal{F}_1^T R_0\mathcal{F}_1\right)^{-1}\left(h_0(x) - \sum_{i=1}^N h_i(x)\beta_{i1}\right)^T,$$

$$C_1 = \left(\mathcal{F}_2^T Y_0\mathcal{F}_2\right)^{-1}\left(\mathcal{G}_0(x) - \sum_{i=1}^N \mathcal{G}_i(x)\beta_{i2}\right)^T.$$

Note that, by choosing appropriate α_{i1}, α_{i2}, β_{i1} and β_{i2}, we have $\sum_{i=1}^N \alpha_{i1}\beta_{i1} \neq I_{p_0}$ and $\sum_{i=1}^N \alpha_{i2}\beta_{i2} \neq I_{m_0}$. Substituting $\mathcal{U}_0^* = [u_0^{*T}, v_0^{*T}]^T$ into (14), we obtain the optimal response for follower $i \in \mathcal{F}$ under the optimal policy $\mathcal{U}_0^*$ of the leader and $\mathcal{U}_{-i}^*$ of the other followers as

$$\mathcal{U}_i^* = \begin{bmatrix} u_i^* \\ v_i^*\end{bmatrix} = \begin{bmatrix} -\beta_{i1}u_0^* - \frac{1}{2}R_i^{-1}h_i^T(x)\nabla V_i^*(x) \\ -\beta_{i2}v_0^* - \frac{1}{2}Y_i^{-1}\mathcal{G}_i^T(x)\nabla V_i^*(x)\end{bmatrix}. \quad (17)$$

Furthermore, by substituting (16) and (17) into (12), we obtain the coupled HJ equation for the leader and ith follower, respectively.

Note that this design is fundamentally different from the conventional simultaneous-move multiplayer game. Here, we consider a hierarchical decision-making process which attains coupling information: the control policy (16) of the leader involves the responses of all the followers and can make the decision first, while the control policy (17) of the ith follower also exists an additional term related to the leader.

Now, we provide that the control policies (16) and (17) are the results of the hierarchical multiplayer system (1) with mismatched uncertainties, which means the developed hierarchical robust-optimal transformation from (1) to (3) is equivalent.

Theorem 1: Consider the auxiliary nominal system (3) with the designed performance index (9) for each player $j \in \mathcal{P}$. Let $V_j^*(x)$ be a solution of the coupled HJ equation (12) with the parameter $\mathcal{T}_j^2(x)$ chosen as

$$\mathcal{T}_j^2(x) = \frac{1}{2}\nabla V_j^{*T}(x)\nabla V_j^*(x) + \mathcal{K}_{v_j}^2(x) \quad (18)$$

where $\mathcal{K}_{v_j}^2(x)$ is an upper bound as $\mathcal{K}_{v_j}^2(x) \geq \left\|\sum_{k=0}^N \mathcal{G}_k(x)v_k^*\right\|^2$. The optimal control policy $\mathcal{U}_0^*$ is given as (16) for the leader and $\mathcal{U}_i^*$ as (17) for the follower $i \in \mathcal{F}$. Then, we have that the control policies $\mathcal{U}_0^*$ and $\mathcal{U}_i^*$ can asymptotically stabilize the hierarchical multiplayer system (1) with mismatched uncertainties.

Proof: Consider the Lyapunov function for each player $j \in \mathcal{P}$ as $L_j(x) = V_j^*(x)$. Taking the derivative of $L_j(x)$ along the hierarchical system trajectory, we obtain

$$\dot{L}_j(x) = \nabla L_j^T(x)\left(f(x) + \sum_{k=0}^N h_k(x)u_k^* + \sum_{k=0}^N d_k(x)\xi_k(x)\right). \quad (19)$$

Based on the transformation design, we have

$$\begin{aligned}\dot{L}_j(x) =& \nabla L_j^T(x)\Big(f(x) + \sum_{k=0}^{N} h_k(x)u_k^* + \sum_{k=0}^{N}\big(I_n - h_k(x)\\ &\cdot h_k^+(x)\big)d_k(x)v_k^*\Big) + \nabla L_j^T(x)\Big(\sum_{k=0}^{N} d_k(x)\xi_k(x)\\ &- \sum_{k=0}^{N}\big(I_n - h_k(x)h_k^+(x)\big)d_k(x)v_k^*\Big). \qquad (20)\end{aligned}$$

From (12), we get

$$\begin{aligned}\dot{L}_j(x) =& -\frac{1}{2}\nabla L_j^T(x)\nabla L_j(x) - \mathcal{K}_{v_j}^2(x) - \mathcal{A}_\varepsilon^2(x)\\ &- \Lambda_j(x,\mathcal{U}_j^*,\mathcal{U}_{-j}^*) + \nabla L_j^T(x)\sum_{k=0}^{N} d_k(x)\xi_k(x)\\ &- \nabla L_j^T(x)\sum_{k=0}^{N}\mathcal{G}_k(x)v_k^*. \qquad (21)\end{aligned}$$

We can further rewrite (21) as

$$\begin{aligned}\dot{L}_j(x) =& -\mathcal{K}_{v_j}^2(x) - \Lambda_j(x,\mathcal{U}_j^*,\mathcal{U}_{-j}^*) - \mathcal{A}_\varepsilon^2(x)\\ &- \left\|\frac{1}{2}\nabla L_j(x) - \sum_{k=0}^{N} d_k(x)\xi_k(x)\right\|^2 + \left\|\sum_{k=0}^{N} d_k(x)\xi_k(x)\right\|^2\\ &- \left\|\frac{1}{2}\nabla L_j(x) + \sum_{k=0}^{N}\mathcal{G}_j(x)v_j^*\right\|^2 + \left\|\sum_{k=0}^{N}\mathcal{G}_k(x)v_k^*\right\|^2\\ \le& -\Lambda_j(x,\mathcal{U}_j^*,\mathcal{U}_{-j}^*) - \left(\mathcal{K}_{v_j}^2(x) - \left\|\sum_{k=0}^{N}\mathcal{G}_k(x)v_k^*\right\|^2\right)\\ &- \left(\mathcal{A}_\varepsilon^2(x) - \left\|\sum_{k=0}^{N} d_k(x)\xi_k(x)\right\|^2\right). \qquad (22)\end{aligned}$$

Based on the bound conditions $\left\|\sum_{k=0}^{N} d_k(x)\xi_k(x)\right\|^2 \le \mathcal{A}_\varepsilon^2(x)$ and $\left\|\sum_{k=0}^{N}\mathcal{G}_k(x)v_k^*\right\|^2 \le \mathcal{K}_{v_j}^2(x)$, we can obtain $\dot{L}_j(x) \le -\Lambda_j(x,\mathcal{U}_j^*,\mathcal{U}_{-j}^*) < 0, \forall x \ne 0$. Therefore, the designed control policies $\mathcal{U}_j^*$ can asymptotically stabilize the hierarchical multiplayer system (1) with mismatched uncertainties, which demonstrates the equivalence of the developed hierarchical robust-optimal transformation. ■

Note that by integrating (18) into (9), we observe that the performance index for each player j requires the information of $v_k, k \in \mathcal{P}$. However, only the leader can access the inputs from all the followers. In this design, we assume each follower transmits $\vartheta_i^2(x) = \mathcal{G}_i(x)v_i^*$, $i \in \mathcal{F}$, to the leader, and then the leader sends back $\mathcal{K}_{v_j}^2 = \sum_{k=0}^{N}\vartheta_k^2$ to each follower.

B. Stackelberg-Nash Equilibrium of the Designed Policies

Now, we show that the designed control policies $\{\mathcal{U}_0^*, \mathcal{U}_1^*, \cdots, \mathcal{U}_N^*\}$ can achieve Stackelberg-Nash equilibrium.

Theorem 2: Assume $V_j^*(x)$ as the solution of the coupled HJ equation (12) with the parameter $\mathcal{T}_j^2(x)$ defined in (18). Then the designed control policies $\{\mathcal{U}_0^*, \mathcal{U}_1^*, \cdots, \mathcal{U}_N^*\}$ can achieve Stackelberg-Nash equilibrium.

Proof: According to Theorem 1, the system (1) under the developed control policies $\mathcal{U}_j^*$ is asymptotically stable. Set $\mathcal{E}_j(x) = V_j^*(x)$. The cost function (4) for each player $j \in \mathcal{P}$ can be rewritten as

$$\begin{aligned}J_j(x,\mathcal{U}_j,\mathcal{U}_{-j}) =& \int_0^\infty \Big\{\mathcal{T}_j^2(x) + \mathcal{A}_\varepsilon^2(x) + \Lambda_j(x,\mathcal{U}_j,\mathcal{U}_{-j})\Big\}d\tau\\ &+ \mathcal{E}_j(x(0)) + \int_0^\infty \dot{\mathcal{E}}_j(x)d\tau\\ =& \int_0^\infty \mathcal{H}_j(x,\nabla\mathcal{E}_j,\mathcal{U}_j,\mathcal{U}_{-j})d\tau + \mathcal{E}_j(x(0)). \qquad (23)\end{aligned}$$

For the leader, assume all the followers choose the optimal control action $\mathcal{U}_{-0}^{\mathcal{U}_0}$ given $\mathcal{U}_0$, where $\mathcal{U}_{-0}^{\mathcal{U}_0} = \{\mathcal{U}_k^{\mathcal{U}_0} | k \in \mathcal{F}\}$. Based on (12), we have $\mathcal{H}_0(x, \nabla\mathcal{E}_0, \mathcal{U}_0^*, \mathcal{U}_{-0}^*) = \mathcal{H}_0(x, \nabla V_0^*, \mathcal{U}_0^*, \mathcal{U}_{-0}^*) = 0$. Therefore, we can further rewrite (23) for the leader as

$$\begin{aligned}J_0(x,\mathcal{U}_0,\mathcal{U}_{-0}^{\mathcal{U}_0}) =& \mathcal{E}_0(x) + \int_0^\infty \Big\{\mathcal{H}_0(x,\nabla\mathcal{E}_0,\mathcal{U}_0,\mathcal{U}_{-0}^{\mathcal{U}_0})\\ &- \mathcal{H}_0(x,\nabla\mathcal{E}_0,\mathcal{U}_0^*,\mathcal{U}_{-0}^*)\Big\}d\tau. \qquad (24)\end{aligned}$$

Since the goal of the control policy $\mathcal{U}_0$ is to minimize $\mathcal{H}_0$, we have $\mathcal{H}_0(x, \nabla\mathcal{E}_0, \mathcal{U}_0, \mathcal{U}_{-0}^{\mathcal{U}_0}) \ge \mathcal{H}_0(x, \nabla\mathcal{E}_0, \mathcal{U}_0^*, \mathcal{U}_{-0}^*)$, which means

$$J_0(x,\mathcal{U}_0,\mathcal{U}_{-0}^{\mathcal{U}_0}) \ge \mathcal{E}_0(x). \qquad (25)$$

According to the definition of $\mathcal{E}_0(x)$, we can easily obtain that $\mathcal{E}_0(x) = V_0^*(x) = J_0(x, \mathcal{U}_0^*, \mathcal{U}_{-0}^*)$. Therefore, it follows

$$J_0(x,\mathcal{U}_0,\mathcal{U}_{-0}^{\mathcal{U}_0}) \ge J_0(x,\mathcal{U}_0^*,\mathcal{U}_{-0}^*). \qquad (26)$$

This proves that the Stackelberg equilibrium holds for the leader in this hierarchical multiplayer decision-making design.

The reminding provides the Nash equilibrium of the followers' policies $\{\mathcal{U}_1^*, \mathcal{U}_2^*, \cdots, \mathcal{U}_N^*\}$. Considering (23), the first term can be further derived as

$$\begin{aligned}\mathcal{H}_j(x,\nabla\mathcal{E}_j,\mathcal{U}_j,\mathcal{U}_{-j}) =& \mathcal{H}_j(x,\nabla\mathcal{E}_j,\mathcal{U}_j^*,\mathcal{U}_{-j}^*) + \Lambda_j(x,\mathcal{U}_j,\mathcal{U}_{-j})\\ &- \Lambda_j(x,\mathcal{U}_j^*,\mathcal{U}_{-j}^*) + \nabla\mathcal{E}_j^T(x)\sum_{k=0}^{N} h_k(x)(u_k - u_k^*)\\ &+ \nabla\mathcal{E}_j^T(x)\sum_{k=0}^{N}\mathcal{G}_k(x)(v_k - v_k^*). \qquad (27)\end{aligned}$$

For follower $i \in \mathcal{F}$, assume that the transition of the control policy $\mathcal{U}_i$ given $\mathcal{U}_0$ can be provided as $\mathcal{U}_i = \mu_i(\mathcal{U}_0)$. Hence, substituting (27) into (23), we obtain

$$\begin{aligned}J_i(x,\mu_i(\mathcal{U}_0^*),\mu_{-i}^*(\mathcal{U}_0^*)) =& \mathcal{E}_i(x) + \int_0^\infty \Big\{\left\|u_i + \beta_i u_0^*\right\|_{R_i}^2\\ &+ \left\|v_i + \beta_i v_0^*\right\|_{Y_i}^2 - \left\|u_i^* + \beta_i u_0^*\right\|_{R_i}^2 - \left\|v_i^* + \beta_i v_0^*\right\|_{Y_i}^2\\ &+ \nabla\mathcal{E}_i^T(x)h_i(x)(u_i - u_i^*) + \nabla\mathcal{E}_i^T(x)\mathcal{G}_i(x)(v_i - v_i^*)\Big\}d\tau.\\ & \qquad (28)\end{aligned}$$

Note that, comparing with (27), the last two terms in (28) only include the control policy $\mathcal{U}_i = [u_i^T, v_i^T]^T$. This is because we consider the other followers choose the optimal control policies $\mu_{-i}^*(\mathcal{U}_0^*) = \{\mathcal{U}_k^* | k \in \mathcal{F}, k \ne i\}$.

Based on (17), we have

$$\nabla\mathcal{E}_i^T(x)h_i(x) = -2(u_i^* + \beta_i u_0^*)^T R_i, \qquad (29)$$

$$\nabla\mathcal{E}_i^T(x)\mathcal{G}_i(x) = -2(v_i^* + \beta_i v_0^*)^T Y_i. \tag{30}$$

Substituting (29) and (30) into (28), we obtain

$$\begin{aligned}
J_i(x, \mu_i(\mathcal{U}_0^*), \mu_{-i}^*(\mathcal{U}_0^*)) = \mathcal{E}_i(x) + \int_0^\infty \Big\{ & \left\| u_i + \beta_i u_0^* \right\|^2_{R_i} \\
& - \left\| u_i^* + \beta_i u_0^* \right\|^2_{R_i} - 2(u_i^* + \beta_i u_0^*)^T R_i (u_i - u_i^*) \\
& + \left\| v_i + \beta_i v_0^* \right\|^2_{Y_i} - \left\| v_i^* + \beta_i v_0^* \right\|^2_{Y_i} - 2(v_i^* + \beta_i v_0^*)^T \\
& \cdot Y_i(v_i - v_i^*) \Big\} d\tau \\
= \mathcal{E}_i(x) + \int_0^\infty \Big\{ & (u_i - u_i^*)^T R_i (u_i - u_i^*) \\
& + (v_i - v_i^*)^T Y_i (v_i - v_i^*) \Big\} d\tau.
\end{aligned} \tag{31}$$

It follows $J_i(x, \mu_i(\mathcal{U}_0^*), \mu_{-i}^*(\mathcal{U}_0^*)) \geq \mathcal{E}_i(x)$. Considering the fact $\mathcal{E}_i(x) = J_i(x, \mu_i^*(\mathcal{U}_0^*), \mu_{-i}^*(\mathcal{U}_0^*))$, we have

$$J_i(x, \mu_i(\mathcal{U}_0^*), \mu_{-i}^*(\mathcal{U}_0^*)) \geq J_i(x, \mu_i^*(\mathcal{U}_0^*), \mu_{-i}^*(\mathcal{U}_0^*)). \tag{32}$$

Therefore, the Nash equilibrium holds for all the followers $\{\mathcal{U}_1^*, \mathcal{U}_2^*, \cdots, \mathcal{U}_N^*\} = \{\mu_1^*(\mathcal{U}_0^*), \mu_2^*(\mathcal{U}_0^*), \cdots, \mu_N^*(\mathcal{U}_0^*)\}$.

Combining (26) and (32), and based on Definition 1, we have that the designed control policies can achieve Stackelberg-Nash equilibrium with $\bar{\mathcal{U}}_0 = \mathcal{U}_0^*$, $\bar{\mathcal{M}}_i(\bar{\mathcal{U}}_0) = \mu_i^*(\mathcal{U}_0^*)$ and $\bar{\mathcal{M}}_{-0}(\mathcal{U}_0) = \mathcal{U}_{-0}^{\mathcal{U}_0}$. This competes the proof. ■

IV. A Two-Level Neural-RL-Based Method

Based on Theorem 1-2, we can solve the robust control problem of hierarchical mismatched uncertain system (1) with the assistance of the designed auxiliary system (3). This is achieved by addressing a set of coupled HJ equation (12). However, solving (12) directly is difficult due to the complex coupling structure associated with the nonlinear dynamic system evolution. The established auxiliary inputs further increase the couplings among players, which makes the problem more challenging. Therefore, this paper proposes the development of a two-level neural-RL-based method to adaptively learn the solution.

A. Adaptive Two-Level Neural-RL-Based Control Structure

An adaptive two-level neural-RL-based control method is designed. Specifically, a critic network is established for the leader at the high level to reconstruct the performance index as

$$V_0^*(x) = \omega_{c,0}^{*T}\phi_{c,0}(x) + \sigma_{c,0}(x) \tag{33}$$

where $\omega_{c,0}^*$, $\phi_{c,0}(x)$, and $\sigma_{c,0}(x)$ are the ideal critic network weights, activation function, and bounded critic network error, respectively, for the leader. Then, the derivative of $V_0^*(x)$ can be provided as

$$\nabla V_0^*(x) = \nabla\phi_{c,0}^T(x)\omega_{c,0}^* + \nabla\sigma_{c,0}(x) \tag{34}$$

where $\nabla\phi_{c,0}(x) = \frac{\partial\phi_{c,0}(x)}{\partial x}$ and $\nabla\sigma_{c,0}(x) = \frac{\partial\sigma_{c,0}(x)}{\partial x}$. Substituting (34) into (16), we obtain the control policy for leader at the high level as

$$\mathcal{U}_0^* = [u_0^{*T}, v_0^{*T}]^T; \tag{35}$$

$$\begin{aligned}
u_0^* = & -\frac{1}{2}C_1\Big(\nabla\phi_{c,0}^T(x)\omega_{c,0}^* + \nabla\sigma_{c,0}(x)\Big) + \frac{1}{2}\mathcal{F}_1\sum_{i=1}^N \alpha_{i1}R_i^{-1} \\
& \cdot h_i^T(x)\Big(\nabla\phi_{c,i}^T(x)\omega_{c,i}^* + \nabla\sigma_{c,i}(x)\Big)
\end{aligned} \tag{36}$$

$$\begin{aligned}
v_0^* = & -\frac{1}{2}C_2\Big(\nabla\phi_{c,0}^T(x)\omega_{c,0}^* + \nabla\sigma_{c,0}(x)\Big) + \frac{1}{2}\mathcal{F}_2\sum_{i=1}^N \alpha_{i2}Y_i^{-1} \\
& \cdot \mathcal{G}_i^T(x)\Big(\nabla\phi_{c,i}^T(x)\omega_{c,i}^* + \nabla\sigma_{c,i}(x)\Big).
\end{aligned} \tag{37}$$

Furthermore, establish the critic network for each follower at the low level with $\omega_{c,i}^*$, $\phi_{c,i}(x)$, and $\sigma_{c,i}(x)$ as the ideal critic network weights, activation function, and bounded critic network error, respectively, for follower $i \in \mathcal{F}$. Therefore, the performance index $V_i^*(x)$ is provided as $V_i^*(x) = \omega_{c,i}^{*T}\phi_{c,i}(x) + \sigma_{c,i}(x)$ with the derivative as

$$\nabla V_i^*(x) = \nabla\phi_{c,i}^T(x)\omega_{c,i}^* + \nabla\sigma_{c,i}(x) \tag{38}$$

where $\nabla\phi_{c,i}(x) = \frac{\partial\phi_{c,i}(x)}{\partial x}$ and $\nabla\sigma_{c,i}(x) = \frac{\partial\sigma_{c,i}(x)}{\partial x}$. Substituting (38) into (17), the control policy for follower $i \in \mathcal{F}$ at the low level is derived as

$$\mathcal{U}_i^* = [u_i^{*T}, v_i^{*T}]^T; \tag{39}$$

$$u_i^* = -\beta_{i1}u_0^* - \frac{1}{2}R_i^{-1}h_i^T(x)\Big(\nabla\phi_{c,i}^T(x)\omega_{c,i}^* + \nabla\sigma_{c,i}(x)\Big) \tag{40}$$

$$v_i^* = -\beta_{i2}v_0^* - \frac{1}{2}Y_i^{-1}\mathcal{G}_i^T(x)\Big(\nabla\phi_{c,i}^T(x)\omega_{c,i}^* + \nabla\sigma_{c,i}(x)\Big). \tag{41}$$

Considering (35) and (39), however, one realizes the ideal weights $\omega_{c,0}^*$ and $\omega_{c,i}^*$ are difficult or impossible to achieve. Therefore, we consider the current estimated critic network weights $\omega_{c,0}$ for leader and $\omega_{c,i}$ for follower $i \in \mathcal{F}$. Then, the estimated performance index is provided as

$$\hat{V}_0(x) = \omega_{c,0}^T\phi_{c,0}(x), \tag{42}$$

$$\hat{V}_i(x) = \omega_{c,i}^T\phi_{c,i}(x), \quad i \in \mathcal{F} \tag{43}$$

where $\hat{V}_0(x)$ is the estimated performance index for leader which is learned at the high level and $\hat{V}_i(x)$ is the estimated performance index for ith follower which is learned at the low level. Therefore, the corresponding derivatives are $\nabla\hat{V}_0(x) = \nabla\phi_{c,0}^T(x)\omega_{c,0}$ and $\nabla\hat{V}_i(x) = \nabla\phi_{c,i}^T(x)\omega_{c,i}$. It follows the estimated control policy for leader at the high level as

$$\mathcal{U}_0 = [u_0^T, v_0^T]^T; \tag{44}$$

$$\begin{aligned}
u_0 = & -\frac{1}{2}C_1\nabla\phi_{c,0}^T(x)\omega_{c,0} + \frac{1}{2}\mathcal{F}_1\sum_{i=1}^N \alpha_{i1}R_i^{-1}h_i^T(x) \\
& \cdot \nabla\phi_{c,i}^T(x)\omega_{c,i}
\end{aligned} \tag{45}$$

$$\begin{aligned}
v_0 = & -\frac{1}{2}C_2\nabla\phi_{c,0}^T(x)\omega_{c,0} + \frac{1}{2}\mathcal{F}_2\sum_{i=1}^N \alpha_{i2}Y_i^{-1}\mathcal{G}_i^T(x) \\
& \cdot \nabla\phi_{c,i}^T(x)\omega_{c,i}
\end{aligned} \tag{46}$$

and for follower $i \in \mathcal{F}$ at the low level as

$$\mathcal{U}_i = [u_i^T, v_i^T]^T; \tag{47}$$

$$u_i = -\beta_{i1}u_0 - \frac{1}{2}R_i^{-1}h_i^T(x)\nabla\phi_{c,i}^T(x)\omega_{c,i} \tag{48}$$

$$v_i = -\beta_{i2}v_0 - \frac{1}{2}Y_i^{-1}\mathcal{G}_i^T(x)\nabla\phi_{c,i}^T(x)\omega_{c,i}. \tag{49}$$

Considering (42) and (43), we can unify the performance index for all the players as $\hat V_j(x) = \omega_{c,j}^T\phi_{c,j}(x), j \in \mathcal{P}$. Hence, based on the neural network design, we have the unified version of the estimated Hamiltonian for player $j \in \mathcal{P}$ as

$$\mathcal{H}_j(x,\nabla\hat V_j,\mathcal{U}_j,\mathcal{U}_{-j}) = \mathcal{T}_j^2(x) + \mathcal{A}_\varepsilon^2(x) + \Lambda_j(x,\mathcal{U}_j,\mathcal{U}_{-j}) + \nabla\phi_{c,j}^T(x)\omega_{c,j}\left(f(x) + \sum_{k=0}^{N} h_k(x)u_k + \sum_{k=0}^{N}\mathcal{G}_k(x)v_k\right). \quad (50)$$

According to the coupled HJ equation (12), we define the error function for the critic network as $e_{c,j} = \mathcal{H}_j(x,\nabla\hat V_j,\mathcal{U}_j,\mathcal{U}_{-j})$ and the objective function as $E_{c,j} = \frac{1}{2}e_{c,j}^T e_{c,j}$. Hence, the critic network weights are updated as

$$\begin{aligned}\dot\omega_{c,j} &= -\eta_j \frac{1}{(D_{c,j}^T D_{c,j}+1)^2}\left(\frac{\partial E_{c,j}}{\partial\hat\omega_{c,j}}\right)\\ &= -\eta_j \frac{D_{c,j}}{(D_{c,j}^T D_{c,j}+1)^2}\Big(\mathcal{T}_j^2(x) + \mathcal{A}_\varepsilon^2(x)\\ &\qquad + \Lambda_j(x,\mathcal{U}_j,\mathcal{U}_{-j}) + D_{c,j}^T\omega_{c,j}\Big), \quad j \in \mathcal{P}\end{aligned} \quad (51)$$

where η_j is the learning rate of the critic network for player $j \in \mathcal{P}$ and $D_{c,j} = \nabla\phi_{c,j}\dot x = \nabla\phi_{c,j}\big(f(x) + \sum_{j=0}^N h_j(x)u_j + \sum_{j=0}^N \mathcal{G}_j(x)v_j\big)$ is for normalization. Note that $\omega_{c,0}$ is updated at the high level for leader and $\omega_{c,i}$ is updated at the low level for follower i. Here, we establish the critic-only structure for all the players, such that the computation load can be reduced for the multiplayer system. The detailed algorithm is provided in Algorithm 1.

B. Stability Analysis

The following theorem provides the stability analysis of the designed online-learning process.

Theorem 3: Consider the nominal hierarchical multiplayer system (3). Establish the critic network for the leader at the high level as in (42) and for the follower $i \in \mathcal{F}$ at the low level as in (43). The weights updating law is provided in (51). If the leader's control policy is designed as in (44) and the follower's control policy as in (47), then the closed-loop control design is uniformly ultimately bounded (UUB).

Proof: Define the Lyapunov function as

$$L_{sys} = \sum_{j=0}^{N} L_j(x) + \sum_{j=0}^{N} L_j(\tilde\omega_{c,j}), \quad j \in \mathcal{P} \quad (52)$$

where $\tilde\omega_{c,j} = \omega_{c,j}^* - \omega_{c,j}$, $L_j(\tilde\omega_{c,j}) = \eta_j^{-1}tr(\tilde\omega_{c,j}^T\tilde\omega_{c,j})$, and $L_j(x) = V_j^*(x)$. Define $L_{sys,j} = L_j(x) + L_j(\tilde\omega_{c,j})$. The first derivative of (52) is provided as $\dot L_{sys} = \sum_{j=0}^N \dot L_{sys,j}$. Therefore, if we have $\dot L_{sys,j} \le 0$, it follows $\dot L_{sys} \le 0$. Hence, we consider $\dot L_{sys,j} = \dot L_j(x) + \dot L_j(\tilde\omega_{c,j})$.

Based on (12), we have the first term $\dot L_j(x) = -\mathcal{T}_j^2(x) - \mathcal{A}_\varepsilon^2(x) - \Lambda_j(x,\mathcal{U}_j,\mathcal{U}_{-j}) \le 0$. Now, we consider the second term

$$\dot L_j(\tilde\omega_{c,j}) = \eta_j^{-1}tr\Bigg(\eta_j\tilde\omega_{c,j}^T\frac{D_{c,j}}{(D_{c,j}^T D_{c,j}+1)^2}\Big(D_{c,j}^T\omega_{c,j} + \mathcal{T}_j^2(x) + \mathcal{A}_\varepsilon^2(x) + \Lambda_j(x,\mathcal{U}_j,\mathcal{U}_{-j})\Big)\Bigg). \quad (53)$$

Algorithm 1 Robust Control with Two-Level Neural-RL-Based Method

```
Choose the learning time $T_\mathcal{L} > 0$ and the execution time $T_\mathcal{E} > 0$. Set parameters $Q_j$, $\mathcal{T}_j(x)$, $\mathcal{A}_\varepsilon(x)$, and $\eta_j$ for player $j \in \mathcal{P}$. Set parameters $\alpha_{i1}$, $\alpha_{i2}$, $\beta_{i1}$, and $\beta_{i2}$ for follower $i \in \mathcal{F}$. Set a threshold $\kappa_j$ for player $j \in \mathcal{P}$.
**Learning Phase**
Initialize $\omega_{c,0}$, $\omega_{c,j}$, $\mathcal{U}_0$, and $\mathcal{U}_i$.
for $t = 0 \to T_\mathcal{L}$ do
    Take actions $\mathcal{U}_0$ and $\mathcal{U}_i$ in (3), and collect $x$;
    Begin the high-level learning
    Determine $\mathcal{U}_0 = [u_0^T, v_0^T]^T$ through (44);
    Compute $\hat V_0(x)$ based on (42);
    Update $\omega_{c,0}$ based on (51) for leader $j = 0$;
    if $||\Delta\omega_{c,0}|| < \kappa_0$ then
        Stop;
    end if
    Begin the low-level learning
    Determine $\mathcal{U}_i = [u_i^T, v_i^T]^T$, $i \in \mathcal{F}$, through (47);
    Compute $\hat V_i(x)$, $i \in \mathcal{F}$, based on (43);
    Update $\omega_{c,i}$ based on (51) for follower $j = i \in \mathcal{F}$;
    if $||\Delta\omega_{c,i}|| < \kappa_i$ then
        Stop;
    end if
end for
return $\omega_{c,0}$ and $\omega_{c,i}$.
**Robust Control Phase**
for $t = 0 \to T_\mathcal{E}$ do
    Compute $u_0$ using (45);
    Compute $u_i$, $i \in \mathcal{F}$, using (48);
    Take actions $u_0$ and $u_i$ in (1), and collect $x$;
end for
```

Since the fact $\omega_{c,j} = \omega_{c,j}^* - \tilde\omega_{c,j}$, we can rewrite (53) as

$$\begin{aligned}\dot L_j(\tilde\omega_{c,j}) = \eta_j^{-1}tr\Bigg(&-\eta_j\tilde\omega_{c,j}^T\frac{D_{c,j}D_{c,j}^T}{(D_{c,j}^T D_{c,j}+1)^2}\tilde\omega_{c,j}\\ &+\eta_j\tilde\omega_{c,j}^T\frac{D_{c,j}}{(D_{c,j}^T D_{c,j}+1)^2}\Big(D_{c,j}^T\omega_{c,j}^* + \mathcal{T}_j^2(x)\\ &+\mathcal{A}_\varepsilon^2(x) + \Lambda_j(x,\mathcal{U}_j,\mathcal{U}_{-j})\Big)\Bigg).\end{aligned} \quad (54)$$

It follows

$$\begin{aligned}&\dot L_j(\tilde\omega_{c,j})\\ &\le -\left\|\frac{D_{c,j}}{D_{c,j}^T D_{c,j}+1}\right\|^2\|\tilde\omega_{c,j}\|^2 + \frac{1}{2}\Bigg(\eta_j\left\|\frac{D_{c,j}}{D_{c,j}^T D_{c,j}+1}\right\|^2\|\tilde\omega_{c,j}\|^2\\ &+\frac{\|D_{c,j}^T\omega_{c,j}^* + \mathcal{T}_j^2(x) + \mathcal{A}_\varepsilon^2(x) + \Lambda_j(x,\mathcal{U}_j,\mathcal{U}_{-j})\|^2}{\eta_j(D_{c,j}^T D_{c,j}+1)^2}\Bigg).\end{aligned} \quad (55)$$

By setting $\Phi_j^{\mathcal{D}} = \frac{D_{c,j}}{D_{c,j}^T D_{c,j}+1}$ and $\Omega_j = D_{c,j}^T\omega_{c,j}^* + \mathcal{T}_j^2(x) +$

$\mathcal{A}_{\varepsilon}^2(x) + \Lambda_j(x, \mathcal{U}_j, \mathcal{U}_{-j}) \le \bar{\Omega}_j$, we have

$$\begin{aligned} \dot{L}_j(\tilde{\omega}_{c,j}) \le & - \left\|\Phi_j^{\mathcal{D}}\right\|^2 \left\|\tilde{\omega}_{c,j}\right\|^2 + \frac{1}{2}\eta_j \left\|\Phi_j^{\mathcal{D}}\right\|^2 \left\|\tilde{\omega}_{c,j}\right\|^2 \\ & + \frac{\left\|\Omega_j\right\|^2}{2\eta_j (D_{c,j}^T D_{c,j} + 1)^2} \\ \le & - (1 - \frac{1}{2}\eta_j) \left\|\Phi_j^{\mathcal{D}}\right\|^2 \left\|\tilde{\omega}_{c,j}\right\|^2 + \frac{\bar{\Omega}_j^2}{2\eta_j}. \end{aligned} \tag{56}$$

If the following conditions hold

$$0 < \eta_j < 2, \qquad \left\|\tilde{\omega}_{c,j}\right\|^2 > \frac{\bar{\Omega}_j^2}{2\eta_j (1 - \frac{1}{2}\eta_j) \left\|\Phi_j^{\mathcal{D}}\right\|^2} \tag{57}$$

then $\dot{L}_j(\tilde{\omega}_{c,j}) < 0$. Hence, we can further derive that $\dot{L}_{sys,j} = \dot{L}_j(x) + \dot{L}_j(\tilde{\omega}_{c,j}) < 0$. This means the designed online learning process for the hierarchical multiplayer decision-making system is UUB. This concludes the proof. ■

Remark 1: This paper designs an adaptive two-level neural-RL-based approach for hierarchical multiplayer systems with mismatched uncertainties. The specific novelty of this work can be summarized as follows:

- Different from the conventional multiplayer systems [18], [20], [23], [41] where all the players respond simultaneously, this work considers Stackelberg-Nash game with a hierarchical decision-making process. The design incorporates coupling information, i.e., the leader's policy (16) involves the responses of all the followers and can act first, while each follower's policy (17) also exists an additional term related to the leader. This relationship increases complexity to the learning-based stability design and theoretical analysis for this new class of problems.
- In contrast to the Stackelberg game problem [34], [35], which contains two players only, this work concerns the hierarchical decision-making process with one leader and multiple followers. The learning-based design for such system is more challenging due to the complex interactions and increased coupling relationships among all the players.
- The designed method can automatically solve the coupled HJ equation and learn the control policy for each player. The established hierarchical structure facilitates a more organized and systematic learning process, where the leader and followers can adapt their strategies in a coordinated and flexible manner.
- Furthermore, compared to the existing related works [36], [39], [40], our designed method can also learn in a more intricate environment with mismatched, noisy, or unreliable information, which enhances the robustness and adaptability of the learning process.

Therefore, this two-level learning-based framework opens up new possibilities for applying RL to complex multiplayer systems where hierarchical decision-making is crucial. In addition, the stability proof of the online learning process ensures the viability and effectiveness of our proposed method.

V. Experiment Studies

Example 1: Consider the following six-player continuous-time nonlinear hierarchical game with mismatched uncertainties (open loop unstable),

$$\dot{x} = f(x) + h_0(x)u_0 + \varepsilon_0(x) + \sum_{i=1}^{5} h_i(x)u_i + \sum_{i=1}^{5} \varepsilon_i(x) \tag{58}$$

with

$$f(x) = \begin{bmatrix} -x_1 + x_2 + \frac{1}{2}x_1^2 x_2 \\ -x_1 - x_2 + x_1 x_2^2 + \frac{1}{4}x_2((\cos(2x_1) + 2)^2 + (\sin(4x_1^2) + 2)^2) \end{bmatrix},$$

$$h_0(x) = \begin{bmatrix} 0 \\ \cos(x_1) \end{bmatrix}, \quad h_1(x) = \begin{bmatrix} 0 \\ \cos(2x_1 + 1) \end{bmatrix},$$

$$h_2(x) = \begin{bmatrix} 0 \\ \sin(x_1 + 2) \end{bmatrix}, \quad h_3(x) = \begin{bmatrix} 0 \\ \sin(4x_1^2) + 1 \end{bmatrix},$$

$$h_4(x) = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad h_5(x) = \begin{bmatrix} 0 \\ 4\cos(x_1^2 + x_2^2) \end{bmatrix}.$$

where $x = [x_1, x_2]^T \in \mathbb{R}^2$ is the state variable and $u_j \in \mathbb{R}$ is the policy controlled by player j, $j \in \{0, 1, 2, 3, 4, 5\}$. We have the player 0 as the leader, who can take the decision first, and other players $\{1, 2, 3, 4, 5\}$ as the followers, who then respond to the leader's decision.

The term $\varepsilon_j(x) = d_j(x)\xi_j(x)$ represents the unknown uncertainty applied on the jth player with

$$\xi_j(x) = \lambda_1 x_1 \cos\left(\frac{1}{x_2 + \lambda_2}\right) + \lambda_3 x_2 \sin\left(\lambda_4 x_1 x_2\right),$$

$$d_0(x) = \begin{bmatrix} \sin(x_1^2) \\ 0 \end{bmatrix}, \quad d_1(x) = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$d_2(x) = \begin{bmatrix} \sin(x_1 + 1) \\ 0 \end{bmatrix}, \quad d_3(x) = \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix},$$

$$d_4(x) = \begin{bmatrix} \cos(2x_2^2) \\ 0 \end{bmatrix}, \quad d_5(x) = \begin{bmatrix} \sin(x_2 + 2) \\ 1 \end{bmatrix}.$$

where $\lambda_1 \in [-1, 1]$, $\lambda_2 \in [-100, 0) \bigcup (0, 100]$, $\lambda_3 \in [-1, 1]$, and $\lambda_4 \in [-100, 100]$ are the unknown parameters. Besides, since $h_j(x) \neq d_j(x)$, we know the hierarchical game contains the mismatched uncertainties.

The developed learning-based control method is applied to address this robust problem. Based on Theorem 1-2, we conclude that the problem can be effectively solved with the assistance of the auxiliary nominal plant. Accordingly, we build the nominal plant as follows

$$\dot{x} = f(x) + h_0(x)u_0 + \mathcal{G}_0(x)v_0 + \sum_{i=1}^{5} h_i(x)u_i + \sum_{i=1}^{5} \mathcal{G}_i(x)v_i \tag{59}$$

where $\mathcal{G}_j(x) = \left(I_n - h_j(x)h_j^+(x)\right)d_j(x)$ and $h_j^+(x) = \left(h_j^T(x)h_j(x)\right)^{-1}h_j^T(x)$, $j \in \{0, 1, 2, 3, 4, 5\}$. Therefore, we have $\mathcal{G}_0(x) = \begin{bmatrix} \sin(x_1^2) \\ 0 \end{bmatrix}$, $\mathcal{G}_1(x) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\mathcal{G}_2(x) = \begin{bmatrix} \sin(x_1 + 1) \\ 0 \end{bmatrix}$, $\mathcal{G}_3(x) = \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}$, $\mathcal{G}_4(x) = \begin{bmatrix} \cos(2x_2^2) \\ 0 \end{bmatrix}$, and $\mathcal{G}_5(x) = \begin{bmatrix} \sin(x_2 + 2) \\ 0 \end{bmatrix}$. Define $\mathcal{K}_{v_j}^2(x) \triangleq \rho \left\| \sum_{k=0}^{N} \mathcal{G}_k(x) v_k^* \right\|^2$ with $\rho = 2$. Considering the uncertainty on each player, we have

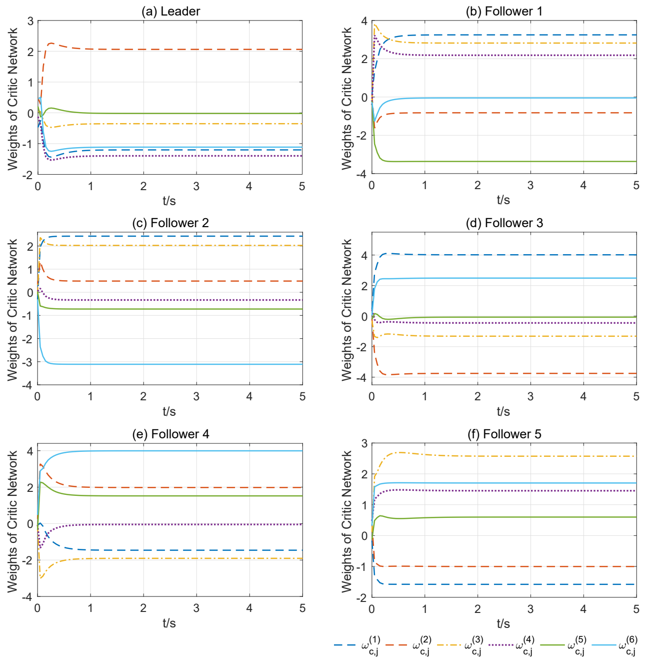

Fig. 1. Convergence process of critic network weights $\omega_{c,j}$, $j \in \{0,1,2,3,4,5\}$: (a) leader, (b) follower 1, (c) follower 2, (d) follower 3, (e) follower 4, and (f) follower 5.

$\|\varepsilon_j(x)\| \leq \|x\| \triangleq \varepsilon_{M,j}(x)$ and $\mathcal{A}^2_\varepsilon \triangleq \sum_{j=0}^{5} \varepsilon^2_{M,j}(x)$. Note that the open-loop configuration of (59) is unstable.

The critic network is designed for each player to learn its performance index $V_j(x)$ and help develop the feedback control policy u_j. Specifically, we establish the critic network for leader 0 at the high level based on (42) and for followers $\{1,2,3,4,5\}$ at the low level based on (43). The neuron structure of each critic network is designed as $4-6-1$, which represents four input neurons, six hidden neurons, and one output neuron. The input of the critic network for player j is $C_j = [x_1, x_2, u_j, v_j]^T$ and the output is the performance index $V_j(x)$. For this three-layer critic network (one hidden layer), we define the activation function as a sigmoid function which is described as

$$\phi_{c,j}(x) = \frac{1 - e^{-h_{c,j}}}{1 + e^{-h_{c,j}}} \tag{60}$$

with $h_{c,j} = \omega^T_{c1,j} C_j$, where $\omega_{c1,j}$ are the weights between the input and hidden layer of the critic network. In this paper, we randomly choose $\omega_{c1,j} \in [-0.5, 0.5]$ at the beginning and fix the values thereafter. Hence, we obtain

$$\begin{aligned}\nabla V_j(x) =& \nabla \phi^T_{c,j}(x) \omega_{c,j} \\ =& \frac{1}{2}\Big(\big(1 - \phi^2_{c,j}(x)\big)\omega_{c1,j}(x)\Big)^T \omega_{c,j}\end{aligned} \tag{61}$$

where $\omega_{c1,j}(x)$ are the fixed weights of x component for input to hidden layer of critic network and $\omega_{c,j}$ are the hidden-to-output layer weights updated based on (51). The initial value of $\omega_{c,j}$ is randomly chosen within $[-0.5, 0.5]$. Besides, choose $\alpha_{ik} = \beta_{ik} = 0.2$, $i \in \{1,2,3,4,5\}$, $k \in \{1,2\}$, and Q_j, R_j and

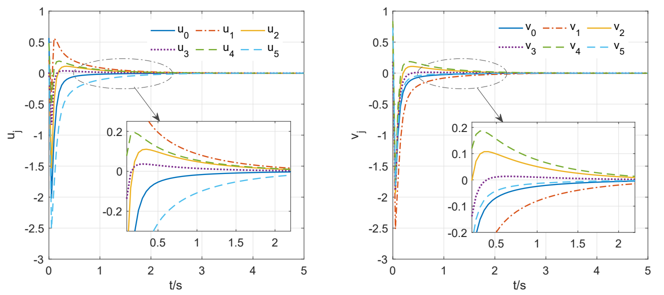

Fig. 2. Evolution of control policies u_j and v_j, $j \in \{0,1,2,3,4,5\}$ during the learning process.

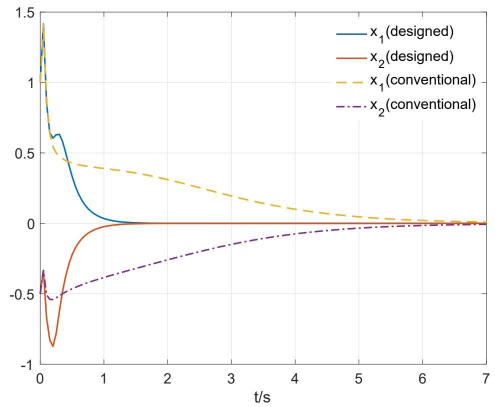

Fig. 3. Comparisons of state trajectories in the learning phase between the designed method and conventional flat actor-critic design [41].

Y_j, $j \in \{0,1,2,3,4,5\}$ are the identity matrix with suitable dimensions. According to Theorem 3, we substitute $\mathcal{A}^2_\varepsilon$, $\mathcal{K}^2_{v_j}$ and $\nabla V_j(x)$ into (51), and obtain the updating rule of the critic network weights for each player.

Select the learning rate as $\eta_j = 0.01$ and the sample interval as $0.05s$. We conduct the training based on the developed RL-based control algorithm with the initial state as $x(0) = [2, -2]$. The training has lasted 100 time steps × 0.05 sample interval $= 5s$. The convergence process of the critic network weights between the hidden and output layers $\omega_{c,j}$ are provided in Fig. 1 for all the players. We observe that the weights can converge quickly, which demonstrate the optimal learning process of the developed method. The evolution of the optimal control policies u_j and v_j during the training process are presented in Fig. 2. It is clearly shown that the designed policies can attain stable equilibrium at about $2s$, where the Stackelberg-Nash equilibrium is achieved. To show the effectiveness of the designed method, we compare our results with the conventional flat actor-critic design for uncertain multiplayer systems [41], where all the players are allowed to optimize their value functions equally. The comparisons under the same initial conditions are provided in Fig. 3. We observe that our developed two-level learning-based control method can converge much faster than the conventional method.

After the learning process, we apply the learned control

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3493833

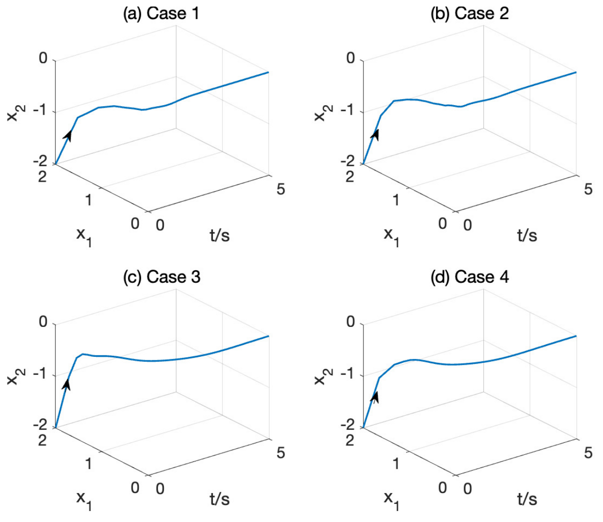

Fig. 4. State trajectories in the robust control phase: (a) Case 1, (b) Case 2, (c) Case 3, and (d) Case 4.

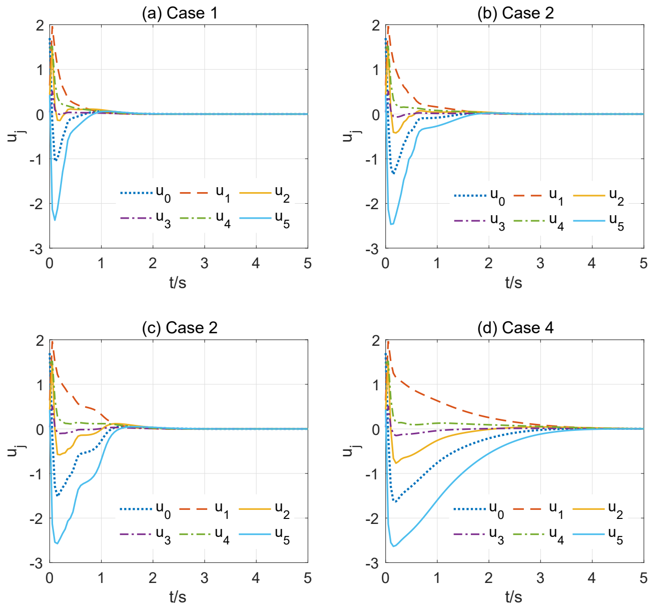

Fig. 5. Control policy trajectories in the robust control phase: (a) Case 1, (b) Case 2, (c) Case 3, and (d) Case 4.

policies u_j on the original uncertain hierarchical system (58) to verify the performance of the established robust-optimal transformation. We consider four cases of the uncertainties applied on the system with the parameters selected as follows:

- Case 1: $\lambda_1 = -1$, $\lambda_2 = -100$, $\lambda_3 = 1$, $\lambda_4 = 50$;
- Case 2: $\lambda_1 = -0.5$, $\lambda_2 = 100$, $\lambda_3 = -1$, $\lambda_4 = -100$;
- Case 3: $\lambda_1 = -0.2$, $\lambda_2 = -50$, $\lambda_3 = -1$, $\lambda_4 = 50$;
- Case 4: $\lambda_1 = 0.2$, $\lambda_2 = 1$, $\lambda_3 = -0.5$, $\lambda_4 = -1$.

With the learned policies, the state trajectories and control evolution during the robust control process for these four cases are given in Fig. 4 and Fig. 5, respectively. We observe that the system states of all the above four cases can quickly converge to the equilibrium point under the designed control policy u_j. In other words, the control policy u_j developed based on the transformed optimal stabilization is the solution of the original robust decision-making problem and can effectively stabilize the hierarchical multiplayer system with mismatched uncertainty.

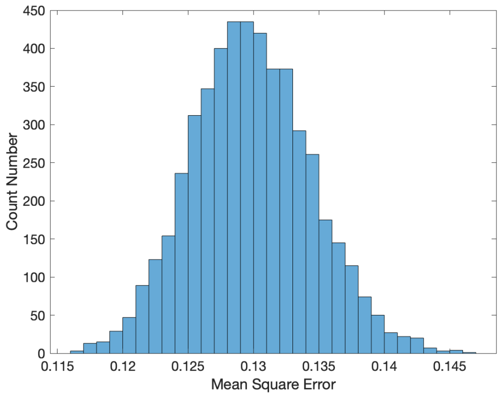

Fig. 6. Histogram of mean square error for x_1.

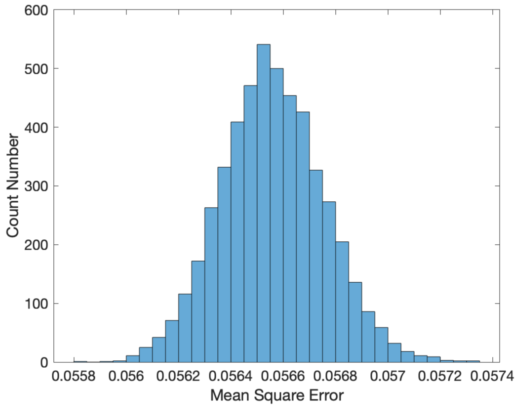

Fig. 7. Histogram of mean square error for x_2.

So far, we consider four specific cases to show the effectiveness of the designed learning-based control approach and hierarchical robust-optimal transformation. From the results, we can clearly observe that different set of uncertain parameters (λ_1, λ_2, λ_3 and λ_4) will drive the state in different trajectories. The convergence rates are also different. Therefore, without loss of generality, we randomly choose the admissible uncertainties and measure the mean square error (MSE) of the state during the robust control process for each set of parameters. We conduct $5,000$ independent runs and provide the histogram of MSE for x_1 and x_2 in Fig. 6 and Fig. 7, respectively. It is shown that the MSE of states are finite under all the admissible uncertainties with the designed control policies, which indicates that our developed intelligent hierarchical multiplayer system can converge to the stable equilibrium and is robust to any admissible uncertainties. These simulation results further demonstrate the equivalence of our established

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3493833

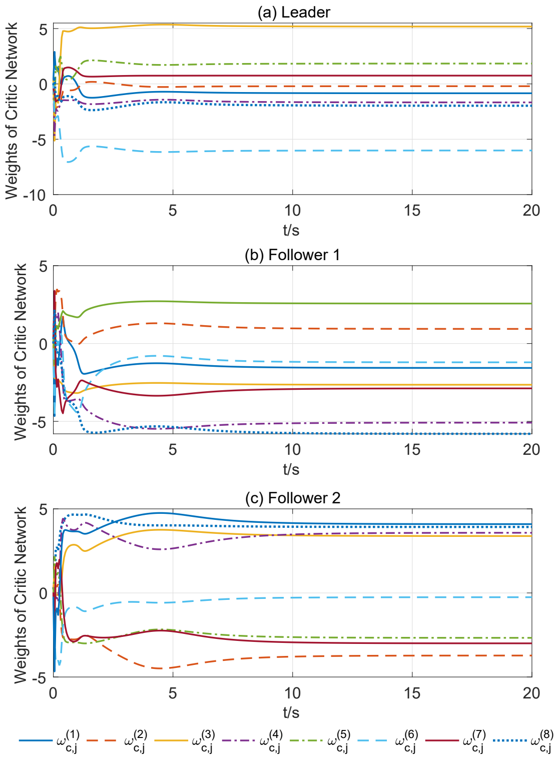

Fig. 8. Convergence process of critic network weights $\omega_{c,j}$, $j \in \{0,1,2\}$: (a) leader, (b) follower 1, (c) follower 2.

robust-optimal transformation, indicate the effectiveness of the developed learning-based control approach, and validate the theoretical studies.

Example 2: A three-player hierarchical system with four state variables $x = [x_1, x_2, x_3, x_4]^T \in \mathbb{R}^4$ has been presented to demonstrate the effectiveness of our proposed method. This problem is more challenging because it involves additional degrees of freedom, leading to more complex interactions and increased potential for uncertainties. Moreover, the system's open-loop configuration is unstable. The system function is given as

$$\dot{x} = f(x) + h_0(x)u_0 + \varepsilon_0(x) + \sum_{i=1}^{2} h_i(x)u_i + \sum_{i=1}^{2} \varepsilon_i(x) \tag{62}$$

where

$$f(x) = \begin{bmatrix} x_2 \\ -2x_2 + x_1x_2 + 0.5x_4(\cos(2x_2+2))^2 \\ x_4 \\ -x_2 + x_3 - 2x_4 + 0.5x_3x_4(\sin(2x_2^2+2))^2 \end{bmatrix}$$

and $h_0(x) = [0,0,0,2]^T$, $h_1(x) = [0,0,0,\cos(2x_1)]^T$, and $h_2(x) = [0,0,0,\cos(4x_2)+1]^T$. Three players $j \in \{0,1,2\}$ need to control this challenging system together with player 0 as the leader who acts first and players $\{1,2\}$ are the followers who respond to the leader's decision. Furthermore, the system involves unknown mismatched uncertainties $\varepsilon_j(x) = d_j(x)\xi_j(x)$, which makes the input and output data unreliable.

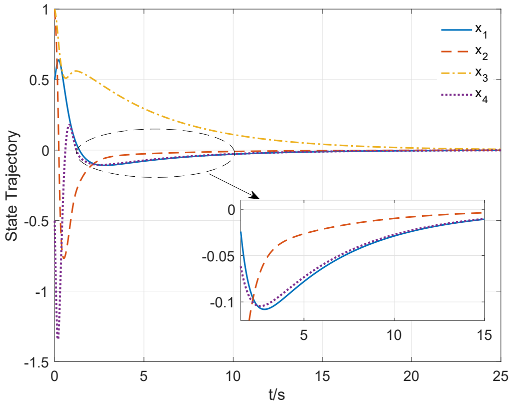

Fig. 9. State trajectories $x = [x_1, x_2, x_3, x_4]^T$ in the robust control phase.

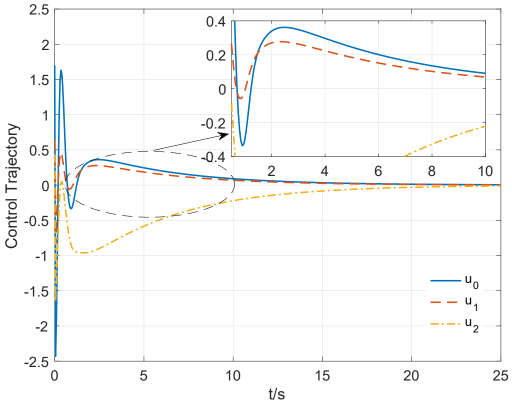

Fig. 10. Robust control trajectories for three players.

The dynamics of the uncertainties are given as

$$d_0(x) = \begin{bmatrix} \cos(x_1^2+1) \\ 0 \\ 0 \\ 1 \end{bmatrix}, d_1(x) = \begin{bmatrix} 0 \\ 2\sin(x_1+1) \\ 0.2 \\ 0 \end{bmatrix}, d_2(x) = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}$$

and

$$\xi_j(x) = p_1x_2\sin(x_1x_2) + p_2x_1\cos(x_3x_4)$$

with $p_1, p_2 \in [-1,1]$ are the unknown parameters. Hence, the upper bound for each uncertainty is defined as $\varepsilon_{M,j}(x) \triangleq 2\|x\|$. Then, we have $\mathcal{A}_\varepsilon^2 \triangleq \sum_{j=0}^{2} \varepsilon_{M,j}^2(x)$.

Apply the developed learning-based control method to solve this robust control problem. Based on Theorem 1-2, we build the corresponding nominal plant as

$$\dot{x} = f(x) + h_0(x)u_0 + \mathcal{G}_0(x)v_0 + \sum_{i=1}^{2} h_i(x)u_i + \sum_{i=1}^{2} \mathcal{G}_i(x)v_i \tag{63}$$

with $\mathcal{G}_0(x) = [\cos(x_1^2+1), 0, 0, 0]^T$, $\mathcal{G}_1(x) = [0, 2\sin(x_1+1), 0.2, 0]^T$, and $\mathcal{G}_2(x) = [0,2,0,0]$. The critic network is established for each player to stabilize the auxiliary system (63). Specifically, the network is built based on (42) for leader 0 at the high level and on (43) for followers $\{1,2\}$ at the low level. The neuron structure for each network is designed as $6-8-1$ with the input as $[x_1, x_2, x_3, x_4, u_j, v_j]$ and the output as $V_j(x)$. Set the learning rate of the critic network as $\eta_j = 0.01$ and select the sample interval as $0.05s$. The initial weights are randomly chosen within $[-0.5, 0.5]$. Then, we fix the input-to-hidden layer weights and update the hidden-to-output layer weights based on (51). The learning process spans a total of 400 time steps × 0.05 sample interval = $20s$ with the initial state as $x = [0.5, 1, 1, -0.5]^T$. The convergence evolution of the critic network weights between the hidden and the output layers is provided in Fig. 8. It shows that the learning phase can achieve equilibrium with the cooperation of three players. After that, we fix the critic network weights and design the feedback controller for the leader based on (45) and for the followers based on (48). The designed controllers are applied to the original uncertain system. Select the uncertain parameters as $p_1 = 1$ and $p_2 = -1$. We conduct the robust control phase for 500 time steps × 0.05 sample interval = $25s$, and present the state trajectories and control evolution in Fig. 9 and Fig. 10, respectively. We can observe that our designed robust-optimal transformation can quickly drive the system to the equilibrium point, and ensure the stability even in the presence of complex uncertainties. These results further validate the equivalence of our robust-optimal transformation mechanism and demonstrate the effectiveness of the developed learning-based control method for hierarchical multiplayer systems with mismatched uncertainties.

VI. Conclusion

In this paper, we design an intelligent hierarchical multiplayer system that is robust to the mismatched uncertainties. This new problem has been formulated as the multiplayer Stackelberg-Nash game integrated with a hierarchical robust-optimal transformation. A two-level neural-RL-based method is developed to stabilize the transformed nominal system, which is also the solution to the original uncertain multiplayer system in hierarchy. This method also ensures that the designed control policies of the players can achieve the Stackelberg-Nash equilibrium. The stability proof of the designed online learning process is also provided. Finally, the numerical studies verify the effectiveness of the developed hierarchical learning-based control approach.

Additionally, we also identify several directions for future research and extensions. For example, we intend to implement our designs in real-world problems, such as autonomous vehicle coordination or smart grid energy management. This will increase the practical applicability of our approach, provide valuable insights into its performance in dynamic environments, and align with societal values by contributing to safer, more efficient, and sustainable systems.

References

[1] D. Liu, H. Liu, J. Lü, and F. L. Lewis, "Time-varying formation of heterogeneous multiagent systems via reinforcement learning subject to switching topologies," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2023, early access.

[2] M. Zhou, J. Luo, J. Villella, Y. Yang, D. Rusu, J. Miao, W. Zhang, M. Alban, I. Fadakar, Z. Chen, *et al.*, "Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving," *arXiv preprint arXiv:2010.09776*, 2020.

[3] J. Xiao and M. Feroskhan, "Learning multi-pursuit evasion for safe targeted navigation of drones," *IEEE Transactions on Artificial Intelligence*, 2024, early access.

[4] A. Krnjaic, J. D. Thomas, G. Papoudakis, L. Schäfer, P. Börsting, and S. V. Albrecht, "Scalable multi-agent reinforcement learning for warehouse logistics with robotic and human co-workers," *arXiv preprint arXiv:2212.11498*, 2022.

[5] A. Shavandi and M. Khedmati, "A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets," *Expert Systems with Applications*, vol. 208, p. 118124, 2022.

[6] D. Xie and X. Zhong, "Semicentralized deep deterministic policy gradient in cooperative starcraft games," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 4, pp. 1584–1593, 2020.

[7] J. Perolat, B. De Vylder, D. Hennes, E. Tarassov, F. Strub, V. de Boer, P. Muller, J. T. Connor, N. Burch, T. Anthony, *et al.*, "Mastering the game of stratego with model-free multiagent reinforcement learning," *Science*, vol. 378, no. 6623, pp. 990–996, 2022.

[8] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, Cambridge, MA, 2018.

[9] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[10] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[11] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[12] J. D. Ellis, R. Iqbal, and K. Yoshimatsu, "Deep q-learning-based molecular graph generation for chemical structure prediction from infrared spectra," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 2, pp. 634–646, 2024.

[13] D. Wang and M. Hu, "Deep deterministic policy gradient with compatible critic network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 4332–4344, 2023.

[14] Y. Yang, Y. Pan, C.-Z. Xu, and D. C. Wunsch, "Hamiltonian-driven adaptive dynamic programming with efficient experience replay," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 3, pp. 3278–3290, 2024.

[15] Y. Yang, B. Kiumarsi, H. Modares, and C. Xu, "Model-free λ-policy iteration for discrete-time linear quadratic regulation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 2, pp. 635–649, 2023.

[16] L. Chen, C. Dong, and S.-L. Dai, "Adaptive optimal consensus control of multiagent systems with unknown dynamics and disturbances via reinforcement learning," *IEEE Transactions on Artificial Intelligence*, 2024, early access.

[17] H.-N. Wu and M. Wang, "Distributed adaptive inverse differential game approach to leader's behavior learning for multiple autonomous followers," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 6, pp. 1666–1678, 2023.

[18] K. G. Vamvoudakis, H. Modares, B. Kiumarsi, and F. L. Lewis, "Game theory-based control system algorithms with real-time reinforcement learning: How to solve multiplayer games online," *IEEE Control Systems Magazine*, vol. 37, no. 1, pp. 33–52, 2017.

[19] Y. Yang, H. Modares, K. G. Vamvoudakis, and F. L. Lewis, "Cooperative finitely excited learning for dynamical games," *IEEE Transactions on Cybernetics*, vol. 54, no. 2, pp. 797–810, 2023.

[20] C. Chen, F. L. Lewis, K. Xie, Y. Lyu, and S. Xie, "Distributed output data-driven optimal robust synchronization of heterogeneous multi-agent systems," *Automatica*, vol. 153, p. 111030, 2023.

[21] K. Shao, Y. Zhu, and D. Zhao, "Starcraft micromanagement with reinforcement learning and curriculum transfer learning," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 1, pp. 73–84, 2019.

[22] X. Zhong and H. He, "Grhdp solution for optimal consensus control of multiagent discrete-time systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 7, pp. 2362–2374, 2020.

[23] H. Zhang, H. Jiang, C. Luo, and G. Xiao, "Discrete-time nonzero-sum games for multiplayer using policy-iteration-based adaptive dynamic programming algorithms," *IEEE transactions on cybernetics*, vol. 47, no. 10, pp. 3331–3340, 2016.

[24] F. Li, J. Qin, and Y. Kang, "Closed-loop hierarchical operation for optimal unit commitment and dispatch in microgrids: A hybrid system approach," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 516–526, 2020.

[25] J. Xu and H. Zhang, "Sufficient and necessary open-loop stackelberg strategy for two-player game with time delay," *IEEE transactions on cybernetics*, vol. 46, no. 2, pp. 438–449, 2016.

[26] M. Latifi, A. Khalili, A. Rastegarnia, and S. Sanei, "Fully distributed demand response using the adaptive diffusion–stackelberg algorithm," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2291–2301, 2017.

[27] P. Paudyal, P. Munankarmi, Z. Ni, and T. M. Hansen, "A hierarchical control framework with a novel bidding scheme for residential community energy optimization," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 710–719, 2019.

[28] K. Ni, Z. Wei, H. Yan, K.-Y. Xu, L.-J. He, and S. Cheng, "Bi-level optimal scheduling of microgrid with integrated power station based on stackelberg game," in *2019 4th International Conference on Intelligent Green Building and Smart Grid (IGBSG)*, pp. 278–281, IEEE, 2019.

[29] M. Dai, T. H. Luan, Z. Su, N. Zhang, Q. Xu, and R. Li, "Joint channel allocation and data delivery for uav-assisted cooperative transportation communications in post-disaster networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16676–16689, 2022.

[30] Y. Huang and J. Zhao, "Active interdiction defence scheme against false data-injection attacks: A stackelberg game perspective," *IEEE Transactions on Cybernetics*, 2023, early access.

[31] J. Huang, S. Wang, and Z. Wu, "Robust stackelberg differential game with model uncertainty," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3363–3380, 2022.

[32] H. Mukaidani and H. Xu, "Incentive stackelberg games for stochastic linear systems with h_∞ constraint," *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1463–1474, 2018.

[33] M. Li, J. Qin, Q. Ma, W. X. Zheng, and Y. Kang, "Hierarchical optimal synchronization for linear systems via reinforcement learning: A stackelberg–nash game perspective," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 4, pp. 1600–1611, 2021.

[34] K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "Open-loop stackelberg learning solution for hierarchical control problems," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 2, pp. 285–299, 2019.

[35] C. Mu, K. Wang, Q. Zhang, and D. Zhao, "Hierarchical optimal control for input-affine nonlinear systems through the formulation of stackelberg game," *Information Sciences*, vol. 517, pp. 1–17, 2020.

[36] M. Li, J. Qin, N. M. Freris, and D. W. Ho, "Multiplayer stackelberg–nash game for nonlinear system via value iteration-based integral reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1429–1440, 2022.

[37] L. Yu, J. Lai, J. Xiong, and M. Xie, "Robust adp-based control for uncertain nonlinear stackelberg games," *Neurocomputing*, vol. 561, p. 126834, 2023.

[38] H. Kebriaei, A. Razminia, *et al.*, "Robust on-line adp-based solution of a class of hierarchical nonlinear differential game," *arXiv preprint arXiv:1907.11414*, 2019.

[39] M. Lin, B. Zhao, and D. Liu, "Event-triggered robust adaptive dynamic programming for multiplayer stackelberg–nash games of uncertain nonlinear systems," *IEEE Transactions on Cybernetics*, 2023, early access.

[40] H. Zhao, N. Zhao, G. Zong, X. Zhao, and N. Xu, "Sliding-mode surface-based approximate optimal control for nonlinear multiplayer stackelberg-nash games via adaptive dynamic programming," *Communications in Nonlinear Science and Numerical Simulation*, p. 107928, 2024.

[41] X. Zhong and Z. Ni, "Data-driven reinforcement learning design for multi-agent systems with unknown disturbances," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2018.

Xiangnan Zhong (Member, IEEE) is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, Florida Atlantic University (FAU), Boca Raton, FL, USA. Her research interests include computational intelligence, reinforcement learning, cyber-physical systems, networked control systems, neural networks, and optimal control.

Prof. Zhong received the National Science Foundation (NSF) Faculty Early Career Development (CAREER) Award in 2021 and the NSF CRII Award in 2019. She was a recipient of the International Neural Network Society (INNS) Aharon Katzir Young Investigator Award in 2021 and the INNS Doctoral Dissertation Award in 2019. She has been serving as an Associate Editor of *IEEE Transactions on Neural Networks and Learning Systems* since 2021 and *IEEE Internet of Things Journal* since 2023.

Zhen Ni (Senior Member, IEEE) is currently an Associate Professor with the Department of Electrical Engineering and Computer Science (EECS), Florida Atlantic University (FAU), Boca Raton, FL, USA. His research interests mainly include artificial intelligence, and computational methods, and reinforcement learning. Dr. Ni received the Senior Faculty Teaching Award from FAU College of Engineering and Computer Science in 2024 and the NSF CAREER Award in 2021. He has been an Associate Editor of *IEEE Internet of Things Journal* since 2021, *IEEE Transactions on Neural Networks and Learning Systems* since 2019, and *IEEE Computational Intelligence Magazine* since 2018.