Hierarchical Safe Reinforcement Learning Control for Leader-Follower Systems with Prescribed Performance

Junkai Tan, Shuangsi Xue, *Member, IEEE,* Huan Li, Zihang Guo, Hui Cao, *Member, IEEE,* and Badong Chen, *Senior Member, IEEE,*

***Abstract*—This paper proposes a hierarchical safe reinforcement learning with prescribed performance control (HSRL-PPC) scheme to address the challenges of interconnected leader-follower systems operating in complex environments. The framework consists of two levels: at the higher level, the leader agent detects and avoids moving obstacles while planning optimal paths; at the lower level, the follower agent tracks the leader within strict prescribed performance bounds. We formulate the optimal prescribed performance safe control problem and solve it using the Hamilton-Jacobi-Bellman (HJB) equation. Due to system nonlinearity and obstacle complexity, we approximate the leader's optimal value function using a state-following neural network that efficiently extrapolates training data to neighboring states, while employing a regular critic neural network for the follower's value function approximation. Lyapunov stability analysis demonstrates the closed-loop system's theoretical guarantees. Experimental results from two simulation examples and hardware tests with a quadcopter-vehicle system validate the effectiveness of the proposed approach in achieving safe navigation and precise tracking performance in dynamic environments.**

***Note to Practitioners*—Challenges exist in unpredictable obstacles and agent limitations for the interconnected leader-follower system. To provide a safe, efficient, and reliable control scheme, hierarchical safe reinforcement learning with prescribed performance control is proposed in this paper. The hierarchical structure is utilized to coordinate the leader and follower agents in the interconnected system, where the leader agent plans the optimal path and avoids obstacles, and the follower agent tracks the leader within prescribed performance bounds. Based on the proposed hierarchical structure, engineers can design efficient and safe control schemes for interconnected leader-follower systems with moving obstacles. In future work, we will address the problem of external disturbances and uncertainties in the interconnected leader-follower system.**

***Index Terms*—Hierarchical structure, prescribed performance, safe reinforcement learning, interconnected leader-follower, obstacle avoidance**

This research is supported by National Natural Science Foundation of China under grant U21A20485 and 62311540022, and China Postdoctoral Science Foundation under grant 2024M762602 (*Corresponding author: Shuangsi Xue*)

Junkai Tan, Shuangsi Xue, Huan Li, Zihang Guo and Hui Cao are with the Shaanxi Key Laboratory of Smart Grid, Xi'an Jiaotong University, Xi'an 710049, China, and also with the State Key Laboratory of Electrical Insulation and Power Equipment, School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China. (e-mail: tanjk@stu.xjtu.edu.cn; xssxjtu@xjtu.edu.cn; lh2000dami@stu.xjtu.edu.cn; guozihang@stu.xjtu.edu.cn; huicao@mail.xjtu.edu.cn)

Badong Chen is with the National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, and the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China. (e-mail: chenbd@mail.xjtu.edu.cn)

I. Introduction

INTERCONNECTED systems consisting of multiple agents with different capabilities are of significant importance as they can efficiently handle complex tasks in various application domains [1]–[3]. Many forms of structure are available for such systems, such as hierarchical [4], decentralized [5], or centralized architectures [6] et al. One typical and widely adopted form is the leader-follower structure, finding applications in scenarios like quadcopter tracking and landing on moving vehicles [7], rocket docking with space stations [8], and many others. Despite the advantages offered by the leader-follower structure, challenges remain to be addressed for its deployment, particularly in dynamic environments with unpredictable obstacles and varying limitations [9]. Therefore, developing efficient and secure control schemes for interconnected leader-follower systems operating in complex environments is critical for risk reduction and reliable operation.

A key factor affecting the control of interconnected systems is the utilization efficiency of system information. In order to enhance the efficiency of controllers in utilizing system information, various types of structures have been widely investigated [10]–[12]. The hierarchical control structure is one of the emerging effective methods for the coordination of agents within the interconnected system [13]. For human-robot interconnected system, game-based hierarchical control structure have been proposed, where the leader interacts with human operators to seek Nash equilibrium [14], [15]. To control interconnected quadcopter-manipulator system, hierarchical controllers are proposed in which quadcopter plans optimal path safely while manipulator executes the precise tasks [16], [17]. More applications of hierarchy structure have been investigated in lower limb exoskeleton systems, where the planner learns from human motion and the executor tracks the replanned trajectory [18], [19]. For the safe and efficient control of mobile robots in unknown environments, deep reinforcement learning is utilized to plan and track optimal motion hierarchically [20], [21]. Overall, hierarchical control structure could provide an effective method for communication and cooperation among interconnected leader-follower systems.

In practical applications, interconnected systems usually have to deal with unpredictable obstacles in complex environments. Ensuring the safety of agents is another significant challenge in controller design. To avoid collisions with obstacles, safe control methods including control barrier functions [22],

Barrier Lyapunov functions [23], and model predictive control [24] have been widely studied. However, the explicit model of system dynamics for these methods is often difficult to obtain in practice. To address this issue, reinforcement learning (RL) is proposed to provide a learning approach to approximate the optimal control strategy by interacting with the environment [25], [26]. With the integration of safe methods with RL, safe reinforcement learning (SRL) has been proposed to learn the safe control policy in the presence of unsafe operation regions [27], [28]. SRL-based approximate optimal planning method in presence of moving obstacles is investigated in [29], where the quadcopter plans the optimal path and avoids moving environmental obstacles. In [30], barrier function-based state transformation is utilized to design safe control policies. Integrating the control barrier function with reinforcement learning [31], [32], SRL has been investigated to learn the safe control policy with unsafe no-entry regions. However, the existing SRL methods are limited to single-agent or simple systems, and the interconnected leader-follower system is not considered, due to the challenge of dealing with complex interconnected leader-follower systems with moving obstacles.

The follower agents from the interconnected system always act as lower-level executors, which track the leader's planned motion. However, due to limited sensor capabilities or processing speed [33], [34], the followers may not have the capability to detect and avoid environmental obstacles, which increases the risk of collisions and accidents [35]. To address this issue, specific constraints are required to be implemented to ensure the safety of the follower agent. In [36], [37], RL-based finite time optimal controller is investigated for the complex interconnected system, which ensures the terminal constraints within finite time bounds. To ensure tracking errors satisfy specified constraints, prescribed performance control (PPC) method have been widely studied [38]–[40]. In [39], a data-driven PPC scheme is proposed for the optimal tracking control of unmanned surface vehicles, where the tracking performance is guaranteed by the prescribed constraints. Literature [41] investigates the prescribed performance event-triggered control for the interconnected multi-input system, where the tracking error is constrained within the prescribed performance bounds. To deal with the spacecraft attitude tracking control problem, RL-based PPC for saturated actuator spacecraft is proposed in [42], [43], where the attitude tracking error of spacecraft is constrained within the prescribed performance bounds.

Motivated by the above challenges posed by navigating complex environments for the interconnected leader-follower system, a hierarchical safe reinforcement learning with prescribed performance control (HSRL-PPC) for leader-follower systems is proposed. Contributions are summarized as follows:

1) The HSRL-PPC scheme is proposed for leader-follower systems operating in complex environments with moving obstacles. In this hierarchical structure, the leader agent detects environmental obstacles and plans optimal avoidance paths, while the follower agent tracks the leader within strict performance bounds. This approach offers advantages over existing hierarchical schemes [14], [16], [18], [20] by integrating obstacle avoidance with prescribed performance tracking in a unified framework.
2) For leader safe optimal path planning, a StaF NN-based SRL approach is developed to efficiently approximate the leader's optimal value function. This technique enables faster convergence with reduced computational requirements in complex environments compared to traditional RL methods [31], [39]. For follower tracking control, we implement PPC that ensures position maintenance within defined constraints during obstacle avoidance maneuvers, providing tighter tracking guarantees than conventional methods [9], [44]. These guarantees are essential for safety-critical applications like landing or docking operations requiring high precision.
3) Effectiveness of proposed HSRL-PPC scheme is validated by two simulations and two hardware experiments featuring a quadcopter-vehicle system. The hardware implementation demonstrates real-world applicability, showing the leader vehicle successfully plans optimal paths around obstacles while the follower quadcopter maintains tracking within prescribed performance.

The paper is organized as follows: Section II introduces interconnected system. Section III formulates optimal safe control problem. Section IV presents the HSRL-PPC scheme. Sections V and VI provide simulation and hardware experiments to verify effectiveness. Section VII concludes the paper.
Notation: $\mathbb{R}^n$ is the n-dimensional Euclidean space; $\mathbb{R}^{m\times n}$ is the set of $m \times n$ real matrices; $\|x\|$ is the Euclidean norm of vector x; I_n is the identity matrix of size n.

II. Preliminaries

A. System description

The interconnected leader-follower system is composed of upper-level leader agents and lower-level follower agents. The leader agents work as motion planners to detect the obstacles in the environment and plan the trajectories, and the follower agents function as bottom executors of detailed duties, which cannot detect obstacles due to the lack of environmental sensors. Consider single leader and single follower with the following nonlinear affine input dynamics:

$$\begin{cases} \dot{x}_l(t) = f_l(x_l(t)) + g_l(x_l(t))u_l(t) \\ \dot{x}_f(t) = f_f(x_f(t)) + g_f(x_f(t))u_f(t) \end{cases} \tag{1}$$

where $x_l \in \mathbb{R}^{n_l}$, $x_f \in \mathbb{R}^{n_f}$ denote the states of the leader and the follower, respectively, $u_l \in \mathbb{R}^{m_l}$, $u_f \in \mathbb{R}^{m_f}$ denote the control inputs, $f_l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$, $f_f : \mathbb{R}^{n_f} \to \mathbb{R}^{n_f}$ denote the drift dynamics, and $g_l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l\times m_l}$, $g_f : \mathbb{R}^{n_f} \to \mathbb{R}^{n_f\times m_f}$ denote the control effectiveness matrices. The leader and the follower are connected by a communication channel, which transmits the real-time state information of the leader to the follower. Assume that the communication channel is ideal, i.e., the leader's state information can be transmitted to the follower without any delay or loss. For the follower, to track the leader agent, the leader's state information is assumed to be

the desired state. The tracking error is defined as $e = x_f - x_l$. The tracking error dynamics is given by:

$$\begin{aligned}\dot{e}(t) =& \dot{x}_f(t) - \dot{x}_l(t) \\ =& f_f(x_f) - f_l(x_l) + g_f(x_f)u_f - g_l(x_l)u_l \end{aligned} \tag{2}$$

Note that the leader's objective is to stabilize system (1) while avoiding obstacles, while the follower's objective is to track the leader by dynamics (2) within prescribed performance bounds. In the environment, moving obstacles can be detected by the upper-level leader agent, but remain undetectable to the lower-level follower agent. With M avoidance obstacles in the environment and $\mathcal{M} = \{1, \cdots, M\}$ denoting this set, the moving obstacles follow nonlinear dynamics:

$$\dot{x}_{o,i}(t) = g_{o,i}(x_{o,i}(t)), \quad i \in \mathcal{M} \tag{3}$$

where $x_{o,i} \in \mathbb{R}^{n_{o,i}}$ denotes the center states (or position in general) of the i-th obstacle, $g_{o,i} : \mathbb{R}^{n_{o,i}} \to \mathbb{R}^{n_{o,i}}$ denotes the smooth drift dynamics of the i-th obstacle, which is equivalent to zero when the obstacle is static.

B. Obstacle modeling

In this subsection, the modeling of moving obstacles and the definition of barrier function are introduced. To differentiate the emergency of the operation space near a moving obstacle, the nearby space of the moving obstacle is modeled as a layered sphere, which is composed of three different regions with different levels of danger: detection region, warning region, and obstacle region.

Assumption 1 (Obstacle modeling)**.** *For obstacle modeling, the following assumptions are made:*

1) *Obstacles are non-overlapping moving entities; any overlapping obstacles are modeled as a larger obstacle.*
2) *Each obstacle is represented by a minimum enclosing sphere, with the sphere radius defining the avoidance boundary condition.*
3) *The i-th obstacle O_i is characterized by center point $p_{o,i}$ and radius $r_{o,i}$, with M total obstacles in environment.*

Define real-time distance between the i-th avoidance obstacle and the leader agent as $d_i(x_l, x_{o,i}, t) = \|x_l(t) - x_{o,i}(t)\|$. To facilitate the avoidance of the obstacles, we define the avoidance region as the set of the states where the distance between the states and the i-th avoidance obstacle satisfies $d_i(x_l, x_{o,i}, t) \leq r_i$. The avoidance regions are composed of three regions with different levels of danger:

1) Detection region $\mathcal{D} = \cup_{i\in\mathcal{M}}\mathcal{D}_i$: the set of regions where the leader can detect the i-th moving obstacle:

$$\mathcal{D}_i = \{x_l \in \mathbb{R}^{n_l} | R_{o,i} < d_i(x_l, x_{o,i}, t) \leq D_{o,i}\}$$

2) Warning region $\mathcal{W} = \cup_{i\in\mathcal{M}}\mathcal{W}_i$: the set of regions where the leader agent is required to take actions to avoid the further approaching, which is defined as:

$$\mathcal{W}_i = \{x_l \in \mathbb{R}^{n_l} | r_{o,i} < d_i(x_l, x_{o,i}, t) \leq R_{o,i}\}$$

3) Obstacle region $\mathcal{O} = \cup_{i\in\mathcal{M}}\mathcal{O}_i$: the set of regions where the leader agent is required to take immediate actions to avoid the collision from obstacle, which is defined as:

$$\mathcal{O}_i = \{x_l \in \mathbb{R}^{n_l} | d_i(x_l, x_{o,i}, t) \leq r_{o,i}\}$$

where $r_{o,i}$, $R_{o,i}$, $D_{o,i}$ are the radius shown in Fig. 1 for the i-th obstacle, which satisfy $r_{o,i} < R_{o,i} < D_{o,i}$. The avoidance region is defined as the union set of all the defined obstacles, which is denoted by $\mathcal{A} = \cup_{i\in\mathcal{M}}(\mathcal{D}_i \cup \mathcal{W}_i \cup \mathcal{O}_i)$.

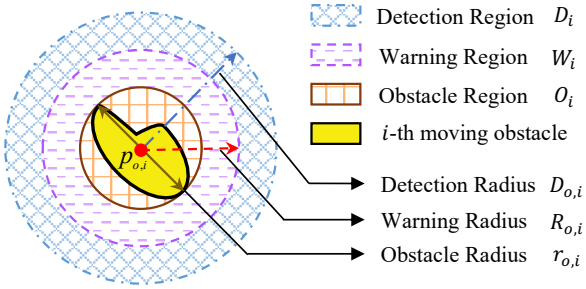


Fig. 1. The obstacle avoidance region.

C. Safe operational region modeling

To guarantee the safety of system (1) while existing moving obstacles, it is necessary to establish a safe operational region to avoid moving obstacles. Therefore, the definition of the forward invariance of the operational region is given:

Definition 1. *(Forward invariance of operational region [32]). Consider a subset region $\mathcal{C} \subset \mathbb{R}^{n_l}$ of leader agent's operational space. $\mathcal{C}$ is forward invariant if for any initial state $x_l(0) \in \mathcal{C}$, trajectory $x_l(t)$ remains within $\mathcal{C}$ for all $t \geq 0$. Region $\mathcal{C}$ is called safe operational region. Define $\mathcal{C}$ as:*

$$\begin{aligned}\mathcal{C} &= \{x_l \in \mathbb{R}^{n_l} | h(x_l) \geq 0\} \\ \partial\mathcal{C} &= \{x_l \in \mathbb{R}^{n_l} | h(x_l) = 0\} \\ Int(\mathcal{C}) &= \{x_l \in \mathbb{R}^{n_l} | h(x_l) > 0\}\end{aligned}$$

where $h(x_l)$ is a continuously differentiable function, with $\partial\mathcal{C}$ and $Int(\mathcal{C})$ denoting the boundary and interior of $\mathcal{C}$.

To keep $\mathcal{C}$ forward invariant while exploring the leader agent's operational space, it is essential to ensure $h(x_l(t)) \geq 0, \forall t \geq 0$. Therefore, the control barrier function is introduced to design safe control policies, which keeps the leader remaining in the safe operational region.

Definition 2. *(Control barrier function [22]). For dynamic (1), there is a differentiable smooth function $B(x_l)$, which is said to be a control barrier function (CBF) if satisfying:*

$$\begin{aligned}&\frac{1}{\alpha_1 h(x_l)} \leq B(x_l) \leq \frac{1}{\alpha_2 h(x_l)} \\ &s.t. \inf_{x_l \in Int(\mathcal{C})} B(x_l) \geq 0, \lim_{x_l \to \partial\mathcal{C}} B(x_l) = \infty\end{aligned}$$

where α_1, α_2 are positive constants.

Consider the control barrier function in the following form:

$$B(x_l) = \frac{K_B s(x_l)}{h(x_l) + \mu} \tag{4}$$

where $h(x_l) = \sum_{i\in\mathcal{M}} h_i(x_l)$ and $s(x_l) = \sum_{i\in\mathcal{M}} s_i(x_l)$ are continuously differentiable smooth functions, K_B, μ are positive constants, and $s_i(x_l)$, $h_i(x_l)$ are continuously differentiable smooth functions for each $i \in \mathcal{M}$, which satisifies:

$$h_i(x_l) = d_i(x_l, x_{o,i}, t) - r_{o,i}$$
$$s_i(x_l) = \begin{cases} 0, & d_i > r_{d,i}, \\ l_1 + l_1\cos(\pi\frac{d_i^2 - D_{o,i}^2}{D_{o,i}^2 - R_{o,i}^2}), & R_{o,i} < d_i \le D_{o,i}, \\ l_2 + l_3\cos(\pi\frac{d_i^2 - r_{o,i}^2}{R_{o,i}^2 - r_{o,i}^2}), & r_{o,i} < d_i \le R_{o,i}, \\ 1, & d_i \le r_{o,i}, \end{cases} \tag{5}$$

where d_i is the simplified notation of $d_i(x_l, x_{o,i}, t)$, l_1, l_2, and l_3 are the parameters to adjust the smoothness of the barrier function, which satisfy $l_2 + l_3 = 1$, $l_2 - l_3 = 2l_1$.

Remark 1 (Multiple Obstacle Handling)**.** *For multiple obstacles, we treat obstacles as non-overlapping entities, with overlapping ones modeled as a single larger obstacle (Assumption 1). Control barrier function $\mathcal{B}$ evaluates the combined effect of all detected obstacles, while the smooth detection function $s_i(x_l, x_{o,i})$ dynamically prioritizes based on proximity. This design prevents the limitation where $\lim_{x_l\to\partial C} B(x_l) \neq \infty$ when evaluating obstacles independently.*

III. Problem Formulation: The Optimal Prescribed Performance Safe Control

To ensure system (1) safety, the leader must detect moving obstacles and plan optimal avoidance paths, while the follower track the leader within specific constraints to ensure precise landing or docking operations. Considering both safety and tracking performance requirements, this section formulates the optimal prescribed performance safe control problem.

A. Leader-follower prescribed performance control

In this subsection, PPC is introduced for follower tracking leader within constraints. First, definition of performance bounds is given:

Definition 3. *(Performance Bounds [38]). A smooth decreasing function $\rho_i(t)$ is called a performance bound if it satisfies:*

$$\lim_{t\to\infty}\rho_i(t) = \rho_{i\infty}, \quad \lim_{t\to 0}\rho_i(t) = \rho_{i0},$$

where $\rho_{i0} > \rho_{i\infty} > 0, i = 1, \cdots, n_f$ are parameters for the i-th performance bound.

To guarantee that the tracking error between the leader and the follower satisfies a specific constraint $\mathcal{E}_{l,i} \le e_i \le \mathcal{E}_{u,i}$, the performance bounds are designed as:

$$\rho_i = (\rho_{i0} - \rho_{i\infty})e^{-\lambda_i t} + \rho_{i\infty}, \quad i = 1, \cdots, n_f \tag{6}$$

where λ_i is the decay rate of the i-th performance bound. Then, the lower and upper performance bounds of the i-th tracking error could be designed as $\mathcal{E}_{l,i} = -\zeta\rho_i$ and $\mathcal{E}_{u,i} = \zeta\rho_i$, where $\zeta \in (0, 1)$ is a user-specified parameter to adjust.

Based on the design of performance bounds, the tracking error is transformed using the following transformation:

$$\epsilon_i = \tan\left(\frac{\pi}{2} \times \frac{2e_i - \mathcal{E}_{l,i} - \mathcal{E}_{u,i}}{\mathcal{E}_{u,i} - \mathcal{E}_{l,i}}\right) \tag{7}$$

$$e_i = \frac{\mathcal{E}_{u,i} - \mathcal{E}_{l,i}}{\pi}\arctan(\epsilon_i) + \frac{\mathcal{E}_{l,i} + \mathcal{E}_{u,i}}{2} \tag{8}$$

where $\epsilon_i, i = 1, \cdots, n_f$ is the i-th transformed tracking error. Then the dynamic of the i-th tracking error is transformed as:

$$\dot{\epsilon}_i = \frac{\partial\epsilon_i}{\partial e_i}\dot{e}_i + \frac{\partial\epsilon_i}{\partial\mathcal{E}_{l,i}}\dot{\mathcal{E}}_{l,i} + \frac{\partial\epsilon_i}{\partial\mathcal{E}_{u,i}}\dot{\mathcal{E}}_{u,i} = \frac{\partial\epsilon_i}{\partial e_i}\dot{e}_i + \Phi_i \tag{9}$$

where $\frac{\partial\epsilon_i}{\partial e_i} = \pi\sec^2\left(\frac{\pi}{2} \times \frac{2e_i - \mathcal{E}_{l,i} - \mathcal{E}_{u,i}}{\mathcal{E}_{u,i} - \mathcal{E}_{l,i}}\right)/(2\mathcal{E}_{u,i} - 2\mathcal{E}_{l,i})$, $\Phi_i = \frac{\partial\epsilon_i}{\partial\mathcal{E}_{l,i}}\dot{\mathcal{E}}_{l,i} + \frac{\partial\epsilon_i}{\partial\mathcal{E}_{u,i}}\dot{\mathcal{E}}_{u,i}, i = 1, 2, \cdots, n_f$. Subsequently, the overall tracking error dynamic is transformed as:

$$\dot{\epsilon} = \mathcal{H}\dot{e} + \Phi \tag{10}$$

where $\mathcal{H} = \text{diag}(\left[\partial\epsilon_1/\partial e_1, ..., \partial\epsilon_{n_f}/\partial e_{n_f}\right]) \in \mathcal{R}^{n_f\times n_f}$, $\Phi = [\Phi_1, ..., \Phi_{n_f}]^\top \in \mathcal{R}^{n_f}$.

Remark 2 (PPC vs. Robust and FT Control)**.** *In real-world applications like quadcopter landing or spacecraft docking operations, unpredictable disturbances can cause tracking errors to exceed safety bounds. Compared to robust control [16] and finite-time control approaches [9], [37], PPC provides tighter, more accurate tracking guarantees by enforcing prescribed performance bounds throughout the entire operation.*

Remark 3 (Ensuring Initial PPC Boundary Conditions)**.** *Ensuring initial tracking error satisfies $e(0) \in \Omega_e$ is critical for PPC implementation. We employ three practical strategies: (1) a brief initialization phase with a preliminary controller to establish boundary conditions; (2) adaptive performance bounds $\mathcal{E}_{l,i}$ and $\mathcal{E}_{u,i}$ with adjustable parameter $\zeta \in (0, 1)$; and (3) a pre-learning phase for neural network training before full controller activation. These approaches were successfully validated in our quadcopter experiments, demonstrating reliability even under challenging initial conditions.*

Accordingly, the dynamics of the interconnected leader-follower system, the dynamics of the tracking error and the dynamics of the moving obstacles are augmented as:

$$\dot{X}(t) = F(X(t)) + G(X(t))U(t) \tag{11}$$

where $X = [x_l^\top, \epsilon^\top, x_{o,1}^\top, ..., x_{o,M}^\top]^\top \in \mathbb{R}^{n_l+n_f+M\times n_o}$ is the augmented state, $U = [u_l^\top, u_f^\top, 0_{1\times M}]^\top \in \mathbb{R}^{m_l+m_f+M}$ is the augmented control input, and

$$F(X(t)) = \begin{bmatrix} f_l(x_l(t)) \\ H\left(f_f(x_f(t)) - f_l(x_l(t))\right) + \Phi \\ f_{o,1}(x_{o,1}(t)) \\ \vdots \\ f_{o,M}(x_{o,M}(t)) \end{bmatrix} \tag{12}$$

$$G(X(t)) = \begin{bmatrix} g_l(x_l(t)) & 0_{n_l\times n_f} & 0_{n_l\times M} \\ -\mathcal{H}g_l(x_l(t)) & \mathcal{H}g_f(x_f(t)) & 0_{n_f\times M} \\ 0_{M\times n_l} & 0_{M\times n_f} & 0_{M\times M} \end{bmatrix} \tag{13}$$

Through tracking error transformation, the problem of optimal prescribed performance safe control is formulated. For system

(1), our goal is to design optimal control input $U(t)$ with two components: one for the leader to plan optimal obstacle-avoiding paths, and another for the follower to track the leader within specific performance constraints.

B. Problem formulation

Based on the augmented dynamics (11) of the interconnected leader-follower system, a quadratic cost function can be defined for the optimal prescribed performance safe control:

$$J(X,U)=\int_{t_o}^{\infty}\left(r(X(\tau),U(\tau))\right)d\tau \tag{14}$$

where the saturated control input $U(t)$ satisifies $\mu_{\min,i}\leq U_i(t)\leq\mu_{\max,i}$, $\mu_{\max,i}=-\mu_{\min,i}=\mu_i$ are the symmetric saturation constraints of the control input. The instantaneous reward function $r(X(\tau),U(\tau))$ is given as:

$$\begin{aligned}r(X,U)=x_l^\top Q_{x_l}x_l+&\epsilon^\top Q_\epsilon\epsilon+\Phi(U)+\mathcal{B}(x_l,x_o)\\&+\sum_{i=1}^{M}s_i(x_l,x_{o,i})x_{o,i}^\top Q_{x_{o,i}}x_{o,i}\end{aligned} \tag{15}$$

where $Q_{x_l}\in\mathbb{R}^{n_l\times n_l}$, $Q_\epsilon\in\mathbb{R}^{n_f\times n_f}$, $Q_{x_{o,i}}\in\mathbb{R}^{n_{o,i}\times n_{o,i}}$ are positive definite state penalty matrices for the leader, the transformed tracking error of the follower, and the state of moving obstacles, respectively. In the penalty term of moving obstacle state, the smooth function $s_i(x_l,x_{o,i})$ from (5) is utilized to facilitate the avoidance of the obstacle when the obstacle is detected. $\mathcal{B}(x_l,x_o)$ is the control barrier function defined in (4). $\Phi(U)$ is the penalty of the control input [34]:

$$\Phi(U)=2R\int_0^U\left(\mu\tanh^{-1}\left(\zeta_U/\mu\right)\right)\,\mathrm{d}\zeta_U. \tag{16}$$

where $R\in\mathbb{R}^{(m_l+m_f)\times(m_l+m_f)}$ is a positive definite control input penalty matrix. $\mu=[\mu_1,\mu_2,\cdots,\mu_{m_l+m_f}]^\top$ is the saturation value of the control input, ζ_U is an integral variable. To construct the optimal prescribed performance safe controller, it is desired to obtain the optimal value function:

$$V^*(X)=\min_{U(\tau)\in\Omega_U}\int_t^{\infty}\left(r(X(\tau),U(\tau))\right)d\tau \tag{17}$$

where $\Omega_U\in\mathbb{R}^{(m_l+m_f)\times1}$ is the admissible set of the control input. To obtain the optimal value function (17), the Hamilton function is introduced:

$$H(X,U,\nabla V^*)=r(X,U)+(\nabla V^*)^\top(F(X)+G(X)U) \tag{18}$$

Following the extreme condition of the value function (17) and the Hamilton function (18), the optimal control input could be given as:

$$U^*(X)=-\mu\tanh\left(R^{-1}G^\top\left(\nabla V^*(X)\right)^\top/(2\mu)\right) \tag{19}$$

Combining the optimal control input (19) with the Hamilton function (18), the HJB equation is obtained as:

$$0=r(X,U^*)+(\nabla V^*)^\top(F(X)+G(X)U^*) \tag{20}$$

The saturated optimal control input (19) and the corresponding optimal value (17) could be derived by solving the HJB equation (20). The problem of optimal prescribed performance safe control is formulated. The next section will introduce a hierarchical reinforcement learning control scheme to solve the optimal prescribed performance safe control problem.

Remark 4 (Obstacle State Penalty)**.** *The obstacle state penalty term $\sum_{i=1}^{M}s_i(x_l,x_{o,i})x_{o,i}^\top Q_{x_{o,i}}x_{o,i}$ in the reward function provides several benefits: (1) it enables the leader to anticipate obstacle movements for proactive planning; (2) it encourages maintaining safe distances even when obstacles aren't immediate threats; and (3) by using the detection function $s_i(x_l,x_{o,i})$ as a weight, the controller prioritizes relevant obstacles based on proximity. This creates a graduated response from distant awareness to active avoidance, producing more efficient navigation compared to binary safety constraints alone.*

IV. Hierarchical Safe Reinforcement Learning with Prescribed Performance Control

The optimal prescribed performance safe controller (19) is designed by solving the HJB equation (20), which yields the optimal value function (17). However, it is difficult to obtain the optimal value function in practice, due to the nonlinearity of interconnected system dynamics and moving obstacles. To address this issue, the HSRL-PPC scheme is designed.

A. Design of hierarchical reinforcement learning

In this subsection, optimal value function (17) is separated into two parts: the optimal value function of the leader agent V_l^*, and the optimal value function of the follower agent V_f^*, which meet the following conditions:

$$V_l^*=\min_{u_l(\tau)\in\Omega_{u_l}}\int_t^{\infty}\left(r_l(X(\tau),u_l(\tau))\right)d\tau \tag{21}$$

$$V_f^*=\min_{u_f(\tau)\in\Omega_{u_f}}\int_t^{\infty}\left(r_f(X(\tau),u_f(\tau))\right)d\tau \tag{22}$$

where $r_l(X,u_l)=x_l^\top Q_{x_l}x_l+\Phi_l(u_l)+\mathcal{B}(x_l,x_o)+\sum_{i=1}^{M}s_i(x_l,x_{o,i})x_{o,i}^\top Q_{x_{o,i}}x_{o,i}$ is the reward function of the leader agent, $r_f(X,u_f)=\epsilon^\top Q_\epsilon\epsilon+\Phi_f(u_f)$ is the reward function of the follower agent, $\Phi_l(u_l)=\Phi([u_l^\top,0_{(m_f+M)\times1}]^\top)$ is the penalty term of the leader agent's control input, and $\Phi_f(u_f)=\Phi([0_{m_l\times1},u_f^\top,0_{M\times1}]^\top)$ is the penalty term of the follower agent's control input. To approximate the optimal value functions for both agents, we utilize distinct neural network approaches. For the leader agent, traditional training is challenging due to sparse safe navigation data and obstacle complexity. We implement a state-following neural network (StaF-NN) [28], [29] that efficiently extrapolates training data to neighboring states, accelerating learning in complex environments. Using this approach, the leader's optimal value function and control input are formulated as:

$$V_l^*(x_l)=W_l^\top\varphi_l(x_l,c(x_l))+\mathcal{B}(x_l,x_o)+\varepsilon_l(x_l) \tag{23}$$

$$\begin{aligned}u_l^*(x_l)=-\mu_l\tanh\Bigg(&\frac{R_l^{-1}g_l^\top}{2\mu_l}\left(\nabla\varphi_l^\top(x_l,c(x_l))W_l\right.\\&\left.+\nabla\mathcal{B}(x_l,x_o)+\nabla\varepsilon_l\right)\Bigg)\end{aligned} \tag{24}$$

where $W_l\in\mathbb{R}^{n_{\varphi_l}}$ is the ideal weight of leader's StaF NN, $\varphi_l(x_l,c(x_l))\in\mathbb{R}^{n_{\varphi_l}}$ is the kernel function of the StaF NN,

n_{φ_l} is the number of the kernel functions, $c(x_l) \in (B_r(x_l))^{n_{\varphi_l}}$ is the center with neighboring states of the kernel function, $\mathcal{B}(x_l, x_o)$ is the control barrier function defined in (4), $\varepsilon_l(x_l)$ is the approximation error of the leader's optimal value function. Furthermore, by employing the NN, the optimal value function and control inputs of the follower agent are approximated as:

$$V_f^*(\epsilon) = W_f^\top \varphi_f(\epsilon) + \varepsilon_f(\epsilon) \tag{25}$$

$$u_f^*(\epsilon) = -\mu_f \tanh\left(\frac{R_f^{-1} g_f^\top}{2\mu_f}\left(\nabla\varphi_f^\top(\epsilon) W_f + \nabla\varepsilon_f\right)\right) \tag{26}$$

where $W_f \in \mathbb{R}^{n_{\varphi_f}}$ is the ideal weight of follower's NN, $\varphi_f(\epsilon) \in \mathbb{R}^{n_{\varphi_f}}$ is the kernel function of the follower's NN, n_{φ_f} is the number of the kernel functions, $\varepsilon_f(\epsilon)$ is the approximation error of the follower's optimal value function. The ideal weights W_l and W_f are unknown in practice. The actor-critic NN structure is employed to estimate the optimal value functions and the control inputs, respectively:

$$\hat{V}_l(x_l) = \hat{W}_{l,c}^\top \varphi_l(x_l, c(x_l)) + \mathcal{B}(x_l, x_o) \tag{27}$$

$$\hat{V}_f(\epsilon) = \hat{W}_{f,c}^\top \varphi_f(\epsilon) \tag{28}$$

$$\hat{u}_l(x_l) = -\mu_l \tanh\left(\frac{R_l^{-1} g_l^\top}{2\mu_l}\left(\nabla\varphi_l^\top(x_l)\hat{W}_{l,a} + \nabla\mathcal{B}(x_l, x_o)\right)\right) \tag{29}$$

$$\hat{u}_f(\epsilon) = -\mu_f \tanh\left(\frac{R_f^{-1} g_f^\top}{2\mu_f}\nabla\varphi_f^\top(\epsilon)\hat{W}_{f,a}\right) \tag{30}$$

where $\hat{W}_{l,c} \in \mathbb{R}^{n_{\varphi_l}}$, $\hat{W}_{f,c} \in \mathbb{R}^{n_{\varphi_f}}$ are the estimated weights of the critic NNs. $\hat{W}_{l,a} \in \mathbb{R}^{n_{\varphi_l}}$, $\hat{W}_{f,a} \in \mathbb{R}^{n_{\varphi_f}}$ are the estimated weights of the actor NNs. Then Bellman errors are generated by integrating the estimated value functions (27) (28) and control inputs (29) (30) with the Hamilton function (18):

$$\begin{aligned}\delta_l(x_l, \hat{W}_{l,c}, \hat{W}_{l,a}) = & r_l(x_l, \hat{u}_l(x_l, \hat{W}_{l,a})) \\ & + \left(\nabla\varphi_l^\top(x_l, c(x_l))\hat{W}_{l,c} + \nabla\mathcal{B}\right)^\top \\ & \times \left(F_l(x_l) + G_l(x_l)\hat{u}_l(x_l, \hat{W}_{l,a})\right)\end{aligned} \tag{31}$$

$$\begin{aligned}\delta_f(\epsilon, \hat{W}_{f,c}, \hat{W}_{f,a}) = & r_f(\epsilon, \hat{u}_f(\epsilon, \hat{W}_{f,a})) + \left(\nabla\varphi_f^\top(\epsilon)\hat{W}_{f,c}\right)^\top \\ & \times \left(F_f(\epsilon) + G_f(\epsilon)\hat{u}_f(\epsilon, \hat{W}_{f,a})\right)\end{aligned} \tag{32}$$

where $\delta_l(x_l, \hat{W}_{l,c}, \hat{W}_{l,a})$ and $\delta_f(\epsilon, \hat{W}_{f,c}, \hat{W}_{f,a})$ are the Bellman errors of the leader and follower agents, respectively. By utilizing the Bellman errors to update the weights, it can train the actor-critic NNs in an online manner. Then the trained actor-critic NNs could be used to obtain the optimal prescribed performance safe controller Fig. 2 illustrates the developed structure of the HSRL-PPC scheme. The leader's value function $V_l(x_l, x_o)$ includes follower's tracking error ϵ, while the follower's value function $V_f(\epsilon, x_l)$ depends on the leader's state. This creates a coordinated framework where the leader plans paths considering follower's capabilities, and the follower optimizes control to maintain prescribed performance bounds relative to the leader's trajectory. This information flow enables agents to anticipate each other's behavior during both planning and execution

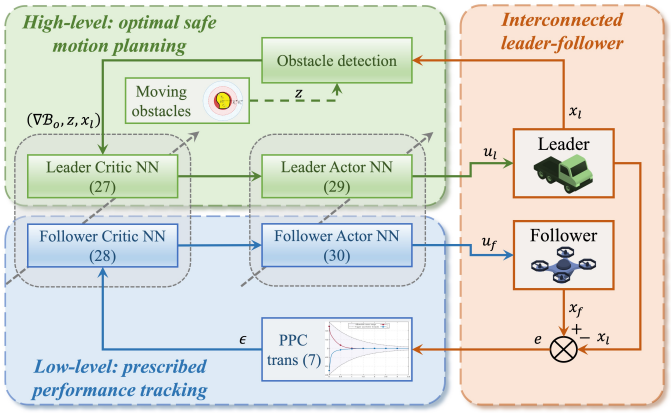


Fig. 2. Structure of the proposed HSRL-PPC scheme for leader-follower.

Remark 5 (Real and Virtual Leader-Follower Agents)**.** *In traditional hierarchical control schemes, leader-follower structures typically exist in two fixed arrangements: (1) virtual leaders paired with real executor followers [18], [20], or (2) real-world leader-follower pairs where leaders have superior capabilities [14], [16], [17]. However, these approaches are often not interchangeable. Our proposed HSRL-PPC algorithm 1 provides a general and flexible control scheme for interconnected leader-follower systems, accommodating both virtual planners and real-world executors as leaders, making it more adaptable for diverse practical applications.*

Remark 6 (Value Functions Coupling)**.** *Hierarchical structure implements coupling between agents where the leader's decisions influence the follower's policy and vice versa. The leader's value function includes follower tracking error while the follower's value function incorporates the leader's state. This bi-directional dependency enables coordinated behavior where the leader plans paths considering follower capabilities, while the follower maintains prescribed performance bounds relative to the leader's trajectory. This coupling is essential for effective performance in interconnected systems.*

Remark 7 (Complementary Constraints)**.** *Our approach separates constraints across system levels: barrier functions for the leader's obstacle avoidance and PPC for the follower's tracking. This hierarchical design, together with control effort penalties $\Phi(U)$ in the reward function, prevents excessive control actions. Experiments in Section VI confirm this efficiency, showing 5.25% lower control effort compared to unconstrained approaches while maintaining safety guarantees.*

B. Online value function approximation

In this subsection, weights of actor-critic NNs are updated online through the minimization of the Bellman errors. To update the weights of the leader agent's actor-critic NNs, a state extrapolation approach is employed. This approach involves simulating a collection of neighboring states $\{x_l^k\}_{k=1}^N, (x_l^k \in B_r(x_l(t)))$ of the kernel function. The extrapolated control input of the leader agent is derived as:

$$\hat{u}_l^k(t) = -\mu_l \tanh\left(\frac{R_l^{-1} g_l^\top}{2\mu_l}\left(\nabla\varphi_l^\top(x_l^k)\hat{W}_{l,a} + \nabla\mathcal{B}(x_l^k, x_o)\right)\right)$$

where $x_l^k(t) \in B_r(x_l(t)), k = 1, 2, \cdots, N$, are the neighboring states of the kernel function. The extrapolated control input is utilized to simulate the neighboring bellman errors of the current state, as specified by:

$$\delta_l^k(t) = r_l(x_l^k, \hat{u}_l^k) + \left(\nabla\varphi_l^\top(x_l^k, c(x_l))\hat{W}_{l,c} + \nabla\mathcal{B}(x_l^k, x_o)\right)^\top \times \left(F_l(x_l^k) + G_l(x_l^k)\hat{u}_l^k\right) \quad (33)$$

Then the dataset of the leader's extrapolated states could be obtained, i.e. $\{\hat{u}_l(t), \delta_l(t), \{\hat{u}_l^k(t), \delta_l^k(t)\}_{k=1}^N\}$, where $\{\hat{u}_l^k(t), \delta_l^k(t)\}$ is the k-th extrapolated data collection. Different from the leader agent, the dataset of the follower agent's states is collected without extrapolation but stored as a historical stack, i.e. $\{\hat{u}_f(t), \delta_f(t), \{\hat{u}_f^j(t), \delta_f^j(t)\}_{k=1}^N\}$, where $\{\hat{u}_f^j(t), \delta_f^j(t)\}$ is the j-th historical stored data collection. Then the weights of the actor-critic NNs can be obtained by minimizing the squared loss function given as:

$$E_{\mathbf{i}} = \delta_{\mathbf{i}}(t)^\top \delta_{\mathbf{i}}(t) + \sum_{k=1}^{N} \delta_{\mathbf{i}}^k(t)^\top \delta_{\mathbf{i}}^k(t), \ \mathbf{i} = l, f \quad (34)$$

To minimize the loss function (34), a concurrent-learning-based update law is employed to update critic NNs weights:

$$\dot{\hat{W}}_{\mathbf{i},c} = -k_{\mathbf{i},c1}\frac{\delta_{\mathbf{i}}(t)\sigma_{\mathbf{i}}(t)}{\left(\sigma_{\mathbf{i}}(t)^\top\sigma_{\mathbf{i}}(t)+1\right)^2} - \frac{k_{\mathbf{i},c2}}{N}\sum_{k=1}^{N}\frac{\delta_{\mathbf{i}}^k(t)\sigma_{\mathbf{i}}^k(t)}{\left((\sigma_{\mathbf{i}}^k(t))^\top\sigma_{\mathbf{i}}^k(t)+1\right)^2}, \ \mathbf{i} = l, f \quad (35)$$

where $k_{\mathbf{i},c\mathbf{j}} > 0, (\mathbf{i} = l, f, \ \mathbf{j} = 1, 2)$ are the learning rates of critic NNs, and functions $\sigma_l = \nabla\varphi_l^\top(x_l, c(x_l))(f_l(x_l) + g_l(x_l)\hat{u}_l(x_l))$, $\sigma_l^k = \nabla\varphi_l^\top(x_l^k, c(x_l))(f_l(x_l^k) + g_l(x_l^k)\hat{u}_l^k)$, $\sigma_f = \nabla\varphi_f^\top(\epsilon)(f_f(\epsilon) + g_f(\epsilon)\hat{u}_f(\epsilon))$, $\sigma_f^j = \nabla\varphi_f^\top(\epsilon^j)(f_f(\epsilon^j) + g_f(\epsilon^j)\hat{u}_f^j)$. For the actor NNs, to update the weights while keeping bounded, a gradient law with projection is employed:

$$\dot{\hat{W}}_{\mathbf{i},a} = \text{Proj}\left(-k_{\mathbf{i},a}F_{\mathbf{i}}\left(\hat{W}_{\mathbf{i},a} - \hat{W}_{\mathbf{i},c}\right)\right), \ \mathbf{i} = l, f \quad (36)$$

where $k_{\mathbf{i},a} > 0, \mathbf{i} = l, f$ are the learning rates of actor NNs. $F_{\mathbf{i}} \in \mathbb{R}^{n_{\varphi_l} \times n_{\varphi_l}}$, $\mathbf{i} = l, f$ are constant positive definite matrices for the update of the actor NNs. Proj$(\cdot)$ is a projection operator to ensure that the weights of the actor NNs are kept within specified constraints [34]. Subsequently, online learning of the HSRL-PPC is completed. Algorithm 1 illustrates the detailed procedure for learning the HSRL-PPC.

Remark 8 (Extension to Multi-Agent Systems and Complex Interconnections)**.** *Our framework can be extended to more complex interconnected or multi-agent systems in several ways: (1) Multiple hierarchical layers where agents serve as both followers and leaders in a command chain; (2) Heterogeneous agent teams with different dynamics and capabilities using adapted performance bounds; (3) Various communication topologies including directed graphs and time-varying networks; and (4) Human-in-the-loop integration where operators provide high-level commands within the hierarchical structure. While these extensions would require modified stability analysis, the core safe guarantees would remain applicable across more complex multi-agent configurations.*

Algorithm 1 HSRL-PPC Scheme

1: Initialize the weights of the leader and the follower agents' actor-critic NNs, $\hat{W}_{l,c}$, $\hat{W}_{l,a}$, $\hat{W}_{f,c}$, $\hat{W}_{f,a}$. Set up the online learning parameters and the termination condition d_{end}.
2: **while** $\|x_l\| > d_{\text{end}}$ **do**
3: Collect current state data $x_l(t)$, $x_f(t)$, $x_o(t)$.
4: Execute control inputs $\hat{u}_l(t)$, $\hat{u}_f(t)$ from (29)-(30).
5: Compute Bellman errors $\delta_l(t)$, $\delta_f(t)$ from (31)-(32), and the extrapolated Bellman error $\delta_l^k(t)$ from (33).
6: Update critic NNs by concurrent learning law (35).
7: Update actor NNs by the gradient projection law (36).
8: **end while**

C. Stability analysis

In this subsection, the closed-loop system states $[x_l^\top, \epsilon^\top]^\top$ and the actor-critic NN weights errors is proved to be ultimate uniform bounded (UUB) under the control of the proposed HSRL-PPC scheme. First, three assumptions are given.

Assumption 2. *The following assumptions are given for the optimal prescribed performance safe control problem:*

1) *On a tight set $X \in \chi \in \mathbb{R}^n$, both $F(X)$ and $G(X)$ are Lipschitz continuous with $F(0) = 0$, and $G(X)$ satisfied bounded condition $\|G(X)\| \leq G_H$ for all $X \in \chi$.*
2) *Cost matrix $Q_{x_\mathbf{i}}$ and $R_{\mathbf{i}}$ $(\mathbf{i} = f, l)$ are bounded, such that $\underline{\lambda}_{Q,\mathbf{i}} \leq \|Q_{x_\mathbf{i}}\| \leq \bar{\lambda}_{Q,\mathbf{i}}$, $\underline{\lambda}_{R,\mathbf{i}} \leq \|R_\mathbf{i}\| \leq \bar{\lambda}_{R,\mathbf{i}}$, where constants $\underline{\lambda}_{Q,\mathbf{i}}, \underline{\lambda}_{R,\mathbf{i}} \geq 0$ and $\bar{\lambda}_{Q,\mathbf{i}}, \bar{\lambda}_{R,\mathbf{i}} > 0$.*

Assumption 3. *Assuming that the following parameters and operators are bounded: $\|\hat{W}_{l,c}\| \leq W_{H1}$, $\|\hat{W}_{f,c}\| \leq W_{H2}$, $\|\sigma_l(x_l)\| \leq \sigma_{H1}$, $\|\sigma_f(x_f)\| \leq \sigma_{H2}$, $\|\nabla\sigma_l(x_l)\| \leq \sigma_{D,H1}$, $\|\nabla\sigma_f(x_f)\| \leq \sigma_{D,H2}$, $\|\varphi_l(x_l, c(x_l))\| \leq \varphi_{H1}$, $\|\nabla\varphi_l(x_l, c(x_l))\| \leq \varphi_{D,H1}$, $\|\varphi_f(x_f)\| \leq \varphi_{H2}$, $\|\nabla\varphi_f(x_f)\| \leq \varphi_{D,H2}$, $\|\varepsilon_l(x_l)\| \leq \varepsilon_{H1}$, $\|\nabla\varepsilon_l(x_l)\| \leq \varepsilon_{D,H1}$, $\|\varepsilon_f(x_f)\| \leq \varepsilon_{H2}$, $\|\nabla\varepsilon_f(x_f)\| \leq \varepsilon_{D,H2}$.*

Assumption 4. *Assuming that the online collected and extrapolated data set for the weights update law satisfies the following excitation condition:*

$$\vartheta_{1,\mathbf{i}} I_{\mathcal{L},\mathbf{i}} \leqslant \int_t^{t+T}\left(\frac{\sigma_{\mathbf{i}}(\tau)\sigma_{\mathbf{i}}(\tau)^\top}{\rho_{\mathbf{i}}(\tau)}\right)\mathrm{d}\tau, \ \mathbf{i} = l, f$$
$$\vartheta_{2,\mathbf{i}} I_{\mathcal{L},\mathbf{i}} \leqslant \inf_{t\in\mathbf{R}_{t\geq t_0}}\left(\frac{1}{N}\sum_{k=1}^{N}\frac{\sigma_{\mathbf{i}}^k(t)\sigma_{\mathbf{i}}^k(t)^\top}{\rho_{\mathbf{i}}^k(t)}\right), \ \mathbf{i} = l, f \quad (37)$$

where $\rho_{\mathbf{i}} = \left(\sigma_{\mathbf{i}}^\top(t)\sigma_{\mathbf{i}}(t) + 1\right)^2$, $\rho_{\mathbf{i}}^k = \left(\sigma_{\mathbf{i}}^k(t)^\top\sigma_{\mathbf{i}}^k(t) + 1\right)^2$, and at least one of nonnegative constants $\vartheta_{1,\mathbf{i}}, \vartheta_{2,\mathbf{i}}$ is positive.

To simplify the analysis, the approximated Hamiltonian error $\delta_{\mathbf{i}}$, $\mathbf{i} = l, f$, or Bellman error, is abbreviated to the following form:

$$\delta_{\mathbf{i}} = -\sigma_{\mathbf{i},c}^\top\tilde{W}_{\mathbf{i},a} + \frac{1}{4}\tilde{W}_{\mathbf{i},a}G_{\mathbf{i},\sigma}\tilde{W}_{\mathbf{i},a} + \Delta_{\mathbf{i}} + \xi_{\mathbf{i},H}, \ \mathbf{i} = l, f \quad (38)$$

$$\delta_{\mathbf{i}}^k = -(\sigma_{\mathbf{i},c}^k)^\top\tilde{W}_{\mathbf{i},a} + \frac{1}{4}\tilde{W}_{\mathbf{i},a}G_{\mathbf{i},\sigma}^k\tilde{W}_{\mathbf{i},a} + \Delta_{\mathbf{i}}^k, \ \mathbf{i} = l, f \quad (39)$$

where $G_{\mathbf{i},\sigma} = \nabla\varphi_{\mathbf{i}}^\top g_{\mathbf{i}}R_{\mathbf{i}}^{-1}g_{\mathbf{i}}^\top\nabla\varphi_{\mathbf{i}}^\top$, $G_{\mathbf{i},\sigma}^k = \nabla\varphi_{\mathbf{i}}^\top g_{\mathbf{i}}R_{\mathbf{i}}^{-1}g_{\mathbf{i}}^\top\nabla\varphi_{\mathbf{i}}^\top$. According to the assumption 2 and 3, $\xi_{\mathbf{i},H}$ is a bounded residual with respect to the construction error $\varepsilon_{\mathbf{i}}$, and $\Delta_{\mathbf{i}}, \Delta_{\mathbf{i}}^k \in \mathbb{R}^{n_{\mathbf{i}}}$

TABLE I
THE PARAMETERS OF THE LEADER-FOLLOWER SYSTEM.

Example	Initial conditions	Controller parameters	Barrier function parameters	Weights update parameters
Example 1: Nonlinear leader-follower system	$x_l(0) = [2,3]^\top$ $x_f(0) = [2.5,2.5]^\top$	$R_l = I_2, R_f = I_2$ $Q_l = I_3, Q_f = 20I_2$ $\mu_l = 0.5, \mu_f = 1.5$ $\rho_{i0} = 0.6, \rho_{i\infty} = 0.01$ $\lambda = 0.9,\ \zeta = 1$	$K_B = 1, \mu = 0.5$ $L_1 = 0.3, L_2 = 0.8$ $L_3 = 0.2, r_o = 0.2$ $r_a = 0.3, r_d = 0.4$ $Q_{X_{o,i}} = I_2$	$k_{l,c1} = k_{f,c1} = 0.5$ $k_{l,c2} = k_{f,c2} = 0.1$ $k_{l,a} = 1, k_{f,a} = 1$ $F_l = I_3$ $F_f = I_6$
Example 2: Follower quadcopter landing at moving leader vehicle	$x_l(0) = [2,3]^\top$ $x_f(0) = [3,2,1]^\top$	$R_l = I_2, R_f = I_3$ $Q_l = I_3, Q_f = 20I_3$ $\mu_l = 0.5, \mu_f = 1.5$ $\rho_{i0} = 1.1, \rho_{i\infty} = 0.01$ $\lambda = 0.8,\ \zeta = 1$	$K_B = 1, \mu = 0.5$ $L_1 = 0.3, L_2 = 0.8$ $L_3 = 0.2, r_o = 0.2$ $r_a = 0.3, r_d = 0.4$ $Q_{X_{o,i}} = I_2$	$k_{l,c1} = k_{f,c1} = 0.5$ $k_{l,c2} = k_{f,c2} = 0.1$ $k_{l,a} = 1, k_{f,a} = 1$ $F_l = I_3$ $F_f = I_6$

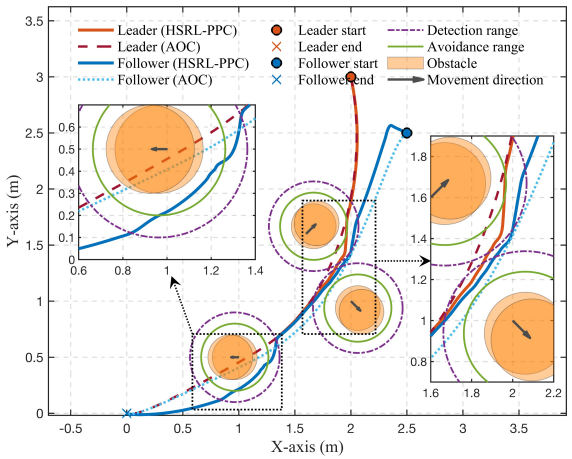


Fig. 3. Example 1: Trajectory of the leader-follower in 2-dimensional space.

are uniformly bounded on χ, $\overline{\|\Delta_{\mathbf{i}}\|}$ and $\overline{\|\Delta_{\mathbf{i}}^k\|}$ decrease as $\overline{\|\nabla\varepsilon_{\mathbf{i}}\|}$ and $\overline{\|\nabla W_{\mathbf{i},a}\|}$ decrease.

Based on the design of the controllers (29) and (30), the following inequality could be obtained:

$$\|u_{\mathbf{i}}^* - \hat{u}_{\mathbf{i}}\|^2 \leq \Sigma_{u_{\mathbf{i}}} \tilde{W}_{\mathbf{i},a}^\top \tilde{W}_{\mathbf{i},a} + \Pi_{u_{\mathbf{i}}},\ \mathbf{i} = l, f \tag{40}$$

where $\Sigma_{u_{\mathbf{i}}}$ is a upper bound related to $\varphi_{H,1}$, $\varphi_{H,2}$, $\varphi_{D,H1}$, $\varphi_{D,H2}$, σ_{H1}, σ_{H2}, $\sigma_{D,H1}$and $\sigma_{D,H2}$, $\tilde{W}_{\mathbf{i},c} = W_{\mathbf{i},c} - \hat{W}_{\mathbf{i},c}$ and $\tilde{W}_{\mathbf{i},a} = W_{\mathbf{i},a} - \hat{W}_{\mathbf{i},a}$ are the estimated error of the actor-critic NNs, $\Pi_{u_{\mathbf{i}}}$ is a upper bound related to $\varepsilon_{D,H1}$ and $\varepsilon_{D,H2}$.

The stability analysis of closed-loop system state $[x_l^\top, \epsilon^\top]^\top$, weight errors $\tilde{W}_{\mathbf{i},c}$ and $\tilde{W}_{\mathbf{i},a}$ is given in the theorem 1.

Theorem 1. *Considering the augmented system (11), assumption 1-4 are satisfied, and algorithm 1 is implemented. The actor-critic NNs are updated by the adaptive update law (35) and (36). Then the close-loop system states $[x_l^\top, \epsilon^\top]^\top$ and estimated weight errors $\tilde{W}_{\mathbf{i},c}, \tilde{W}_{\mathbf{i},a}$ will be UUB provided that $\|Z\| \geq \sqrt{\Pi_{\mathrm{ALL}}/\lambda_{\min}(\mathcal{H})}$, where $Z = \left[x_l^\top, \epsilon^\top, \tilde{W}_{l,c}^\top, \tilde{W}_{l,a}^\top, \tilde{W}_{f,c}^\top, \tilde{W}_{f,a}^\top\right]^\top$.*

Proof. Theorem 1 is proved based on the Lyapunov stability theory. The detailed proof is given in the appendix A. □

V. NUMERICAL SIMULATIONS

In this section, two numerical simulation examples are conducted to verify the effectiveness of the proposed HSRL-PPC scheme.

A. Example 1: Nonlinear leader-follower system

Simulation setup: In this example, a representative nonlinear leader-follower system is designed to complete the path-planning and tracking control task with moving obstacles in 2-dimensional space. The dynamic models of the leader-follower system (1) are both selected as nonlinear affine system with detailed parameters cited from references [29], [32]:

$$f_{\mathbf{i}}(x_{\mathbf{i}}) = \begin{bmatrix} -x_{\mathbf{i},1} + x_{\mathbf{i},2} \\ -\frac{1}{2}x_{\mathbf{i},1} - \frac{1}{2}x_{\mathbf{i},2}\left(1 - (\cos(2x_{\mathbf{i},1}) + 2)^2\right) \end{bmatrix},$$
$$g_{\mathbf{i}}(x_{\mathbf{i}}) = \begin{bmatrix} \sin(2x_{\mathbf{i},1}) + 2 & 0 \\ 0 & \cos(2x_{\mathbf{i},1}) + 2 \end{bmatrix},\ \mathbf{i} = l, f.$$

Design three moving obstacles O_1, O_2, and O_3 with the dynamics described by the following linear systems:

$$g_{o,1}(x_{o,1}) = 0.06s_1\,[1,1]^\top, x_{o,1}(0) = [1.6, 1.6]^\top,$$
$$g_{o,2}(x_{o,2}) = 0.06s_2\,[1,-1]^\top, x_{o,2}(0) = [2, 1.0]^\top,$$
$$g_{o,3}(x_{o,3}) = 0.06s_3\,[-1,0]^\top, x_{o,3}(0) = [1, 0.5]^\top,$$

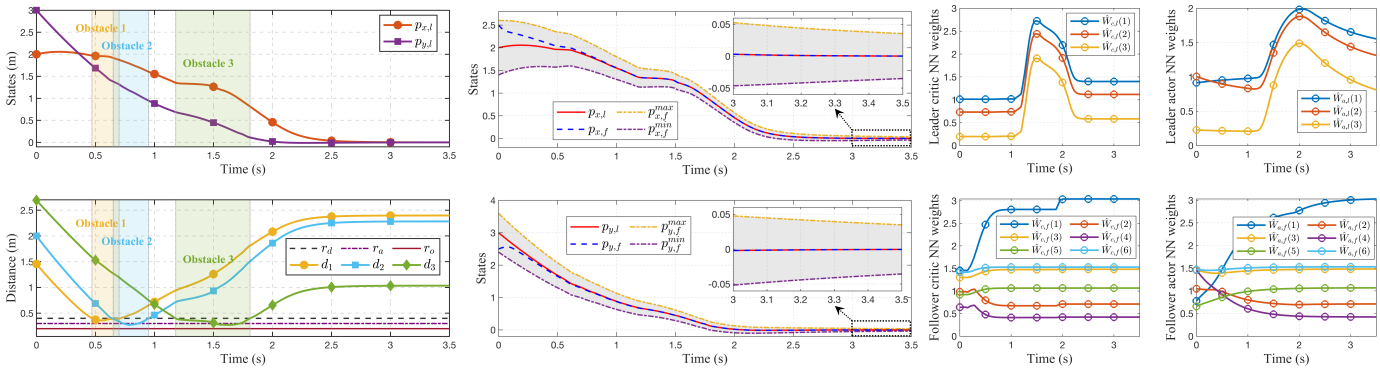
where s_1, s_2, s_3 are the functions defined in (5). For the actor-critic NNs, the weights are initialized randomly as $\hat{W}_{l,c} = \hat{W}_{l,a} = \mathrm{rand}(n_{\varphi l}, 1)$, $\hat{W}_{f,c} = \hat{W}_{f,a} = \mathrm{rand}(n_{\varphi f}, 1)$, where $n_{\varphi_l} = 3$, $n_{\varphi_f} = 3$. The leader's NNs are chosen as the StaF NNs, with its basis function being chosen as $\varphi_l(x_l, c) = [\varphi_1(x_l, c_1(x_l)), \varphi_2(x_l, c_2(x_l)), \varphi_3(x_l, c_3(x_l))]^\top$. The kernel functions of the basis function are designed as $\varphi_i(x_l, c_i(x_l)) = x_l^\top c_i(x_l) - 1, i = 1,2,3$, where $c_i(x_l) = x_i + d_i(x_i), i = 1,2,3$, $d_1(x_i) = 0.001v(x_i) \times [0,1]^\top$, $d_2(x_i) = 0.001v(x_i) \times [0.866, -0.5]^\top$, $d_3(x_i) = 0.001v(x_i) \times [-0.866, -0.5]^\top$, and $v(x_i) = (x_i^\top x_i + 0.01)/(x_i^\top x_i + 1)$. The NNs of the follower are chosen in the following form:

$$\varphi_f(x_f) = [x_f(1)^2, x_f(1) \times x_f(2), x_f(2)^2, x_f(1)^2 \times x_f(2),$$
$$x_f(1) \times x_f(2)^2, x_f(1)^2 \times x_f(2)^2]^\top.$$

The leader's task is to obtain the optimal path around moving obstacles, while the follower is required to track the leader within the prescribed performance bounds. To control the leader and follower, the proposed hierarchical controllers (29) and (30) are employed, and the weights of the actor-critic NNs are updated online by the law (35) and (36). Table I shows the detailed parameters designed for the simulation.

TABLE II
RESULTS OF NUMERICAL SIMULATIONS AND EXPERIMENTS.

Case	PL	RPL	Minimum Distance to Obstacles			Relative Min. Distance			Final
	(m)	(%)	MDO_1 (m)	MDO_2 (m)	MDO_3 (m)	RMDO_1 (%)	RMDO_2 (%)	RMDO_3 (%)	FDTP (m)
Simulation 1	3.98	94.34	0.37	0.28	0.27	186.00	141.54	132.71	7.88e-08
Simulation 2	5.03	114.12	0.36	0.36	0.38	179.74	181.19	190.76	1.04e-06
Experiment 1	46.19	109.19	—	—	—	—	—	—	0.02
Experiment 2	6.48	137.32	0.31	—	—	205.28	—	—	0.02



(a) (b) (c)

Fig. 4. Example 1: (a) Position of the leader and distance to the moving obstacles. (b) Position of the follower. (c) Revolution of NNs weights.

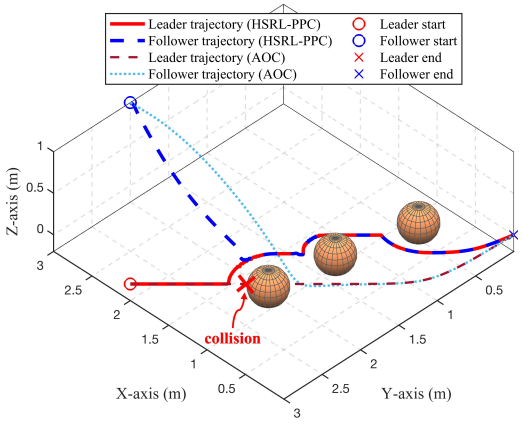


Fig. 5. Example 2: Trajectory of the leader-follower in 3-dimensional space.

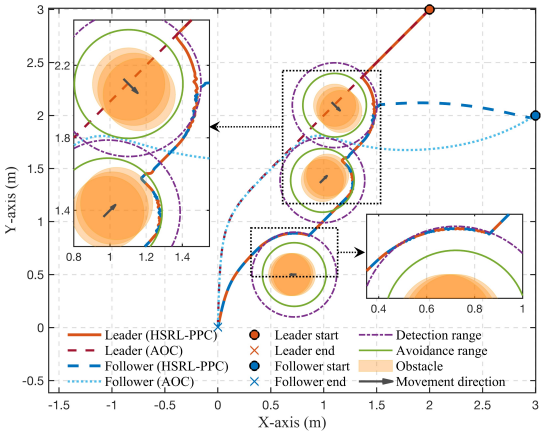


Fig. 6. Example 2: Trajectory of the leader-follower in 2-dimensional space.

Result: The main results of example 1 are shown in Fig. 3-4. In Fig. 3, the leader obtains the optimal path and avoids three moving obstacles. Regarding the follower, it cannot sense obstacles, and simply follows the leader's trajectory within prescribed performance bounds to avoid collisions, while the comparison AOC method [26], [45] fails to guarantee the safety of the leader, also the tracking performance of the follower is not guaranteed. Fig. 4(a) shows the position of the leader and its real-time distance to three moving obstacles. Fig. 4(b) provides the position of the follower, where the tracking error is bounded by prescribed performance bounds $p_{i,l}^{max}$ and $p_{i,l}^{min}, i = x, y, z$. Fig. 4(c) illustrates the revolution of the actor-critic NNs weights.

B. Example 2: Follower quadcopter landing at leader vehicle

Simulation setup: In this example, the leader-follower system is designed to complete a landing task with a moving leader vehicle. The leader is chosen as a moving vehicle in the 'X-Y' plane, and the follower is selected as a quadcopter

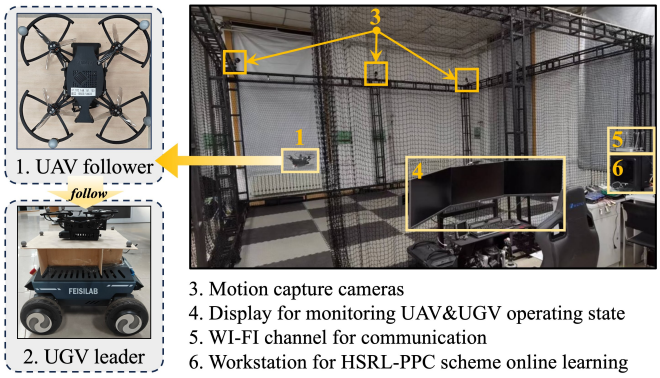


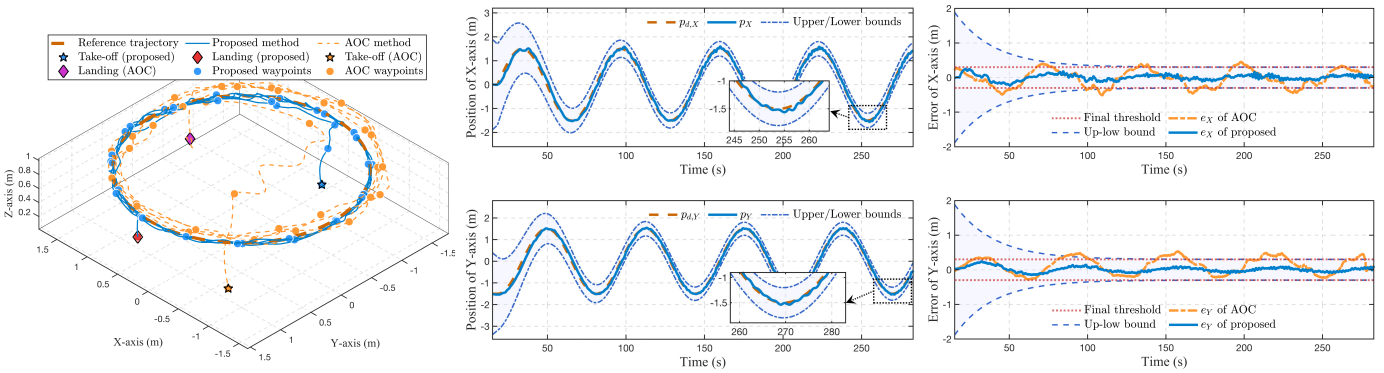
Fig. 7. The hardware setup for experimental validation.

moving in 'X-Y-Z' 3-dimensional space. The dynamic models of the leader-follower system are designed as:

$$f_l = 0_{2\times1},\ g_l = I_{2\times2}, x_l \in \mathcal{R}^2, u_l \in \mathcal{R}^2,$$
$$f_f = 0_{3\times1},\ g_f = I_{3\times3}, x_f \in \mathcal{R}^3, u_f \in \mathcal{R}^3, \qquad (41)$$

TABLE III
PERFORMANCE COMPARISON BETWEEN AOC AND PROPOSED CONTROLLER

	MSE			Max	Total	Mean Input		Peak	RMS			Cost	
Method	**X**	**Y**	**Total**	**Error**	**Energy**	**X**	**Y**	**Input**	**X**	**Y**	**Total**	**X**	**Y**
AOC	0.0688	0.0644	0.0666	0.5324	5.5964	**0.1001**	0.1016	**0.3299**	**0.1193**	0.1253	0.1224	**2.6598**	2.9352
Proposed PPC	**0.0082** ↓	**0.0075** ↓	**0.0078** ↓	**0.2540** ↓	**5.3174** ↓	0.1054	**0.0995** ↓	0.3445	0.1241	**0.1142** ↓	**0.1193** ↓	2.8789	**2.4380** ↓



(a) (b) (c)

Fig. 8. Case 1: (a) Trajectory of the quadcopter in 3-dimensional space. (b) Position of the quadcopter in $X-Y$ plane. (c) Comparison of the tracking error.

which is a widely used "position-velocity" kinematic model [9], [29], [32]. Similar to the previous example, the leader is required to obtain an approximate optimal path and avoid obstacles, while the follower tracks the leader in prescribed performances. However, the follower is also required to land on the leader vehicle in this example. The actor-critic NNs for the leader are designed as the StaF NNs in the same form as the previous example. The NNs for the follower are selected in the following form:

$$\varphi_f(x_f) = [x_f(1)^2, x_f(2)^2, x_f(3)^2, x_f(1)\times x_f(2), x_f(1)\times x_f(3), x_f(2)\times x_f(3)]^\top.$$

Result: The main results of example 2 are shown in Fig. 5-6. In Fig. 5, the leader-follower system successfully completes the landing task in 3-dimensional space using our proposed HSRL-PPC method, while the comparative AOC method fails due to collision. Fig. 6 shows the trajectory in 2-dimensional space, where our approach guarantees both leader and follower safety through effective path planning and obstacle avoidance. In contrast, the AOC method collides with the first moving obstacle, demonstrating the superior performance of our hierarchical safe control framework.

To evaluate the performance of the proposed HSRL-PPC method, several performance indexes are defined: (1) Path length (PL), which is the approximated optimal path length of the leader, a smaller value indicates a better performance of path planning; (2) Minimum distance to the ith obstacle (MDO$_i$), a bigger value indicates a safer performance; (3) Relative path length (RPL), which is the ratio of the follower's trajectory length to the leader's trajectory length, a value close to 1 indicates a better performance; (4) Relative minimum distance to the ith obstacle (RMDO$_i$), which is the ratio of MDO$_i$ to the obstacle radius r_i; (5) Final distance to the target point (FDTP), which reflects the stabilization error to the final point; The detailed value of the performance index for the numerical simulations is shown in Table II.

VI. HARDWARE EXPERIMENTS

A. Experiment setup

In this section, hardware experiments are conducted to verify the effectiveness of the proposed HSRL-PPC scheme. The experiment is performed on a leader vehicle and a follower quadcopter. The leader vehicle is a 4-wheel drive car, and the follower quadcopter is an X150 quadcopter. Both the leader vehicle and the follower quadcopter are equipped with an RK3566 processor and 4-GB RAM. The real-time position of the vehicle and quadcopter are obtained by an 8-cameras motion capture system. The developed controller (29) and (30) are calculated by a workstation equipped with an Intel i7-12700 CPU @3.60 GHz. The control inputs are implemented as velocity commands through 5GHz WI-FI channel.

Case 1: Verification of lower-level PPC of the follower

To evaluate the performance of lower-level PPC, a case of quadcopter tracking horizontal circle trajectory is conducted. The circle is set as $x_{f,d} = [1.5\sin(0.2t), 1.5\cos(0.2t), 1]^\top$.

Experiment results: The experiment results are shown in Fig. 8. Fig. 8(a) shows the trajectory of the follower quadcopter in 3-dimensional space. Fig. 8(b) provides the position of the follower quadcopter in the $X-Y$ plane, which shows that the follower quadcopter tracks reference trajectory within a prescribed performance. Fig. 8(c) compares tracking error performance between our proposed method (blue line) and the approximate optimal control (AOC) method (orange line). The results clearly demonstrate that our PPC-based approach maintains significantly smaller tracking errors that consistently remain within the prescribed performance bounds, unlike the AOC method which exhibits larger deviations. This confirms the superior capability of our lower-level PPC controller for precise trajectory tracking even under disturbances. Detailed performance metrics are provided in Table II, with comprehensive comparison results shown in Table III. Across multiple

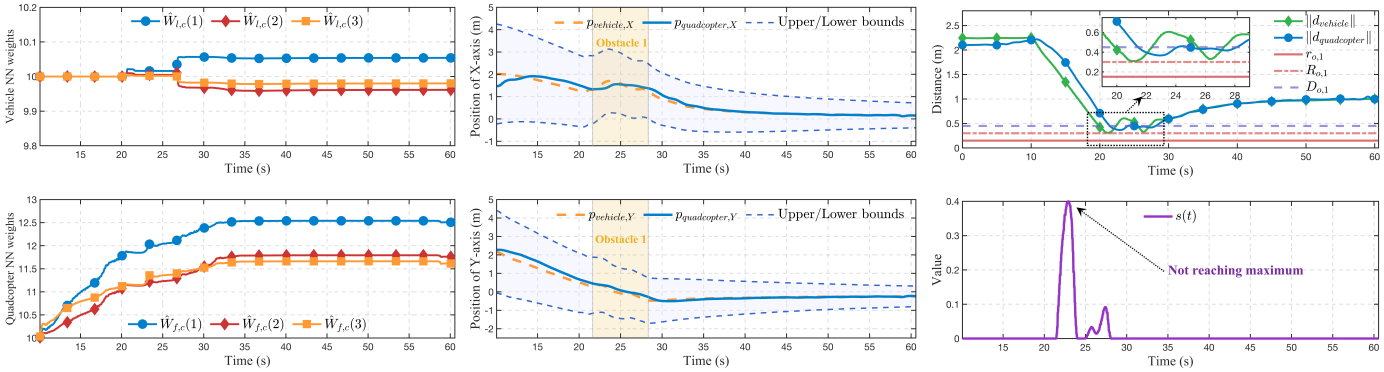


Fig. 9. Case 2: (a) Weights of the critic NNs. (b) Position of the leader and follower in $X-Y$ plane. (c) Distance to the obstacles and the value of s.

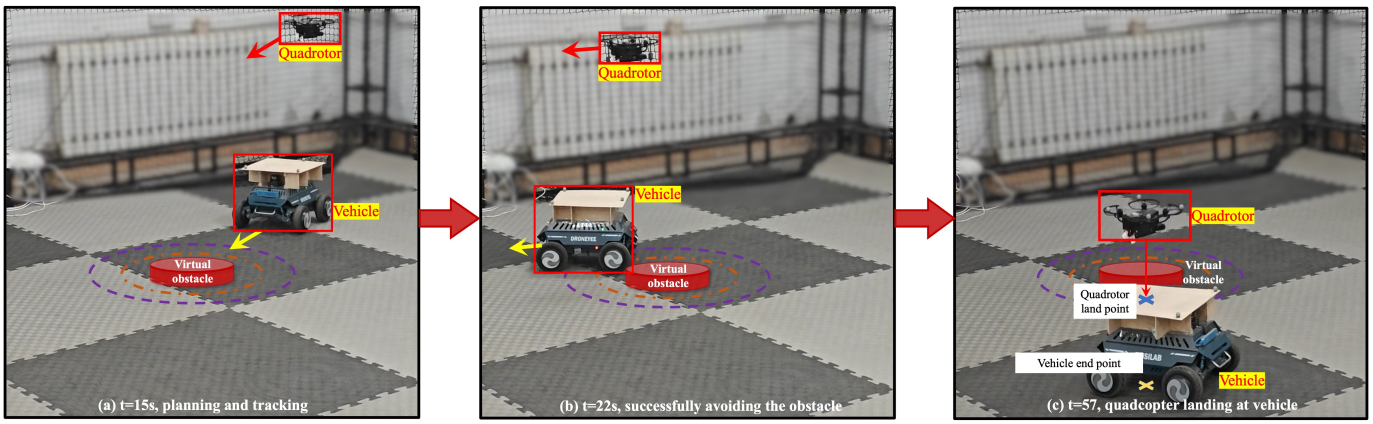


Fig. 10. Case 2: snapshots of the leader-follower system tracking and landing.

evaluation criteria including mean square error, maximum error, and control energy consumption, our proposed scheme demonstrates superior tracking accuracy and disturbance rejection compared to the conventional AOC controller.

Case 2: Verification of HSRL-PPC of the leader-follower

To verify the effectiveness of the proposed HSRL-PPC scheme, a case of the quadcopter-vehicle system is conducted. The leader vehicle is required to obtain an approximate optimal path and avoid obstacles, while the follower quadcopter is required to track and land on the leader vehicle precisely.

Experiment results: The experiment results are shown in Fig. 9. Fig. 9(a) illustrates the revolution weights of the critic NNs. Fig. 9(b) shows the position of the leader and the follower in the $X-Y$ plane, where the follower tracks the leader vehicle within prescribed performance bounds. Fig. 9(b) provides the real-time distance to the obstacles and the value of s, which shows the vehicle avoids the obstacles successfully. The snapshots of the leader-follower system are shown in Fig. 10, which illustrates the quadcopter landing on the leader vehicle.

VII. Conclusion

An SRL-PPC scheme is proposed for the interconnected leader-follower system. Considering the different capabilities and duties of the leader and the follower. Leader in the higher-level of HSRL-PPC approximates the optimal path and avoids obstacles. while the follower in the lower-level of HSRL-PPC tracks the leader within prescribed performances. Actor-critic neural networks are employed to approximate the value function and the control input of the leader and the follower. The effectiveness is verified by simulations and hardware experiments. The proposed HSRL-PPC scheme provides enhanced safety through obstacle avoidance, improved tracking precision via prescribed performance control, and computational efficiency through its hierarchical structure. Despite its advantages, challenges remain in real-world implementation, including state measurement accuracy requirements, neural network training demands, and sensitivity to communication delays. Future work will address external disturbances, model uncertainties, and extend the approach to multi-agent systems.

Appendix

Proof of Theorem 1: The convergence of the HSRL-PPC algorithm is shown in this appendix. The following Lyapunov function is chosen:

$$V_L(Z,t)=\sum_{\mathbf{i}=f,l}\left(V_{\mathbf{i}}^{*}+\frac{1}{2}\tilde{W}_{\mathbf{i},c}^{\top}\tilde{W}_{\mathbf{i},c}+\frac{1}{2}\tilde{W}_{\mathbf{i},a}^{\top}\tilde{W}_{\mathbf{i},a}\right) \quad (42)$$

Then the derivative of V_L with respect to time is given by:

$$\begin{aligned}\dot{V}_L=&\sum_{\mathbf{i}=f,l}\left(\nabla V_{\mathbf{i}}^{*}\left(f_{\mathbf{i}}+g_{\mathbf{i}}u_{\mathbf{i}}\right)+\tilde{W}_{\mathbf{i},a}^{\top}F_{\mathbf{i}}\left(\tilde{W}_{\mathbf{i},c}-\tilde{W}_{\mathbf{i},a}\right)\right.\\&-\tilde{W}_{\mathbf{i},c}^{\top}\left(-k_{\mathbf{i},c1}\frac{\sigma_{\mathbf{i}}}{\rho_{\mathbf{i}}}\left(-\sigma_{\mathbf{i}}^{\top}\tilde{W}_{\mathbf{i},c}+\frac{1}{4}\tilde{W}_{\mathbf{i},a}^{\top}G_{\mathbf{i},\sigma}\tilde{W}_{\mathbf{i},a}+\Delta_{\mathbf{i}}\right)\right)\\&-\tilde{W}_{\mathbf{i},c}^{\top}\left(-\frac{k_{\mathbf{i},c2}}{N}\sum_{k=1}^{N}\frac{\sigma_{\mathbf{i}}^{k}}{\rho_{\mathbf{i}}^{k}}\frac{1}{4}\tilde{W}_{\mathbf{i},a}^{\top}G_{\mathbf{i},\sigma}^{k}\tilde{W}_{\mathbf{i},a}\right)\\&\left.-\tilde{W}_{\mathbf{i},c}^{\top}\left(-\frac{k_{\mathbf{i},c2}}{N}\sum_{k=1}^{N}\frac{\sigma_{\mathbf{i}}^{k}}{\rho_{\mathbf{i}}^{k}}\left(-(\sigma_{\mathbf{i}}^{k})^{\top}\tilde{W}_{\mathbf{i},c}+\Delta_{\mathbf{i}}^{k}\right)\right)\right)\end{aligned}\tag{43}$$

Substitute Bellman errors (38)-(39) into derivative (43) and employing Young's inequality and assumptions 2-4, the derivative can be rewritten as:

$$\begin{aligned}\dot{V}_L\leq&-Z^{\top}\begin{bmatrix}h_1&0&0&0&0&0\\0&h_2&0&0&0&0\\0&0&h_3&0&0&0\\0&0&h_4&h_5&0&0\\0&0&h_6&0&h_7&0\\0&0&0&h_8&0&h_9\end{bmatrix}Z_i+\Pi_{\mathrm{ALL}}\\=&-Z^{\top}\mathcal{H}Z+\Pi_{\mathrm{ALL}}\end{aligned}$$

where $h_1=\underline{\lambda}_{Q,l}$, $h_2=\underline{\lambda}_{Q,f}$, $h_3=\frac{1}{2}k_{l,c1}\sigma_l(t)\sigma_l^{\top}(t)+\frac{1}{2}k_{l,c2}\vartheta_{2,l}I_{\mathcal{L},l}$, $h_4=(k_{l,c1}+k_{f,c1})\sigma_l(t)\sigma_f^{\top}(t)$, $h_5=\frac{1}{2}k_{f,c1}\sigma_f(t)\sigma_f^{\top}(t)+\frac{1}{2}k_{f,c2}\vartheta_{2,f}I_{\mathcal{L},f}$, $h_6=-I_{\mathcal{L},l}$, $h_7=I_{\mathcal{L},l}-\bar{\lambda}_{R,l}\Sigma_{u_l}I_{\mathcal{L},l}$, $h_8=-I_{\mathcal{L},f}$, $h_9=-I_{\mathcal{L},f}+\bar{\lambda}_{R,f}\Sigma_{u_f}I_{\mathcal{L},f}$, and

$$\begin{aligned}\Pi_{\mathrm{ALL}}=\sum_{\mathbf{i}=f,l}&\left(\frac{1}{2}k_{\mathbf{i},c1}\left(\frac{1}{4}\tilde{W}_{\mathbf{i},a}^{\top}G_{\sigma}\tilde{W}_{\mathbf{i},a}+\xi_{\mathbf{i},H}+\Delta_{\mathbf{i}}\right)^{2}\right.\\&\left.+\frac{1}{2}k_{\mathbf{i},c2}\left(\frac{1}{4}\tilde{W}_{\mathbf{i},a}^{\top}G_{\sigma,k}\tilde{W}_{\mathbf{i},a}+\Delta_{\mathbf{i}}^{k}\right)^{2}+\bar{\lambda}_{R,\mathbf{i}}\Pi_{u_{\mathbf{i}}}\right)\end{aligned}$$

When a suitable positive definite matrix $\mathcal{H}$ is chosen, the closed-loop system state $[x_l^{\top},\epsilon^{\top}]^{\top}$ and the network weight errors $\tilde{W}_{\mathbf{i},c},\tilde{W}_{\mathbf{i},a}$ will end up being UUB when the condition $\|Z\|\geq\sqrt{\Pi_{\mathrm{ALL}}/\lambda_{\min}(\mathcal{H})}$ is satisfied. The proof is completed.

REFERENCES

[1] Z. Jin, A. Liu, W.-A. Zhang, L. Yu, and C.-Y. Su, "A Learning Based Hierarchical Control Framework for Human–Robot Collaboration," *IEEE Transactions on Automation Science and Engineering*, vol. 20, pp. 506–517, Jan. 2023.

[2] J. Zhu, J. Zhu, Z. Wang, S. Guo, and C. Xu, "Hierarchical Decision and Control for Continuous Multitarget Problem: Policy Evaluation With Action Delay," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 464–473, Feb. 2019.

[3] J. Zhou, S. Zhong, and W. Wu, "Hierarchical Motion Learning for Goal-Oriented Movements With Speed–Accuracy Tradeoff of a Musculoskeletal System," *IEEE Transactions on Cybernetics*, vol. 52, pp. 11453–11466, Nov. 2022.

[4] W. Chen, H. Wan, X. Luan, and F. Liu, "Self-triggered control for linear systems based on hierarchical reinforcement learning," *International Journal of Robust and Nonlinear Control*, p. rnc.7452, May 2024.

[5] A. D. Saravanos, Y. Aoyama, H. Zhu, and E. A. Theodorou, "Distributed Differential Dynamic Programming Architectures for Large-Scale Multi-agent Control," *IEEE Transactions on Robotics*, vol. 39, pp. 4387–4407, Dec. 2023.

[6] 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, pp. 33–52, Feb. 2017.

[7] A. Borowczyk, D.-T. Nguyen, A. P.-V. Nguyen, D. Q. Nguyen, D. Saussié, and J. Le Ny, "Autonomous Landing of a Quadcopter on a High-Speed Ground Vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 40, pp. 2378–2385, Sept. 2017.

[8] A. Weiss, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Model Predictive Control for Spacecraft Rendezvous and Docking: Strategies for Handling Constraints and Case Studies," *IEEE Transactions on Control Systems Technology*, vol. 23, pp. 1638–1647, July 2015.

[9] J. Tan, S. Xue, Q. Guan, K. Qu, and H. Cao, "Finite-time safe reinforcement learning control of multi-player nonzero-sum game for quadcopter systems," *Information Sciences*, vol. 712, p. 122117, Sept. 2025.

[10] Z. Ji, Z. Wang, H. Lin, and Z. Wang, "Interconnection topologies for multi-agent coordination under leader–follower framework," *Automatica*, vol. 45, pp. 2857–2863, Dec. 2009.

[11] H. Cai and G. Hu, "Distributed Tracking Control of an Interconnected Leader–Follower Multiagent System," *IEEE Transactions on Automatic Control*, vol. 62, pp. 3494–3501, July 2017.

[12] S. Xue, J. Liu, H. Cao, X. Zheng, H. Li, and J. Zhang, "Leader-Following Formation Tracking of Multiagent Systems Using Adaptive Scaling Mechanism Under Spatial Constraints," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, pp. 1214–1225, Feb. 2024.

[13] J. Tan, S. Xue, Q. Guan, T. Niu, H. Cao, and B. Chen, "Unmanned aerial-ground vehicle finite-time docking control via pursuit-evasion games," *Nonlinear Dynamics*, Mar. 2025.

[14] M. Li, J. Qin, J. Li, Q. Liu, Y. Shi, and Y. Kang, "Game-Based Approximate Optimal Motion Planning for Safe Human-Swarm Interaction," *IEEE Transactions on Cybernetics*, pp. 1–12, 2023.

[15] K. Tong, M. Li, J. Qin, Q. Ma, J. Zhang, and Q. Liu, "Differential Game-Based Control for Nonlinear Human–Robot Interaction System With Unknown Desired Trajectory," *IEEE Transactions on Cybernetics*, pp. 1–11, 2024.

[16] H. Cao, Y. Li, C. Liu, and S. Zhao, "ESO-Based Robust and High-Precision Tracking Control for Aerial Manipulation," *IEEE Transactions on Automation Science and Engineering*, vol. 21, pp. 2139–2155, Apr. 2024.

[17] H. Cao, J. Shen, C. Liu, B. Zhu, and S. Zhao, "Motion Planning for Aerial Pick-and-Place With Geometric Feasibility Constraints," *IEEE Transactions on Automation Science and Engineering*, pp. 1–18, 2024.

[18] Y. Sun, J. Hu, Z. Peng, and B. K. Ghosh, "Hierarchical critic learning optimal control for lower limb exoskeleton robots with prescribed constraints," *International Journal of Robust and Nonlinear Control*, vol. 34, pp. 2162–2183, Feb. 2024.

[19] M. Deng, Z. Li, Y. Kang, C. L. P. Chen, and X. Chu, "A Learning-Based Hierarchical Control Scheme for an Exoskeleton Robot in Human–Robot Cooperative Manipulation," *IEEE Transactions on Cybernetics*, vol. 50, pp. 112–125, Jan. 2020.

[20] R. Chai, H. Niu, J. Carrasco, F. Arvin, H. Yin, and B. Lennox, "Design and Experimental Validation of Deep Reinforcement Learning-Based Fast Trajectory Planning and Control for Mobile Robot in Unknown Environment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, pp. 5778–5792, Apr. 2024.

[21] W. Zhu and M. Hayashibe, "A Hierarchical Deep Reinforcement Learning Framework With High Efficiency and Generalization for Fast and Safe Navigation," *IEEE Transactions on Industrial Electronics*, vol. 70, pp. 4962–4971, May 2023.

[22] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, vol. 62, pp. 3861–3876, Aug. 2017.

[23] W. Ren, J. Li, J. Xiong, and X.-M. Sun, "Vector Control Lyapunov and Barrier Functions for Safe Stabilization of Interconnected Systems," *SIAM Journal on Control and Optimization*, vol. 61, pp. 3209–3233, Oct. 2023.

[24] B. Barros Carlos, A. Franchi, and G. Oriolo, "Towards Safe Human-Quadrotor Interaction: Mixed-Initiative Control via Real-Time NMPC," *IEEE Robotics and Automation Letters*, vol. 6, pp. 7611–7618, Oct. 2021.

[25] K. Zhang, R. Su, H. Zhang, and Y. Tian, "Adaptive Resilient Event-Triggered Control Design of Autonomous Vehicles With an Iterative Single Critic Learning Framework," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 5502–5511, Dec. 2021.

[26] Y. Lv, W. Zhang, J. Zhao, and X. Zhao, "Finite-horizon optimal control for nonlinear multi-input systems with online adaptive integral reinforcement learning," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 802–812, 2025.

[27] J. Tan, J. Wang, S. Xue, H. Cao, H. Li, and Z. Guo, "Human–Machine Shared Stabilization Control Based on Safe Adaptive Dynamic Programming With Bounded Rationality," *International Journal of Robust and Nonlinear Control*, p. rnc.7931, Mar. 2025.
[28] C. Mu, K. Wang, X. Xu, and C. Sun, "Safe Adaptive Dynamic Programming for Multiplayer Systems With Static and Moving No-entry Regions," *IEEE Transactions on Artificial Intelligence*, pp. 1–13, 2023.
[29] P. Deptula, H.-Y. Chen, R. A. Licitra, J. A. Rosenfeld, and W. E. Dixon, "Approximate Optimal Motion Planning to Avoid Unknown Moving Avoidance Regions," *IEEE Transactions on Robotics*, vol. 36, pp. 414–430, Apr. 2020.
[30] Y. Yang, K. G. Vamvoudakis, H. Modares, Y. Yin, and D. C. Wunsch, "Safe Intermittent Reinforcement Learning With Static and Dynamic Event Generators," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, pp. 5441–5455, Dec. 2020.
[31] M. H. Cohen and C. Belta, "Safe exploration in model-based reinforcement learning using control barrier functions," *Automatica*, vol. 147, p. 110684, Jan. 2023.
[32] K. Wang, C. Mu, Z. Ni, and D. Liu, "Safe Reinforcement Learning and Adaptive Optimal Control With Applications to Obstacle Avoidance Problem," *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2023.
[33] K. Zhang, S. Luo, H.-N. Wu, and R. Su, "Data-Driven Tracking Control for Non-Affine Yaw Channel of Helicopter via Off-Policy Reinforcement Learning," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–13, 2025.
[34] J. Tan, S. Xue, H. Li, Z. Guo, H. Cao, and D. Li, "Prescribed Performance Robust Approximate Optimal Tracking Control Via Stackelberg Game," *IEEE Transactions on Automation Science and Engineering*, pp. 1–1, 2025.
[35] Z. Zhang and J. F. Fisac, "Safe Occlusion-aware Autonomous Driving via Game-Theoretic Active Perception," in *Robotics: Science and Systems XVII*, July 2021.
[36] K. Zhang, Z.-X. Zhang, X. P. Xie, and J. D. J. Rubio, "An Unknown Multiplayer Nonzero-Sum Game: Prescribed-Time Dynamic Event-Triggered Control via Adaptive Dynamic Programming," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 8317–8328, 2025.
[37] Z. Zhang, K. Zhang, X. Xie, and V. Stojanovic, "ADP-Based Prescribed-Time Control for Nonlinear Time-Varying Delay Systems With Uncertain Parameters," *IEEE Transactions on Automation Science and Engineering*, pp. 1–11, 2024.
[38] H. Dong, X. Zhao, and B. Luo, "Optimal Tracking Control for Uncertain Nonlinear Systems With Prescribed Performance via Critic-Only ADP," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, pp. 561–573, Jan. 2022.
[39] N. Wang, Y. Gao, and X. Zhang, "Data-Driven Performance-Prescribed Reinforcement Learning Control of an Unmanned Surface Vehicle," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 5456–5467, Dec. 2021.
[40] H. Yang, Q. Hu, H. Dong, X. Zhao, and D. Li, "Optimized Data-Driven Prescribed Performance Attitude Control for Actuator Saturated Spacecraft," *IEEE/ASME Transactions on Mechatronics*, vol. 28, pp. 1616–1626, June 2023.
[41] Y. Tang, Y. Lv, J. Zhao, L. Jian, and L. Li, "Prescribed performance event-triggered optimal control of nonlinear multi-input systems," *Neurocomputing*, vol. 637, p. 130044, July 2025.
[42] H. Yang, H. Dong, and X. Zhao, "ADP-Based Spacecraft Attitude Control Under Actuator Misalignment and Pointing Constraints," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 9, 2022.
[43] H. Dong, X. Zhao, Q. Hu, H. Yang, and P. Qi, "Learning-Based Attitude Tracking Control With High-Performance Parameter Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, pp. 2218–2230, June 2022.
[44] L. Zhang and Y. Chen, "Finite-Time Adaptive Dynamic Programming for Affine-Form Nonlinear Systems," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2023.
[45] R. Kamalapurkar, J. A. Rosenfeld, and W. E. Dixon, "Efficient model-based reinforcement learning for approximate online optimal control," *Automatica*, vol. 74, pp. 247–258, Dec. 2016.

Junkai Tan received the B.E. degree in electrical engineering at the School of Electrical Engineering in Xi'an Jiaotong University, Xi'an, China. He is currently working toward the M.E. degree in electrical engineering at the School of Electrical Engineering, Xi'an Jiaotong University.

His current research interest includes adaptive dynamic programming and inverse reinforcement learning.

Shuangsi Xue (M'24) received the B.E. degree in electrical engineering and automation from Hunan University, Changsha, China, in 2014, and the M.E. and Ph.D. degrees in electrical engineering from Xian Jiaotong University, Xian, China, in 2018 and 2023, respectively. He is currently an Assistant Professor at the School of Electrical Engineering, Xian Jiaotong University.

His current research interest includes adaptive control and data-driven control of networked systems.

Huan Li received the B.E. degree in electrical engineering at the School of Electrical Engineering in Xi'an Jiaotong University, Xi'an, China. She is currently working toward the M.E. degree in electrical engineering at the School of Electrical Engineering, Xi'an Jiaotong University.

Her current research interest includes consensus control of multi-agent systems and sliding-mode control.

Zihang Guo received the B.E. degree in electrical engineering at the School of Electrical Engineering in Xi'an Jiaotong University, Xi'an, China. He is currently working toward the M.E. degree in electrical engineering at the School of Electrical Engineering, Xi'an Jiaotong University.

His current research interest includes neural network and sliding mode-based path planning and tracking methods.

Hui Cao (M'11) received the B.E., M.E., and Ph.D. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2000, 2004, and 2009, respectively.

He is a Professor at the School of Electrical Engineering, Xi'an Jiaotong University. He was a Postdoctoral Research Fellow at the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, from 2014 to 2015. He has authored or coauthored over 30 scientific and technical papers in recent years. His current research interest includes knowledge representation and discovery. Dr. Cao was a recipient of the Second Prize of National Technical Invention Award.

Badong Chen (SM'13) received the Ph.D. degree in Computer Science and Technology from Tsinghua University, Beijing, China, in 2008. He is currently a professor with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China. His research interests are in signal processing, machine learning, artificial intelligence and robotics. He has authored or coauthored over 200 articles in various journals and conference proceedings (with 18000+ citations in Google Scholar), and has won the 2022 Outstanding Paper Award of IEEE Transactions on Cognitive and Developmental Systems. Dr. Chen serves as a Member of the Machine Learning for Signal Processing Technical Committee of the IEEE Signal Processing Society, and serves (or has served) as an Associate Editor for several international journals including IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Cognitive and Developmental Systems, IEEE Transactions on Circuits and Systems for Video Technology, Neural Networks and Journal of The Franklin Institute. He has served as a PC or SPC Member for prestigious conferences including UAI, IJCAI and AAAI, and served as a General Co-Chair of 2022 IEEE International Workshop on Machine Learning for Signal Processing.