Neural Computing and Applications
https://doi.org/10.1007/s00521-024-10639-8

S.I.: REINFORCEMENT LEARNING AND ADAPTIVE DYNAMIC PROGRAMMING FOR CONTROL SYSTEMS: THEORY AND APPLICATIONS

Safe adaptive control for multi-agent systems under uncertain dynamic environments: a robust barrier-certified learning approach

Shawon Dey[1] · Hao Xu[1]

Received: 7 October 2023 / Accepted: 7 October 2024

Abstract
In this study, a decentralized safe learning-based adaptive control is developed for multi-agent systems (MAS) operating in uncertain environments. Ensuring the safety of MAS in uncertain environments becomes a challenging task, particularly when the dynamics of the individual agent are unknown and their accurate state information is unavailable to the local agent. Due to the fact that the safe set of the local agent depends on the other agents' states, the uncertainty of these external systems leads to an uncertain safety set. As a consequence, the safe control design of the local agent system in a multi-agent setting under environmental uncertainties becomes intractable. To address this challenge and ensure the safety of the local agent, a neural network (NN)-based adaptive observer is designed to estimate the state of the unknown external agents in the multi-agent environment. Based on the state estimation of external agents, an adaptive interplay control barrier function (AI-CBF) is formulated. The AI-CBF is designed by considering both the local agent's state and the estimated states of external agents. Notably, the limitation of forward invariance for the approximated safe set without guaranteeing the same for the actual safe set is acknowledged in AI-CBF design. The AI-CBF incorporates the bounds on state estimation errors of external agents to guarantee the strict safety requirements of the local agent. Based on the safety constraint enforced by the AI-CBF, a control framework is formulated using a quadratic programming (QP) method that integrates the safety and stability of the system. In addition, a stability analysis based on Lyapunov theory is performed to demonstrate the convergence of the neural network-based adaptive observer as well as the closed-loop stability of the overall system. Eventually, experimental validation and comparison study confirm the effectiveness of the developed approach that can ensure multi-agent system safety under challenging conditions in a decentralized manner.

Keywords Safe learning · Neural network · Control barrier function · Adaptive control

1 Introduction

Ensuring safety stands as a paramount consideration in real-world control design especially for automated systems such as unmanned aerial vehicles (UAVs) [1, 2], robots [3], autonomous ground vehicles [4], and so on. With the potential risks and hazards associated with these technologies, it becomes crucial to emphasize the development of robust safety measures. Thoroughly addressing safety considerations during the control design process allows for the cultivation of trust, reduction of accidents, and protection of human lives and valuable assets in practical settings [5, 6]. Since the safety concept has been introduced to real-time system design in [7], a significant body of research has been conducted in the fields of safe control and safe learning systems. Notably, a considerable amount of research has been dedicated to certified learning-based control approaches with constraint-set certification in the presence of dynamic uncertainty [8–10]. This certification process relies on the introduced robust positive control invariant safe sets. Two distinct approaches for

✉ Shawon Dey
sdey@unr.edu

Hao Xu
haoxu@unr.edu

[1] University of Nevada, Reno, 1664 N Virginia St, Reno, NV 89557, USA

Published online: 09 December 2024

state constraint-set certification are the control barrier function (CBF) method [10–12] and the Hamilton-Jacobi reachability [13–15] analysis. Although both methods aim to ensure safety, they differ in their relevant computational requirements [16]. The CBF-based approaches focus on guaranteeing safety without explicitly determining the reachability set, which is computationally intensive. Instead, it utilizes a control barrier function that enforces constraints on the behavior of the system. By developing a controller that satisfies the requirement of barrier function, safety can be assured without the need for extensive reachability analysis. This CBF-based approach has proven effective in diverse applications, such as collision avoidance in multi-agent systems [17, 18] and car racing environments [19], safe lane change maneuvers [20], adaptive cruise control [21], and the safe control of robots [22]. Most of these existing techniques rely on precise agent state information to guarantee safety. However, implementing CBF to ensure the forward invariance of a safe set becomes more complex and challenging since uncertainty widely exists in the environment. Numerous studies have focused on addressing the issue of system dynamic uncertainty in CBF-based safe controls. In a recent paper [23], the authors incorporated a probabilistic control Lyapunov function (CLF) and CBF design to handle uncertainties in the dynamic model and real-time safety constraints. To mitigate the effects of unknown structured systems, [23] introduced an element-wise parameter identification law. Additionally, [24] presented a safety control scheme that utilized CBF and addressed uncertainty by dismissing the uncertainty associated with the unknown structured system. Furthermore, a reinforcement learning-based technique was designed in [25] to learn and address model uncertainties present in both CLF and CBF. However, in a multi-agent system, when a local agent shares its environment with other agents, achieving safety for the local agent becomes challenging due to the environment uncertainty associated with the safe set. In practice, agents like robots, UAVs, and automobiles often operate within complex and uncertain environments. For example, consider a UAV flying in airspace alongside other aircraft. The uncertainty of shared airspace presents challenges for ensuring the safety of the UAV and other aircraft. Similarly, a self-driving car navigating through a road system with other cars poses safety concerns due to the shared road environment. The shared environment introduces complexities for ensuring the safety of individual agents, requiring a safe control mechanism to avoid collision and promote safe interactions. In a recent study by Marvi et al. [16], a safe controller is designed for system maneuvering in shared environments. They designed a control strategy to ensure the forward invariance criterion for the intersection of the actual safety set and the approximated safety set. Please note that the actual safe set in this study is formed by using the states of external agents. However, in practice, the external agents' states cannot be measured directly and only the system's input and output are measurable. Lack of full state information on external systems, achieving the actual and approximated safe sets' intersection is very difficult and even impossible. This assumption of the actual safe set may violate the safety of the local agent when the exact full state information of the external agents is not available. In this paper, a novel observer is designed to estimate external agents' states and further used for generating safety sets for local agents. However, the external agents' systems are unknown and uncertain. So the known model-based observer design cannot be directly implemented in this system. This problem has been addressed in [26, 27] which combined observer design with neural network-based system identification.

These studies used a static approximation of the gradient by assuming the system state to be constant and remain unchanged over time. However, this presumption lacks practicality and efficiency in practice. In the real world, external agents' states are usually time-varying and the local agent has no authority over these external agents' actions, i.e., control inputs. To address these issues, an online neural network (NN)-based adaptive observer has been developed to learn external agents' unknown dynamics as well as their states. A modified objective function for the gradient descent weight update mechanism is designed while addressing the limitation of static gradient approximation. Additionally, a mathematical demonstration of the stability of the observer-based identification of the external system is provided. Then, the existing control barrier function can be reformulated using the local agents' system state and observed external system states. Using the reformulated adaptive interplay control barrier function (AI-CBF), the safety criteria are designed as a function based on the local agents' own state and also the state of the external agents obtained from the observer. As the actual safe set remains unknown at the initial learning stage, and the local agent relies solely on the approximated safe set, an error bound is introduced for the estimated external agent state. This error bound is then incorporated in the AI-CBF to avoid the violation of strict safety while the local system is learning the external agents' dynamics. Later, the estimated external agents' states will converge to actual states along with the convergence of learning. Moreover, the approximated safe set can gradually converge toward the actual safe set. This developed algorithm differs from the previously discussed algorithm by offering an AI-CBF-based robust decentralized safe control design, please see Fig. 1. It addresses the uncertainty of external agents within the safe set by introducing a novel neural network-based adaptive observer design. A comparative study of the proposed AI-CBF

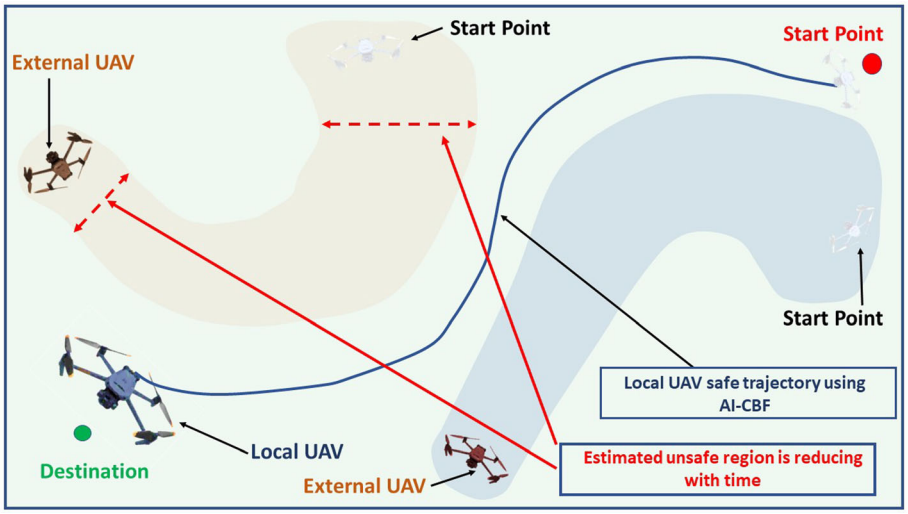


Fig. 1 An illustration of AI-CBF-based safe maneuver of a UAV sharing an environment with other external UAVs in uncertain environments. The states of external UAVs are estimated using a multi-neural network-based adaptive observer

approach has been conducted, including scenarios without CBF and with other existing CBF approaches [14]. Our approach introduces the following contributions to the field:

- A decentralized safe control design based on AI-CBF is developed for individual agents navigating in a multi-agent shared environment characterized by the presence of unknown external agent dynamics. The developed control design incorporates both the local agent and observer-based external agent estimated states. The developed method ensures robust safety, even in cases where the accuracy of the external agents' state information is compromised.
- A novel multi-NN-based adaptive observer is designed to estimate the state of the unknown external agents and further ensure the local system safety.
- The stability of the NN-based observer for external multi-agent systems and also the closed-loop stability of the local agent is guaranteed through Lyapunov stability analysis. Additionally, a comparative study against other existing algorithms has been conducted to validate the effectiveness of the developed algorithm.

The structure of the paper is given next. In Sect. 2, the decentralized safe control problem is formulated. Then in Sect. 3, the multi-neural network-based observer and AI-CBF are designed. Moving on to Sect. 4, the control framework is presented. Then the simulation results are demonstrated in Sect. 5, followed by the conclusion in Sect. 6.

2 Problem formulation and background

Consider a nonlinear affine system of local agent $\mathcal{A}$ in multi-agent systems given by the following dynamics

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \tag{1}$$

where $x(t) = [x_1(t), x_2(t), ..., x_n(t)]^{\mathrm{T}} \in \mathbb{R}^n$ represents the state and $u(t) = [u_1(t), u_2(t), ..., u_m(t)]^{\mathrm{T}} \in \mathbb{R}^m$ represents the control input of the local agent. Also, $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ demonstrates the intrinsic dynamics of the system. Moreover, recall to [28], the function $f(.)$ satisfies the condition $f(0) = 0$. Also, $f(x) + g(x)u$ is bounded by a Lipschitz constant. The objective of the agent $\mathcal{A}$ is to reach a predefined destination safely by avoiding collision in a multi-agent environment. Now, the dynamic of an external agent i can be defined as:

$$\dot{z}_i(t) = f_a(z_i(t), u_i); \quad y_i(t) = Cz_i(t) \tag{2}$$

where $z_i(t) = [z_{i,1}(t), z_{i,2}(t), ..., z_{i,n}(t)]^{\mathrm{T}} \in \mathbb{R}^n$ represents the state, $y_i(t) = [y_{i,1}, y_{i,2}, ..., y_{i,n}]^{\mathrm{T}} \in \mathbb{R}^p$ represents the output and u_i denotes the control input of the external agent i. Also, f_a is an unknown nonlinear function that captures the effect of an external agent. It is assumed that the external multi-agent systems are observable. Now, the dynamic of the external agent can be rewritten as:

$$\dot{z}_i(t) = Az_i(t) + F(z_i(t), u_i); \quad y_i(t) = Cz_i(t) \tag{3}$$

with $F(z_i(t), u_i) = f_a(z_i(t), u_i) - Az_i(t)$ and A represents Hurwitz matrix. Also, x_d represents the desired destination that the agent is required to achieve. Then the error can be defined as $e = x - x_d$ with the error dynamic given as:

$$\dot{e} = f_a(e) + g_a(e)u \tag{4}$$

where $f_a(e) = f(e + x_d)$ and $g_a(e) = g(e + x_d)$. The agent's safety cannot be solely determined by its own control inputs and characteristics but also relies on the interplay with other external systems in an uncertain multi-agent environment. Thus, to maintain the safety of local agents, it is needed to consider both local agent's behavior

as well as interaction with other agents. However, in real-world scenarios, the states of the external agents are not available while only the outputs of the external systems are measurable. Moreover, the dynamics of the external systems are also unknown. Hence, the objective of this research is to guarantee the safety and stability of the agent $\mathcal{A}$ in a multi-agent setting where the external agent states are not directly measurable and the dynamics of these external agents are unknown. The objectives are listed as follows:

1. Design a feedback controller for the agent $\mathcal{A}$ in a decentralized manner, which guarantees the trajectory of the agent stays inside a safe set in the multi-agent environment while achieving a desired destination x_d that satisfies

$$h_i(x(t), z_i(t)) \geq 0 \quad \text{for} \quad t \geq 0 \tag{5}$$

with $h_i(x(t), z_i(t))$ being a continuously differentiable function. Also, the safe set for the agent $\mathcal{A}$ is defined as the intersection of the sets associated with all other external agents, i.e.,

$$S = S_1 \cap S_2 \ldots \cap S_N \tag{6}$$

with N being the external agents' number in the environment.
2. Design a multiple NN-based adaptive observer to estimate the state of all the other external agents to guarantee the safety of local agent $\mathcal{A}$.
3. Guaranteeing the stability of local agent $\mathcal{A}$ and the observer-based external agent's state estimation.

Before proceeding with the developed algorithm, it is essential to provide a concise introduction to the control barrier function (CBF).

Control Barrier Function: The safety framework [11] is characterized by an invariant set, known as the actual safe set. This set, denoted as $\mathcal{S}$, can be derived as a super-level set of a continuously differentiable smooth function $h : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$. Then, for the dynamical system provided in Eq. (1), the safe set can be defined as:

$$\begin{aligned} \mathcal{S} &= \{x \in \mathcal{X} \subset \mathbb{R}^n : h(x) \geq 0\}, \\ \partial\mathcal{S} &= \{x \in \mathcal{X} \subset \mathbb{R}^n : h(x) = 0\}, \\ \text{Int}(\mathcal{S}) &= \{x \in \mathcal{X} \subset \mathbb{R}^n : h(x) > 0\}. \end{aligned}$$

Here, $\partial\mathcal{S}$ represents the boundary of the safe set and $\text{Int}(\mathcal{S})$ denotes the interior of the safe set $\mathcal{S}$. Now, h can be referred to as a control barrier function if there exists an extended class $\mathcal{K}_\alpha$ function α such that the following condition is satisfied for the given dynamical system:

$$\sup_{u \in U} [L_f h(x) + L_g h(x) u] \geq -\alpha(h(x)) \tag{7}$$

for all $x \in \mathcal{X}$. where $L_f = \frac{\partial h}{\partial x} f(x)$ and $L_g = \frac{\partial h}{\partial x} g(x)$ represents the Lie derivatives of $h(x)$ along the intrinsic functions f and g, respectively. Then, the extended $\mathcal{K}_\alpha$ function can be defined as follows:

Definition 1 A function $\alpha : \mathbb{R} \to \mathbb{R}$ is known as an extended class $\mathcal{K}_\alpha$ function if the function is strictly increasing and $\alpha(0) = 0$. Please see [11] for the definition.

The above condition can also be defined for discrete cases [29]. Now, the control inputs set that satisfies Eq. (7) and renders the set $\mathcal{S}$ safe can be defined as:

$$K_{\text{cbf}}(x) = \{u \in U : L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0\}. \tag{8}$$

The proof of this result relies on Nagumo's theorem [30] which necessitates the fulfillment of a regularity condition [11]. This condition states that the function h, serving as a control barrier on the set $\mathcal{X}$, must satisfy $\frac{\partial h}{\partial x}(x) \neq 0$, $\forall x \in \partial\mathcal{S}$. According to Nagumo's theorem, to establish the existence of a set to be positive invariant, the dynamics $\dot{x}$ should either point inward toward the boundary or be tangent to it [16].

3 Neural network-based adaptive observer with adaptive interplay control barrier function design

In this paper, an adaptation has been introduced to the control barrier function (CBF) within the context of a shared environment comprising multiple agents. These agents possess unknown dynamics, and their states are not accessible for direct online measurement. The barrier function is now reformulated as a function of x and z_i, where i is the index for the external agent. Specifically, a neural network-based adaptive observer has been designed to estimate the states of the external agents first. Then, employing the estimated states of external agents from the observer, the CBF is reformulated to AI-CBF using both the local agent $\mathcal{A}$th state x and estimated state $\hat{z}_i$ of any external agent i. Lastly, to integrate safety and stability by combining the Lyapunov function and CBF, a feedback controller is constructed using quadratic programming (QP).

3.1 Multiple NN-based adaptive observer design for external agents

In this part of the paper, a multiple NN-based observer is designed, as shown in Fig. 2, to estimate the external agents' state. Here, neural networks are used to identify the external agents' unknown dynamics, and observers are

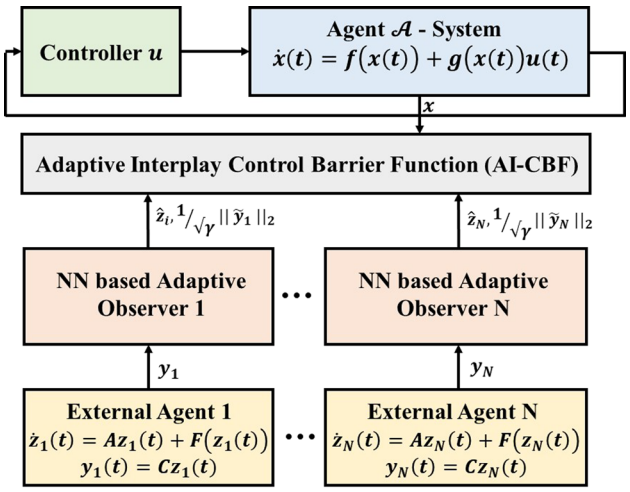


Fig. 2 The structure of multi-NN adaptive observer-based safe control scheme

used along with NNs to estimate the states of these agents. Now, the model of the observer can be described as:

$$\begin{aligned} \dot{\hat{z}}_i(t) &= A\hat{z}_i(t) + F(\hat{z}_i(t), u_i) + H(y_i(t) - C\hat{z}_i(t)) \\ \hat{y}_i(t) &= C\hat{z}_i(t) \end{aligned} \tag{9}$$

with $\hat{z}_i$ and $\hat{y}_i$ being the state and the output of the observer for external agent i. The selection of the observer gain $H \in \mathbb{R}^{n\times p}$ ensures that the matrix $A - HC$ is Hurwitz. Please note that the observability of pair (C, A) depends on the selection of the matrix A. By selecting A properly, the existence of the gain H is ensured. According to universal approximation theory [31], the unknown function of the external agent is represented as:

$$F(z_i(t), u_i) = W_i^{\mathrm{T}}\phi(z_i) + \varepsilon_f \tag{10}$$

with $W \in \mathbb{R}^{l\times n}$ being the neural network ideal weight and l is the neuron numbers in the hidden layer. Please note that the activation function is bounded as $\|\phi(z_i)\| \le \phi_M$ and the ideal weight is bounded as $\|W_i\| \le W_M$. Now, the function of the external agent is approximated as

$$\hat{F}(\hat{z}_i(t), u_i) = \hat{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) \tag{11}$$

where $\hat{W} \in \mathbb{R}^{l\times n}$ is the neural network estimated weight. Now, replacing the approximation of the unknown function, the observer model is represented as

$$\begin{aligned} \dot{\hat{z}}_i(t) &= A\hat{z}_i(t) + \hat{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + H(y_i(t) - C\hat{z}_i(t)) \\ \hat{y}_i(t) &= C\hat{z}_i(t). \end{aligned} \tag{12}$$

Then, the state and output estimation error of the observer is defined as $\tilde{z}_i = z_i - \hat{z}_i$ and $\tilde{y}_i = y_i - \hat{y}_i$. Next, the state and output error dynamics can be evaluated using Eqs. (12) and (10) as

$$\begin{aligned} \dot{\tilde{z}}_i(t) &= \dot{z}_i(t) - \dot{\hat{z}}_i(t) \\ &= Az_i(t) + W_i^{\mathrm{T}}\phi(z_i) + \varepsilon_f - A\hat{z}_i(t) - \hat{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) \\ &\quad - H(y_i(t) - C\hat{z}_i(t)) \\ &= A\tilde{z}_i(t) + W_i^{\mathrm{T}}\phi(z_i) + \varepsilon_f - \hat{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) - HC\tilde{z}_i(t) \\ &= (A - HC)\tilde{z}_i(t) + W_i^{\mathrm{T}}\phi(z_i) - \hat{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + \varepsilon_f. \end{aligned} \tag{13}$$

And,

$$\tilde{y}_i(t) = y_i(t) - \hat{y}_i(t) = C\tilde{z}_i(t). \tag{14}$$

Then, the estimation error of the neural network weight is $\tilde{W}_i = W_i - \hat{W}_i$ and the activation function approximation error is $\tilde{\phi}(\tilde{z}_i) = \phi(z_i) - \hat{\phi}(\hat{z}_i)$. Also, $A - HC = A_o$. Equation (13) is now written using the weight and activation function approximation errors:

$$\begin{aligned} \dot{\tilde{z}}_i(t) &= A_o\tilde{z}_i(t) + W_i^{\mathrm{T}}\phi(z_i) - W_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + W_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) - \hat{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + \varepsilon_f \\ &= A_o\tilde{z}_i(t) + W_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) + \tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + \varepsilon_f. \end{aligned} \tag{15}$$

Assumption 1 The external system activation function approximation error is Lipschitz continuous. It implies the existence of the Lipschitz function, denoted as L_ϕ, that satisfies the inequality $\|\tilde{\phi}(\tilde{z}_i)\| \le L_\phi\|\tilde{z}_i\|$.

Once the neural network structure has been established, it is essential to develop an appropriate learning algorithm to update the neural network weights. In this paper, a modified weight update law has been designed to guarantee the stability of the NN-based observer. Taking the first derivative of the approximated output of the external agent in Eq. (14)

$$\begin{aligned} \dot{\tilde{y}}_i(t) &= C\dot{\tilde{z}}_i(t) = C[A_o\tilde{z}_i(t) + W_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) + \tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + \varepsilon_f] \\ &= CA_o\tilde{z}_i(t) + CW_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) + C\tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + C\varepsilon_f \\ &= CA_o(C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}\tilde{y}_i + CW_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) + C\tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + C\varepsilon_f \end{aligned} \tag{16}$$

where $(C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}} = C^{+}$ is the pseudoinverse of the matrix C. After substituting the pseudoinverse, Eq. (16) is now written as follows:

$$\dot{\tilde{y}}_i(t) - CA_oC^+\tilde{y}_i = CW_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) + C\tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + C\varepsilon_f. \quad (17)$$

Now, the objective function can be defined as

$$J_i = \frac{1}{2}(\dot{\tilde{y}}_i(t) - CA_oC^+\tilde{y}_i)^{\mathrm{T}}(\dot{\tilde{y}}_i(t) - CA_oC^+\tilde{y}_i). \quad (18)$$

The gradient descent-based update law is defined as follows:

$$\begin{aligned}\dot{\hat{W}}_i &= -\alpha\frac{\partial J_i}{\partial \hat{W}_i} = -\alpha(\dot{\tilde{y}}_i(t) - CA_oC^+\tilde{y}_i)^{\mathrm{T}}\frac{\partial(\dot{\tilde{y}}_i(t) - CA_oC^+\tilde{y}_i)}{\partial \hat{W}_i} \\ &= \alpha C\hat{\phi}(\hat{z}_i)\left[CW_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) + C\tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + C\varepsilon_f\right]^{\mathrm{T}}\end{aligned} \quad (19)$$

where α represents the learning rate of the neural network. The weight approximation error dynamic can be defined as:

$$\dot{\tilde{W}}_i = \dot{W}_i - \dot{\hat{W}}_i = -\alpha C\hat{\phi}(\hat{z}_i)\left[CW_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) + C\tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + C\varepsilon_f\right]^{\mathrm{T}}. \quad (20)$$

Theorem 1 *The weight update law is given by Eq. (19), with positive constants representing the learning rate α for the neural network. When the neural network reconstruction error is present, the weight approximation error $\tilde{W}_i$ and state approximation error $\tilde{z}_i$ are uniformly ultimately bounded (UUB). Also, in the absence of reconstruction error [32], the weight approximation error converges to zero.*

Proof Consider the Lyapunov candidate function as follows

$$\mathcal{V}_s = \frac{1}{2}\tilde{z}_i^{\mathrm{T}}P\tilde{z}_i + \frac{1}{2}\mathrm{tr}\{\tilde{W}_i^{\mathrm{T}}\tilde{W}_i\} \quad (21)$$

where P represents a positive definite matrix. Also, P satisfies the following condition

$$A_o^{\mathrm{T}}P + PA_o = -Q \quad (22)$$

where Q represents a positive definite matrix and A_o is a Hurwitz matrix. Also, please note that the norm of a vector $a \in \mathbb{R}^n$ is denoted as $\|a\| = \sqrt{a^Ta}$. Also, the Frobenius norm [26] of a matrix $M \in \mathbb{R}^{m\times n}$ is represented as $\|M\|_F^2 = \mathrm{tr}(M^TM)$. Next, we have omitted the subscript F from the matrix norm to simplify the notation throughout the rest of the paper. Now, the first derivative of Eq. (21) can be written as

$$\dot{\mathcal{V}}_s = \frac{1}{2}\dot{\tilde{z}}_i^{\mathrm{T}}P\tilde{z}_i + \frac{1}{2}\tilde{z}_i^{\mathrm{T}}P\dot{\tilde{z}}_i + \mathrm{tr}\{\tilde{W}_i^{\mathrm{T}}\dot{\tilde{W}}_i\}. \quad (23)$$

Using Eqs. (15) and (20), Eq. (23) can be rewritten as:

$$\begin{aligned}\dot{\mathcal{V}}_s &= \frac{1}{2}[A_o\tilde{z}_i(t) + W_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) + \tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + \varepsilon_f]^{\mathrm{T}}P\tilde{z}_i \\ &\quad + \frac{1}{2}\tilde{z}_i^{\mathrm{T}}P[A_o\tilde{z}_i(t) + W_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) \\ &\quad + \tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + \varepsilon_f] + \mathrm{tr}\{\tilde{W}_i^{\mathrm{T}}(-\alpha C\hat{\phi}(\hat{z}_i)[CW_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) \\ &\quad + C\tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + C\varepsilon_f]^{\mathrm{T}})\} \\ &= \frac{1}{2}\tilde{z}_i^{\mathrm{T}}P\tilde{z}_iA_o^{\mathrm{T}} + \frac{1}{2}\tilde{\phi}^{\mathrm{T}}(\tilde{z}_i)P\tilde{z}_iW_i \\ &\quad + \frac{1}{2}\hat{\phi}^{\mathrm{T}}(z_i)P\tilde{z}_i\tilde{W}_i + \frac{1}{2}\varepsilon_f^{\mathrm{T}}P\tilde{z}_i + \frac{1}{2}\tilde{z}_i^{\mathrm{T}}PA_o\tilde{z}_i + \frac{1}{2}\tilde{z}_i^{\mathrm{T}}P \\ &W_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) + \frac{1}{2}\tilde{z}_i^{\mathrm{T}}P\tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + \frac{1}{2}\tilde{z}_i^{\mathrm{T}}P\varepsilon_f \\ &\quad + \mathrm{tr}\{\tilde{W}_i^{\mathrm{T}}[-\alpha C^TC\hat{\phi}(\hat{z}_i)\tilde{\phi}^{\mathrm{T}}(\tilde{z}_i)W_i \\ &\quad - \alpha C^TC\|\hat{\phi}(\hat{z}_i)\|^2\tilde{W}_i - \alpha C^TC\hat{\phi}(\hat{z}_i)\varepsilon_f^{\mathrm{T}}]\} \\ &= \frac{1}{2}(A_o^{\mathrm{T}}P + PA_o)\tilde{z}_i^{\mathrm{T}}\tilde{z}_i + P\tilde{z}_iW_i^{\mathrm{T}}\tilde{\phi}(\tilde{z}_i) \\ &\quad + P\tilde{z}_i\tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i) + \varepsilon_f^TP\tilde{z}_i - \mathrm{tr}\{\alpha C^TC\hat{\phi}(\hat{z}_i) \\ &\quad \tilde{\phi}^{\mathrm{T}}(\tilde{z}_i)\tilde{W}_i^{\mathrm{T}}W_i\} - \mathrm{tr}\{\alpha C^TC\hat{\phi}^T(\hat{z}_i)\hat{\phi}(\hat{z}_i)\tilde{W}_i^{\mathrm{T}}\tilde{W}_i\} \\ &\quad - \mathrm{tr}\{\alpha C^TC\tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i)\varepsilon_f^{\mathrm{T}}\}.\end{aligned} \quad (24)$$

Consider the bounds of activation function and ideal weights, given by $\|\phi(z_i)\| \le \phi_M$ and $\|W_i\| \le W_M$ and also using Eq. (22) and *Assumption 1*, Eq. (24) can be written as

$$\begin{aligned}\dot{\mathcal{V}}_s &\le -\frac{1}{2}\tilde{z}_i^{\mathrm{T}}Q\tilde{z}_i + \|P\|W_iL_\phi\|\tilde{z}_i\|\|\tilde{z}_i\| + \|P\|\hat{\phi}(\hat{z}_i)\tilde{W}_i^{\mathrm{T}}\tilde{z}_i \\ &\quad + \|P\|\tilde{z}_i\varepsilon_f - \mathrm{tr}\{\alpha C^TC\hat{\phi}(\hat{z}_i)\tilde{\phi}^{\mathrm{T}}(\tilde{z}_i)\tilde{W}_i^{\mathrm{T}}W_i\} \\ &\quad - \alpha C^TC\|\hat{\phi}(\hat{z}_i)\|^2\|\tilde{W}_i\|^2 - \mathrm{tr}\{\alpha C^TC\tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i)\varepsilon_f^{\mathrm{T}}\}. \\ &\le -\frac{1}{2}\tilde{z}_i^{\mathrm{T}}Q\tilde{z}_i + \|P\|W_iL_\phi\|\tilde{z}_i\|^2 + \|P\|\hat{\phi}(\hat{z}_i)\tilde{W}_i^{\mathrm{T}}\tilde{z}_i \\ &\quad + \|P\|\tilde{z}_i\varepsilon_f - \mathrm{tr}\{\alpha C^TC\hat{\phi}(\hat{z}_i) \\ &\tilde{\phi}^{\mathrm{T}}(\tilde{z}_i)\tilde{W}_i^{\mathrm{T}}W_i\} - \alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2\|\tilde{W}_i\|^2 \\ &\quad - \mathrm{tr}\{\alpha C^TC\tilde{W}_i^{\mathrm{T}}\hat{\phi}(\hat{z}_i)\varepsilon_f^{\mathrm{T}}\}.\end{aligned} \quad (25)$$

Using Young's inequality [33], Eq. (25) is written as

$$
\begin{aligned}
\dot{\mathcal{V}}_s \leq & -\frac{1}{2}\lambda_{\min}(Q)\|\tilde{z}_i\|^2 + \|P\|W_M L_\phi\|\tilde{z}_i\|^2 \\
& + \frac{1}{\alpha}\|P\|\hat{\phi}(\hat{z}_i)\alpha\tilde{W}_i^{\mathrm{T}}\tilde{z}_i + \frac{1}{2}\|p\|^2\|\tilde{z}_i\|^2 + \frac{1}{2} \\
& \|\varepsilon_f\|^2 - \alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2\|\tilde{W}_i\|^2 \\
& + \frac{1}{2}\alpha^2(\|C\|^2)^2\|\hat{\phi}(\hat{z}_i)\|^2\|\tilde{W}_i\|^2 + \frac{1}{2}W_M^2 L_\phi^2\|\tilde{z}_i\|^2 \\
& + \frac{1}{2}\alpha^2(\|C\|^2)^2\|\hat{\phi}(\hat{z}_i)\|^2\|\tilde{W}_i\|^2 + \frac{1}{2}\|\varepsilon_f\|^2 \\
\leq & -\frac{1}{2}\lambda_{\min}(Q)\|\tilde{z}_i\|^2 + \|P\|W_M L_\phi\|\tilde{z}_i\|^2 \\
& + \frac{1}{2\alpha^2}\|P\|^2\|\tilde{z}_i\|^2 + \frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2\|\tilde{W}_i\|^2 \\
& + \frac{1}{2}\|p\|^2\|\tilde{z}_i\|^2 - \alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2\|\tilde{W}_i\|^2 \\
& + \alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2\|\tilde{W}_i\|^2 + \frac{1}{2}W_M^2 L_\phi^2\|\tilde{z}_i\|^2 + \|\varepsilon_f\|^2 \\
\leq & -\big[\frac{1}{2}\lambda_{\min}(Q) - \|P\|W_M L_\phi - \frac{1}{2\alpha^2}\|P\|^2 - \frac{1}{2}\|p\|^2 \\
& - \frac{1}{2}W_M^2 L_\phi^2\big]\|\tilde{z}_i\|^2 - \big[\alpha\|C\|^2 \\
& \|\hat{\phi}(\hat{z}_i)\|^2 - \alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2 \\
& - \frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2\big]\|\tilde{W}_i\|^2 + \|\varepsilon_f\|^2
\end{aligned}
\tag{26}
$$

where $\lambda_{\min}(Q)$ represents the minimum eigenvalue of Q. Let,

$$
\kappa_{i,z} = \frac{1}{2}\lambda_{\min}(Q) - \|P\|W_M L_\phi - \frac{1}{2\alpha^2}\|P\|^2 - \frac{1}{2}\|P\|^2 - \frac{1}{2}W_M^2 L_\phi^2. \tag{27}
$$

$$
\kappa_{i,W} = \alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2 - \alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2 - \frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2. \tag{28}
$$

Equation (26) can be rewritten as

$$
\dot{\mathcal{V}}_s \leq -\kappa_{i,z}\|\tilde{z}_i\|^2 - \kappa_{i,W}\|\tilde{W}_i\|^2 + \|\varepsilon_f\|^2. \tag{29}
$$

The first derivative of the Lyapunov function $\dot{\mathcal{V}}_s$ is less than zero outside a compact set if

$$
\|\tilde{z}_i\| > \sqrt{\frac{1}{\kappa_{i,z}}\|\varepsilon_f\|^2} = B_z; \quad \|\tilde{W}_i\| > \sqrt{\frac{1}{\kappa_{i,W}}\|\varepsilon_f\|^2} = B_w. \tag{30}
$$

Here, the condition given in Eq. (30) on $\|\tilde{z}\|$ and $\|\tilde{W}_i\|$ ensures the negative semi-definiteness of $\dot{\mathcal{V}}_s$. Additionally, the minimum eigenvalue $\lambda_{\min}(Q)$ and learning α are lower bounded by $\lambda_{\min}(Q) > \lambda_l$ and α_l, respectively, to guarantee the negative semi-definiteness of $\dot{\mathcal{V}}_s$. This provides an assurance of the stability and convergence of the neural network.

Remark 1 Considering the second term in Eq. (29) is negative [26], $\dot{\mathcal{V}}_s$ is negative definite outside the ball with radius B_z, defined as $\mathcal{Z} = \{\tilde{z}_i \mid \|\tilde{z}_i\| > B_z\}$, ensuring that $\tilde{z}_i$ is uniformly ultimately bounded (UUB). The region within the ball is attractive because an increase in $\dot{\mathcal{V}}_s$ for the values of $\|\tilde{z}_i\|$ leads to an increase $\mathcal{V}_s$ and $\tilde{z}$, pushing $\tilde{z}_i$ outside the ball $\mathcal{Z}$. In this outer region, $\dot{\mathcal{V}}_s$ is negative semi-definite, which in turn reduces $\mathcal{V}_s$ and $\tilde{z}_i$. This ensures the UUB of $\tilde{z}_i$. Additionally, since the activation function $\hat{\phi}(\hat{z}_i)$ is bounded by ϕ_{em} and both C and $\tilde{z}_i$ are bounded, then the boundedness of $\tilde{W}_i$ is also ensured.

Remark 2 If the neural network reconstruction error is ignored, then the weight and state estimation error converges to zero as time progresses.

3.2 AI-CBF design for observer-based state estimation of external agents

The characterization of the safety framework necessitates the safety set positive invariance property, which needs to be carefully formulated to guarantee the safety of the local agent $\mathcal{A}$ in a multi-agent environment, please see Fig. 3. While the local agent is operating within a shared environment with multi-agents, the safe set is defined as the intersection of all the sets associated with the different external systems present in the shared environment. This approach ensures that the local agent remains within the boundaries of safety, accounting for the various dynamics and behaviors exhibited by the external agents. However, as stated earlier, only the output of the external agents states are available and the dynamics as well as full state information of the external agents are unknown. Since there is no accurate state information available for the external agents, the actual safe set of agent $\mathcal{A}$ (local agent) is not available. Therefore, local agent $\mathcal{A}$ depends on the estimated safe set for safe action on the environment. In this regard, the control barrier function is reformulated to incorporate the state of agent $\mathcal{A}$ and external agents. Besides that, a state approximation error bound is considered for the worst-case scenario to ensure strict safety even if the actual safe set is not available to the local agent. In the initial stages of the NN-based observer's training, the estimation error $\tilde{z}$ tends to be relatively large. Consequently, this leads to a substantial bound on the approximation error, resulting in a larger unsafe region for the local agent. However, while the NN-based observer is well-trained, the approximation error decreases significantly. Consequently, agents have more flexibility and a larger safe maneuvering space as the unsafe region diminishes in size. Now, let the output measurement error $\tilde{y}_i$ belong to a sector [34] that can be defined as

$$
\gamma\|\tilde{z}_i\|^2 \leq \|\tilde{y}_i\|^2 \leq \beta\|\tilde{z}_i\|^2. \tag{31}
$$

Here, $\tilde{z}_i$ is the state approximation error of the external agent i. Also, γ and β are real numbers that satisfies $\beta \geq \gamma$.

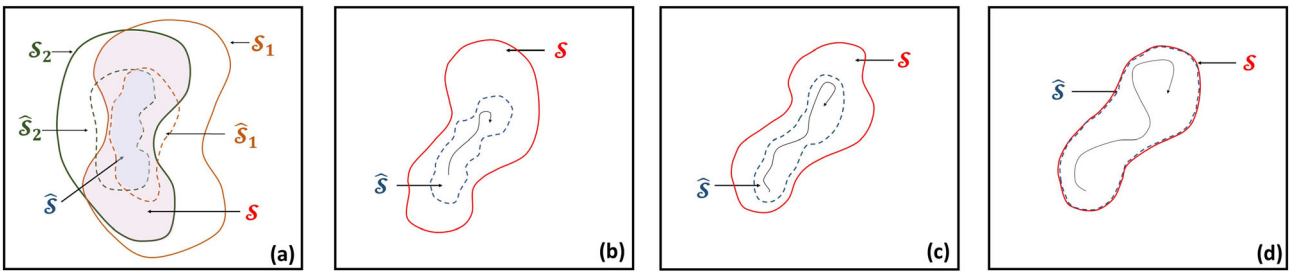


Fig. 3 Approximated safe set convergence to actual safe set. **a** Actual safe set S_1 and S_2 and approximated safe set $\hat{S}_1$ and $\hat{S}_2$ with respect to external agents 1 and 2. Also, the actual safe $\mathcal{S}$ set of agent $\mathcal{A}$ is the intersection of all actual safe sets associated with different agents. Similarly, the approximated safe set $\hat{\mathcal{S}}$ of agent $\mathcal{A}$ is shown in the figure. **b** Safety is ensured since the approximated safe region is inside the boundary of the actual safe region. **c** The approximated safe set is approaching the actual safe set as learning progress **d** By converging $\hat{\mathcal{S}}$ to $\mathcal{S}$, the agent $\mathcal{A}$ has more space for safe maneuver

Using this sector-bounded condition, the upper bound of the state approximation error is now defined as follows:

$$\|\tilde{z}_i\|^2 \leq \frac{1}{\gamma}\|\tilde{y}_i\|^2 \Longrightarrow \|\tilde{z}_i\| \leq \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\| \tag{32}$$

For the given dynamical system of agent $\mathcal{A}$ in Eq. (1) and external agent dynamic in (2), the approximated safe set associated with external agent i can be defined as

$$\hat{\mathcal{S}} = \{x \in \mathbb{R}^n : h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|) \geq 0\},$$
$$\partial\hat{\mathcal{S}} = \{x \in \mathbb{R}^n : h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|) = 0\},$$
$$\mathrm{Int}(\hat{\mathcal{S}}) = \{x \in \mathbb{R}^n : h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|) > 0\}.$$

The function $h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|)$ represents a smooth function that incorporates both the variables x and $\hat{z}_i$. Please note that $\hat{z}_i$ is the estimated state information of the external agent i which is available to the agent $\mathcal{A}$. Besides that, the bounded error is incorporated to ensure strict safety of the agent $\mathcal{A}$ for the observer-based external agent state estimation. The function $h_i(.)$ describes the constraint imposed by the external system. Now, the safe set for the agent $\mathcal{A}$ is derived by taking the intersection of sets associated with all external agents in the shared environment. This intersection ensures that the agent operates within the overlapping boundaries of safety defined by external agents. Then, the approximated safe set of the local agent $\mathcal{A}$ is derived as follows:

$$\hat{\mathcal{S}}(x, \hat{z}) = \hat{\mathcal{S}}_1(x, \hat{z}_1) \cap \hat{\mathcal{S}}_2(x, \hat{z}_2) \ldots \cap \hat{\mathcal{S}}_N(x, \hat{z}_N) \tag{33}$$

with N being the external systems number. The function $h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|)$ is the adaptive interplay control barrier function (AI-CBF). If there exists an extended class $\mathcal{K}_\alpha$ function $\alpha : \mathbb{R} \to \mathbb{R}$ such that for given dynamical system in (1) and (2), the following conditions hold:

$$\sup_{u \in \mathcal{U}}[L_f h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|) + L_g h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|)u + L_F h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|)] \geq -\alpha h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|) \tag{34}$$

where $L_f = \frac{\partial h_i}{\partial x} f(x)$, $L_g = \frac{\partial h_i}{\partial x} g(x)$ and $L_F = \frac{\partial h_i}{\partial \hat{z}_i} F(x)$ are Lie derivatives of $h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|)$ along f, g and F, respectively. Now, the set of control input that satisfies (34) is defined as:

$$K_{\mathrm{cbf}} = \{u \in \mathcal{U} : L_f h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|) + L_g h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|_2)u + L_F h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|) + \alpha h_i(x, \hat{z}_i \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|) \geq 0\}. \tag{35}$$

The safe set is defined here for both agent $\mathcal{A}$ and the external agents' systems. The establishment of the safe set is intrinsically tied to the external system, wherein the learning phase can only yield an approximate safe set. To ensure safety for the agent $\mathcal{A}$, the AI-CBF is formulated as a function of both the local agent and external agents and incorporates the worst-case scenario for the observer-based state estimation of the external agents. The formulated AI-CBF ensures the forward invariance of the approximated safe set $\hat{\mathcal{S}}$, if the following two assumptions are satisfied.

Assumption 2 The condition requires that the agent $\mathcal{A}$ starts its operation from an initial state, and belongs to the interior of the safe set:

$$x_0 \in \mathrm{int}(\mathcal{S}). \tag{36}$$

Assumption 3 The initial state of any external systems denoted as i satisfies the following condition

$$\mathcal{S}_i(x_0, z_{i,0}) = \{x_0 \in \mathbb{R}^n : h_{i,0}(x_0, z_{i,0}) \geq 0\}. \tag{37}$$

Algorithm 1 Safe control with Multiple NN-based observers

Initialize the state of the local agent x that meets the safety condition $x_0 \in \text{int}(\mathcal{S})$.
Observe the output y_i of any external agent i.
Initialize the weights $W_1, W_2, \ldots, W_N$ of the neural network for N external agents.
Initialize the safe approximation $\hat{z}_i$ for external agent i.
Initialize the objective function error threshold δ_{J_i} for external agent i.
while TRUE **do**
  while $J_i \geq \delta_{J_i}$ **do**
    Update NN weight using equation (19).
    Update objective function using equation (18).
  end while
  Calculate the upper bound $\frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|$.
  Design AI-CBF.
  Evaluate the controller using Quadratic Programming in (38).
  Apply the controller $u(x, \hat{z}_i)$.
end while

4 Control framework

It is essential to design a control input to guarantee safety and maintain the local system stability in a multi-agent environment. This requirement highlights the importance of integrating a Lyapunov function $\mathcal{V}_e(x, x_d)$. The constraint in the derivative of the Lyapunov function $\mathcal{V}_e(x, x_d)$ and AI-CBF constraint are unified to achieve robust safety and stability performance. Then, providing a nominal controller $\bar{u}$ for local agent $\mathcal{A}$ to achieve the desired destination, the safety of the local agent operating in a multi-agent shared environment needs to be guaranteed. The nominal controller guides the local agent to the destination. Next, a quadratic programming (QP) [12, 35]-based method has been adopted. Building upon prior research efforts [11, 16], this QP-based controller unifies stability and safety constraints within an optimization framework. By leveraging quadratic programming, the controller facilitates continuous updates of the control actions. The formulation of AI-CBF-based quadratic programming controller is given as:

$$\begin{aligned} u(x, \hat{z}_i) = \underset{(u,\delta)}{\arg\min} \quad & \frac{1}{2}\|u - \bar{u}\|^2 + p\delta^2 \\ \text{s.t.} \quad & L_f h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|) + L_g h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|)u \\ & + L_F h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|) + \alpha h_i(x, \hat{z}_i, \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|) \geq 0 \\ & \dot{\mathcal{V}}_{sys}(x, x_d) \leq B_{\text{cls}}. \end{aligned} \tag{38}$$

where δ serves as a relaxation variable to guarantee quadratic program solvability and p represents the relaxation factor coefficient. Additionally, the controller incorporates the Lyapunov function denoted as $\mathcal{V}_{sys}$ to achieve stability.

Lemma 1 *There exists a control policy u for the dynamic equation given in (4) to guarantee the systems stability.*

$$e^{\mathrm{T}}\Big\{f_a(e(t)) + g_a(e(t))u\Big\} \leq -\gamma\|e\|^2. \tag{39}$$

Theorem 2 (*Closed loop stability*) *The NN weight is updated by Eq. (19) and the learning rate α is a positive constant. Then, NN weight approximation error $\tilde{W}$, external agent state estimation error $\tilde{z}_i$, and the local agent $\mathcal{A}$ regulation error e are all ultimately uniformly bounded (UUB). Moreover, $\tilde{W}$, $\tilde{z}_i$, and e are asymptotically stable when the reconstruction error and relaxation variable δ is zero [32].*

Proof Consider the Lyapunov candidate function

$$\mathcal{V}_{sys} = \mathcal{V}_e + \mathcal{V}_s \tag{40}$$

with $\mathcal{V}_e = \frac{1}{2}\text{tr}\{e^{\mathrm{T}}(t)e(t)\}$ and $\mathcal{V}_s = \frac{1}{2}\tilde{z}_i^{\mathrm{T}}P\tilde{z}_i + \frac{1}{2}\text{tr}\{\tilde{W}_i^{\mathrm{T}}\tilde{W}_i\}$. Here the relaxation variable δ has an upper bound to ensure the system stability. Equation (40) can be written as:

$$\mathcal{V}_{sys} = \frac{1}{2}\text{tr}\{e^{\mathrm{T}}(t)e(t)\} + \frac{1}{2}\tilde{z}_i^{\mathrm{T}}P\tilde{z}_i + \frac{1}{2}\text{tr}\{\tilde{W}_i^{\mathrm{T}}\tilde{W}_i\}. \tag{41}$$

Taking the first derivative of Eq (40)

$$\dot{\mathcal{V}}_{sys} = \text{tr}\{e^{\mathrm{T}}(t)\dot{e}(t)\} + \frac{1}{2}\dot{\tilde{z}}_i^{\mathrm{T}}P\tilde{z}_i + \frac{1}{2}\tilde{z}_i^{\mathrm{T}}P\dot{\tilde{z}}_i + \text{tr}\{\tilde{W}_i^{\mathrm{T}}\dot{\tilde{W}}_i\}. \tag{42}$$

Consider the bounds of activation function and ideal weights, given by $\|\phi(z_i)\| \leq \phi_M$ and $\|W_i\| \leq W_M$ and substituting Lemma 1 and Eq. (26), Eq. (42) can be rewritten as:

$$
\begin{aligned}
\dot{\mathcal{V}}_{sys} \leq & \operatorname{tr}\{e^{\mathrm{T}}[f_a(e)+g_a(e)\hat{u}(x,\hat{z}_i)]\} \\
& -\left[\frac{1}{2}\lambda_{\min}(Q)-\|P\|W_M L_\phi-\frac{1}{2\alpha^2}\|P\|^2\right. \\
& \left.-\frac{1}{2}\|p\|^2-\frac{1}{2}W_M^2L_\phi^2\right]\|\tilde{z}_i\|^2-\left[\alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2\right. \\
& \left.-\alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2-\frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2\right]\|\tilde{W}_i\|^2+\|\varepsilon_f\|^2 \\
\leq & \operatorname{tr}\{e^{\mathrm{T}}[f_a(e)+g_a(e)u-g_a(e)u+g_a(e)\hat{u}]\} \\
& -\left[\frac{1}{2}\lambda_{\min}(Q)-\|P\|W_M L_\phi-\frac{1}{2\alpha^2}\right. \\
& \|P\|^2-\frac{1}{2}\|p\|^2-\frac{1}{2}W_M^2L_\phi^2\Big]\|\tilde{z}_i\|^2-\left[\alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2-\alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2\right. \\
& \left.-\frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2\right]\|\tilde{W}_i\|^2+\|\varepsilon_f\|^2 \\
\leq & -\gamma\|e\|^2-\operatorname{tr}\{e^{\mathrm{T}}g_a(e)\tilde{u}\} \\
& -\left[\frac{1}{2}\lambda_{\min}(Q)-\|P\|W_M L_\phi-\frac{1}{2\alpha^2}\|P\|^2-\frac{1}{2}\|p\|^2\right. \\
& \left.-\frac{1}{2}W_M^2L_\phi^2\right]\|\tilde{z}_i\|^2-\left[\alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2-\alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2\right. \\
& \left.-\frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2\right]\|\tilde{W}_i\|^2+\|\varepsilon_f\|^2 \\
\leq & -\frac{1}{2}\gamma\|e\|^2-\frac{1}{2}\gamma\|e\|^2-\operatorname{tr}\{e^{\mathrm{T}}g_a(e)\tilde{u}\}-\frac{2}{\gamma}\|g_a(e)\tilde{u}\|^2 \\
& +\frac{2}{\gamma}\|g_a(e)\tilde{u}\|^2-\left[\frac{1}{2}\lambda_{\min}\right. \\
& \left.(Q)-\|P\|W_M L_\phi-\frac{1}{2\alpha^2}\|P\|^2-\frac{1}{2}\|p\|^2\right. \\
& \left.-\frac{1}{2}W_M^2L_\phi^2\right]\|\tilde{z}_i\|^2-\left[\alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2\right. \\
& \left.-\alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2-\frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2\right]\|\tilde{W}_i\|^2+\|\varepsilon_f\|^2 \\
\leq & -\frac{1}{2}\gamma\|e\|^2-\left[\frac{1}{2}\gamma\|e\|^2+\operatorname{tr}\{e^{\mathrm{T}}g_a(e)\tilde{u}\}+\frac{2}{\gamma}\|g_a(e)\tilde{u}\|^2\right] \\
& +\frac{2}{\gamma}\|g_a(e)-\left[\frac{1}{2}\lambda_{\min}\right. \\
& \left.(Q)\tilde{u}\|^2-\|P\|W_M L_\phi-\frac{1}{2\alpha^2}\|P\|^2-\frac{1}{2}\|p\|^2\right. \\
& \left.-\frac{1}{2}W_M^2L_\phi^2\right]\|\tilde{z}_i\|^2-\left[\alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2\right. \\
& \left.-\alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2-\frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2\right]\|\tilde{W}_i\|^2+\|\varepsilon_f\|^2 \\
\leq & -\frac{1}{2}\gamma\|e\|^2-\left[\sqrt{\frac{\gamma}{2}}\|e\|\right. \\
& \left.+\sqrt{\frac{2}{\gamma}}\|g_a\tilde{u}\|\right]^2+\frac{2}{\gamma}\|g_a(e)\tilde{u}\|^2-\left[\frac{1}{2}\lambda_{\min}(Q)-\|P\|W_M\right. \\
& \left.L_\phi-\frac{1}{2\alpha^2}\|P\|^2-\frac{1}{2}\|p\|^2-\frac{1}{2}W_M^2L_\phi^2\right]\|\tilde{z}_i\|^2-\left[\alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2\right. \\
& \left.-\alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2-\frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2\right]\|\tilde{W}_i\|^2+\|\varepsilon_f\|^2 \\
\leq & -\frac{1}{2}\gamma\|e\|^2+\frac{2}{\gamma}g_l^2L_u^2\|\tilde{z}_i\|^2-\left[\frac{1}{2}\lambda_{\min}(Q)\right. \\
& -\|P\|W_M L_\phi-\frac{1}{2\alpha^2}\|P\|^2-\frac{1}{2}\|p\|^2W_M^2 \\
& \left.-\frac{1}{2}L_\phi^2\right]\|\tilde{z}_i\|^2-\left[\alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2-\alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2\right. \\
& \left.-\frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2\right]\|\tilde{W}_i\|^2+\|\varepsilon_f\|^2
\end{aligned}
\tag{43}
$$

with g_l being the Lipschitz constant of the function g_a. Also, there exists a Lipschitz constant L_u that satisfy the inequality $\|\tilde{u}(x,\tilde{z}_i)\| \leq L_u\|\tilde{z}_i\|$. Now, adding the relaxation variable, Eq. (43) can be rewritten as:

$$
\begin{aligned}
\dot{\mathcal{V}}_{sys} \leq & -\frac{1}{2}\gamma\|e\|^2-\left[\frac{1}{2}\lambda_{\min}(Q)-\frac{2}{\gamma}g_l^2L_u^2-\|P\|W_M L_\phi\right. \\
& -\frac{1}{2\alpha^2}\|P\|^2-\frac{1}{2}\|p\|^2-\frac{1}{2}W_M^2 \\
& \left.L_\phi^2\right]\|\tilde{z}_i\|^2-\left[\alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2-\alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2\right. \\
& \left.-\frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2\right]\|\tilde{W}_i\|^2+\|\varepsilon_f\|^2+\delta \\
\leq & -\frac{1}{2}\gamma\|e\|^2-\kappa_{i,zc}\|\tilde{z}_i\|^2-\kappa_{i,Wc}\|\tilde{W}_i\|^2+\|\varepsilon_f\|^2+\delta=B_{cls}
\end{aligned}
\tag{44}
$$

with,

$$
\begin{aligned}
\kappa_{i,zc} = & \frac{1}{2}\lambda_{\min}(Q)-\frac{2}{\gamma}g_l^2L_u^2-\|P\|W_M L_\phi-\frac{1}{2\alpha^2}\|P\|^2 \\
& -\frac{1}{2}\|p\|^2-\frac{1}{2}W_M^2L_\phi^2 \\
\kappa_{i,Wc} = & \alpha\|C\|^2\|\hat{\phi}(\hat{z}_i)\|^2-\alpha^2\|C\|^4\|\hat{\phi}(\hat{z}_i)\|^2 \\
& -\frac{1}{2}\alpha^2\|\hat{\phi}(\hat{z}_i)\|^2\|\varepsilon_f\|^2.
\end{aligned}
$$

Now the first derivative of the Lyapunov function $\dot{\mathcal{V}}_{sys}$ is less than zero outside a compact set if

$$
\begin{aligned}
& \|e\| > \sqrt{\frac{2}{\gamma}\|\varepsilon_f\|^2+\delta};\quad \|\tilde{z}\| > \sqrt{\frac{1}{\kappa_{i,zc}}\|\varepsilon_f\|^2+\delta}; \\
& \|\tilde{W}_i\| > \sqrt{\frac{1}{\kappa_{i,Wc}}\|\varepsilon_f\|^2+\delta}.
\end{aligned}
\tag{45}
$$

Here, the condition defined in above equation for $\|e\|$, $\|\tilde{z}\|$ and $\|\tilde{W}_i\|$ ensures the negative semi-definiteness of $\dot{\mathcal{V}}_{sys}$. Also, in this study, the relaxation variable δ is bounded to ensure the system's stability with safety.

5 Simulation result

In this simulation section, we implement the developed algorithm into a multi-agent system to illustrate the secure maneuvering of an autonomous Unmanned Aerial Vehicle (UAV) within a shared airspace environment, alongside other external UAVs.

The primary objective of the local UAV is to successfully reach a predetermined destination while employing collision avoidance strategies using the developed safe controller to evade other concurrently deployed UAVs within the uncertain shared environment. In the experiment, the local UAV does not have direct access to the external UAVs' current states, and it is also unaware of the external UAVs' dynamics. Hence, multiple neural network-based adaptive observers are developed to learn the

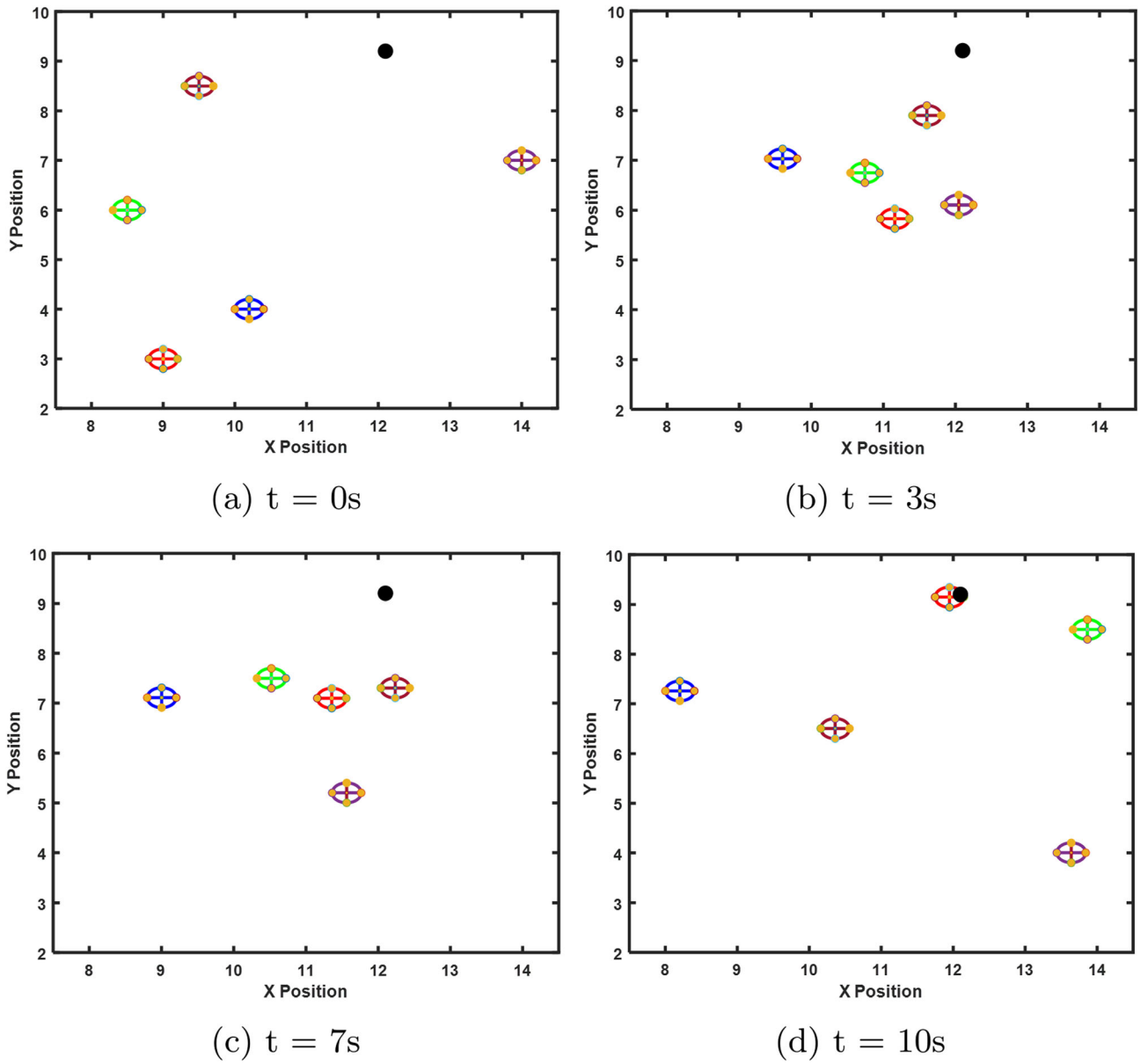
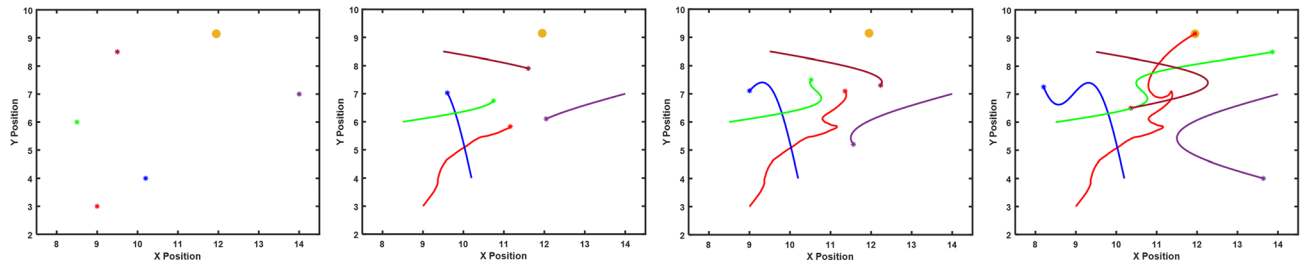


Fig. 4 The motion of local UAV in a multi-UAV environment. Here, the UAV with red color represents the local UAV, and blue, green, magenta, and maroon colored UAVs are external UAVs. Also, the black circle on the upper part of each figure is the destination point for the local UAV. **a** The initial position of all the UAVs in the environment at time $t = 0s$. **b** The position of the UAVs at time $t = 3s$. **c** The position of UAVs at time $t = 7s$. **d** The final position of all the UAVs at time $t = 10s$ (Color figure online)

states of the external UAVs. Also, there are a total of four external UAVs in the system. Now the initial state of the local UAV is selected as $x = [9\ \ 3\ \ 0\ \ 0]^{\mathrm{T}}$ with its position and velocity. Also, the predefined destination point of the UAV is given as $x = [12.2\ \ 9\ \ 0\ \ 0]^{\mathrm{T}}$. Moreover, the initial states of the UAV-1, UAV-2, UAV-3 and UAV-4 are selected as $z_1 = [10.2\ \ 4\ \ 0\ \ 0]^{\mathrm{T}}$, $z_2 = [8.5\ \ 6\ \ 0\ \ 0]^{\mathrm{T}}$, $z_3 = [14\ \ 7\ \ 0\ \ 0]^{\mathrm{T}}$ and $z_4 = [9.5\ \ 8.5\ \ 0\ \ 0]^{\mathrm{T}}$. The intrinsic dynamic function of the local UAV is defined as:

$$f(x) = \begin{bmatrix} -x_1 + \frac{1}{2}x_2^2 \\ -0.4x_2^2 \\ x_2[\cos(2x_1 + 1)^2 - 1] - x_1 \\ x_4[\cos(2x_3 + 1)^2 - 1] - x_3 \end{bmatrix}$$

$$g(x) = [0 \quad 0 \quad \cos(2x_1 + 1) \quad \cos(2x_3 + 1)]^{\mathrm{T}}$$



(a) t = 0s (b) t = 3s (c) t = 7s (d) t = 10s

Fig. 5 The trajectory of all UAVs in the environment. The trajectory of the local UAV is represented by the red curve. Also, blue, green, magenta, and maroon curves are the trajectories of external UAVs in the environment. This figure shows the local UAV is approaching the predefined destination while avoiding collision with other UAVs to ensure the safety of the system over time (Color figure online)

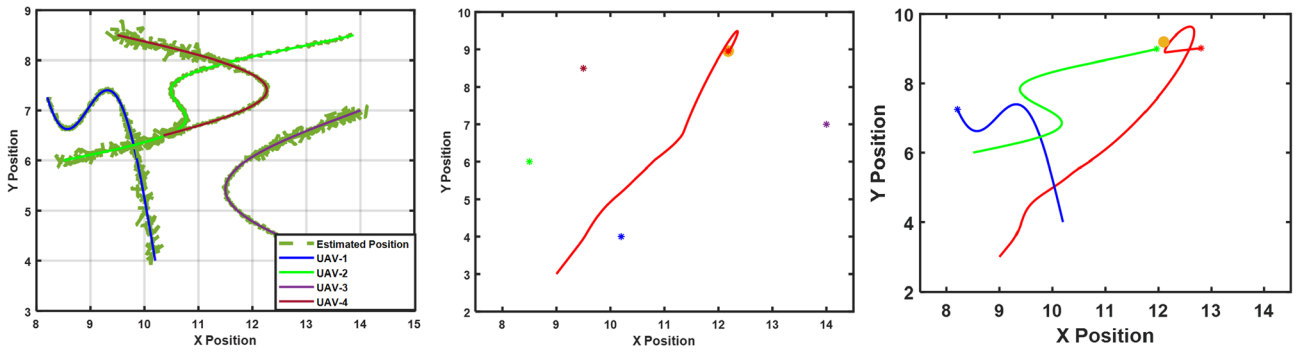
Moreover, the dynamics of the UAV-1, UAV-2, UAV-3 and UAV-4 are chosen as [36]. To design a neural network-based observer, it is essential that the square matrix A be a Hurwitz matrix. This means that all of its eigenvalues must have negative real parts that are strictly negative, specifically $\mathrm{Re}[A] < 0$. Now, the external UAVs' states are observable if the observability matrix $O = [C \;\; CA \ldots \; CA^{n-1}]$ is full rank, i.e., the matrix determinant is nonzero. The selection of matrices A ensures that both A and $A - HC$ have eigenvalues with strictly negative real parts, making them Hurwitz matrices.

$$A = \begin{bmatrix} -1 & 2 & 0 & 0 \\ -1 & -3 & 0 & 0 \\ 0 & 0 & -10 & 12 \\ 0 & 0 & -7 & -4 \end{bmatrix}$$

In this multi-neural network adaptive observer-based design, the activation function of each NN is selected as a hyperbolic tangent function, i.e., $\tanh(.)$. Also, the learning rate α of the NN is selected as 1×10^{-4}. Next, the AI-CBF is defined as $h_i(x, z_i) = \|x - z_i\|_2^2 - \frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|_2 - r_{\min}$. Please note that the second term $\frac{1}{\sqrt{\gamma}}\|\tilde{y}_i\|_2$ is used to incorporate the estimation error bound of the external agents into the CBF design with $\gamma = 1.5$. This term ensures the strict safety of the local UAV. Also, $r \geq r_{\min}$ with $r_{\min} = \frac{\|v\|_2}{2}$. Here, r represents the minimum safe distance between two UAVs. Since the state information only includes the positions of objects, merely measuring the distance between two UAVs isn't sufficient to guarantee safety. To ensure that no part of the local UAV comes into contact with other UAVs, we include the velocity $v = [x_3 \;\; x_4]^{\mathrm{T}}$ in the barrier function. Then, the relaxation factor coefficient for the quadratic program (QP) in (38) is selected as $p = 0.5$.

The differential equations are solved using MATLAB *ode45* and QP is solved using MATLAB *quadprog* function. The effectiveness of the developed algorithm has been showcased through a series of figures. In Fig. 4, we illustrate the safe maneuvering of the local agent within an environment shared with other agents. In Fig. 4a, we display the initial positions of the UAVs in the system, where the red color represents the local UAV, and the other colors represent external UAVs. These UAVs initiate their motion from the different corners of the figure and move in various directions. The goal of the local UAV is to reach a predefined destination indicated by a black circle. Given that other agents are navigating within the same environment, the local agent must ensure its safety by avoiding collisions with them. Figure 4 clearly demonstrates that the local UAV approaches and ultimately reaches its destination while maintaining safety in the presence of other UAVs.

The depicted UAV locations correspond to times $t = 0\,s$, $t = 3\,s$, $t = 7s$, and $t = 10s$ in Fig. 4a, d. To provide a more comprehensive representation of the secure maneuvers executed by the local UAV, we have plotted the trajectories of the UAVs in Fig. 5 at various time points: $t = 0\,s$, $t = 3\,s$, $t = 7\,s$, and $t = 10s$. Here, the red curve represents the trajectory of the local UAV, while the blue, green, magenta, and maroon curves represent the trajectories of external UAV-1, UAV-2, UAV-3, and UAV-4, respectively. Figure 5 effectively illustrates that the local UAV successfully reaches its destination while ensuring collision avoidance with the other UAVs. In Fig. 5c, d, it is evident that the local UAV adjusts its direction to maintain a safe distance from the other UAVs before continuing toward its destination. It is important to note that the points at which the red curve (local UAV) intersects with the blue, green, and maroon curves (external UAVs) do not represent simultaneous collisions but occur at different times. In Fig. 6a, we present the state estimation of external agents using the developed neural network (NN) adaptive observer. The true positions of UAV-1, UAV-2, UAV-3, and UAV-4 are represented by the blue, green, magenta, and maroon curves, while the estimated positions are shown in dark green color. Figure 6a effectively illustrates that as time progresses and the neural network observer is well-trained, the state estimation error decreases and approaches



(a) State estimation of external UAVs

(b) Original path of local UAV for static external UAVs

(c) Same final destination for local UAV and UAV-2

Fig. 6 This figure (**a**) illustrates both the estimated and actual positions of external agents, **b** the original path of local UAV for static external UAVs, and **c** a scenario demonstrating how the local UAV ensures strict safety in the presence of two external UAVs through the developed method

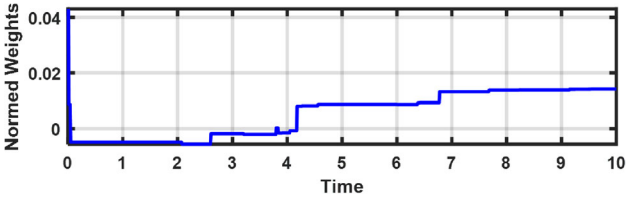


Fig. 7 NN normed weight convergence for UAV-1

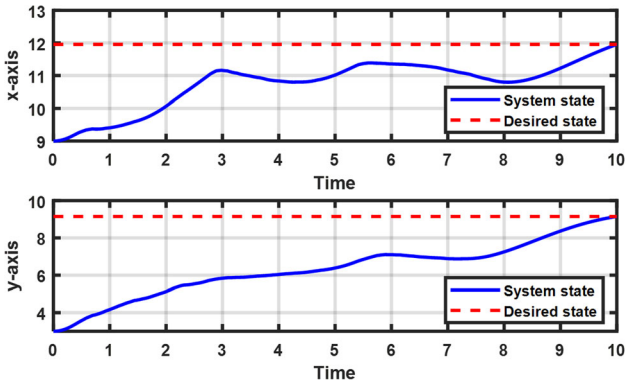


Fig. 8 Desired state tracking error of local UAV in x- and y-direction

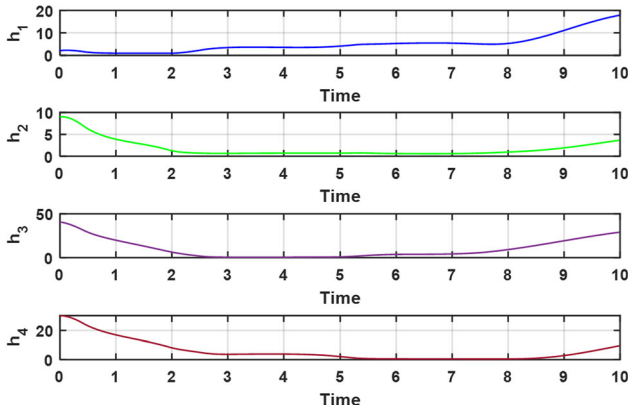


Fig. 9 AI-CBF h_1, h_2, h_3 and h_4 for UAV-1, UAV-2, UAV-3 and UAV-4, respectively

zero. This observation serves as compelling evidence for the effectiveness of the observer design. In Fig. 6b, the initial path planning of the local UAV is demonstrated in the presence of static external UAVs. As there are no collisions due to the lack of movement from the other UAVs, the local UAV successfully reaches its destination with minimal deviation from its planned path. In Fig. 6c, a scenario is presented to illustrate how the local UAV ensures strict safety using the developed methodology in the presence of UAV-1 and UAV-2. Here, the final destination of UAV-2 is in close proximity to the desired destination of the local UAV. To prioritize safety, the local UAV places a strong emphasis on collision avoidance with UAV-2, even at the expense of reaching its desired destination. Figure 6c visually depicts the local UAV altering its course at the last moment to maintain a safe distance from UAV-2. It serves as a demonstrative example of the effectiveness of our approach, particularly in risky situations. The normed weight of the neural network for learning the unknown dynamic of UAV-1 is plotted in Fig. 7 which demonstrates the convergence of NN weight over time. Figure 8 illustrates the tracking error of the local UAV's desired state in both the x- and y-axes.

The blue curve represents the actual state of the local UAV, while the red dashed line indicates the predefined desired state. The adaptive interplay control barrier functions (AI-CBF) for UAV-1, UAV-2, UAV-3 and UAV-4 are shown in Fig. 9. In this figure, all the barrier functions remain positive indicating that the local UAV always ensures safety in the multi-UAV environment. Initially,

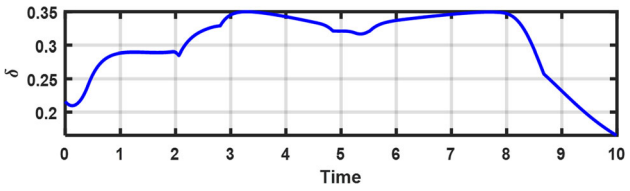
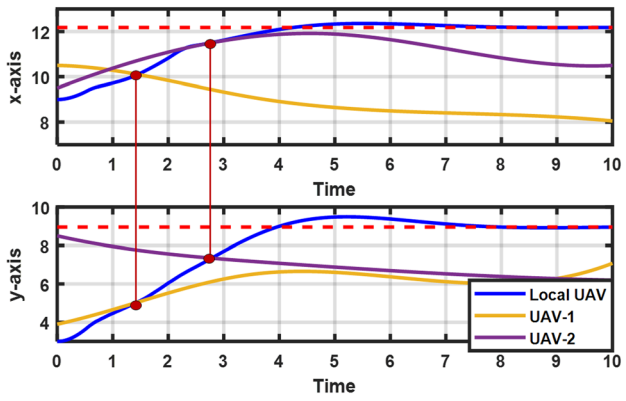


Fig. 10 Evolution of relaxation variable δ

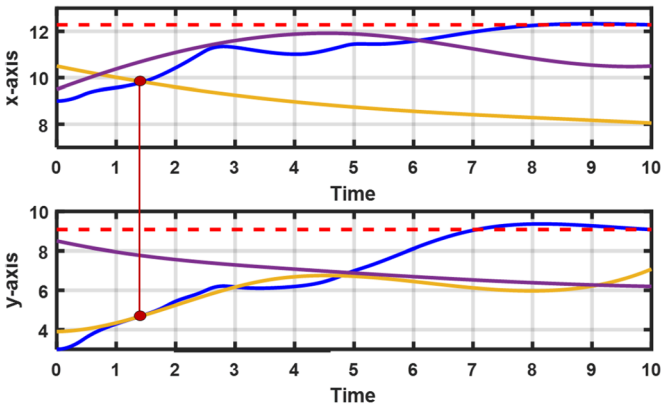
after the deployment of UAVs, the local UAV approaches UAV-1, causing its states to approach the boundary of the safe set.

At that period, the value of the function is close to zero. Subsequently, the AI-CBF h_1 increases as the local UAV moves away from UAV-1.

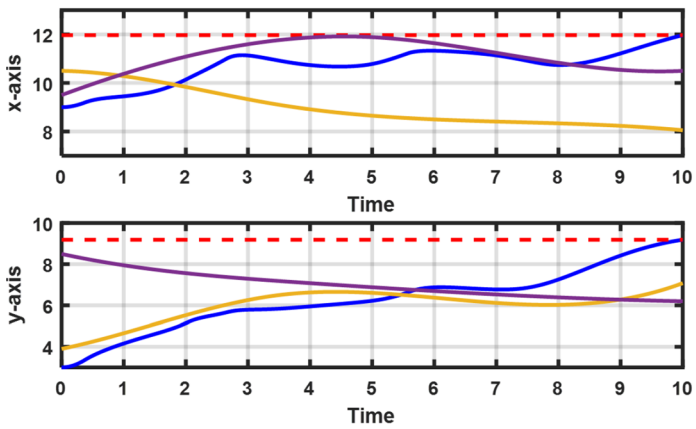
Similarly, the AI-CBFs h_2, h_3, and h_4 experience decreases at different time points when UAV-2, UAV-3, and UAV-4 come closer to the local UAV on different occasions. Next, the relaxation variable δ is shown over time in Fig. 10. The QP relaxes by increasing the parameter δ to ensure safety over the tracking performance by enforcing AI-CBF. Here, in this figure, the value of δ increases on different occasions to guarantee the local agent's safety. A comparison study is shown in Fig. 11 to show the effectiveness of the developed AI-CBF-based decentralized safe control design. The trajectories of local and external UAVs are shown for three scenarios: without CBF, with CBF, and with AI-CBF. This comparison highlights the performance differences in crash avoidance among the three cases. In all scenarios, the local UAV has



(a) Position trajectory of UAVs in x and y directions without considering the safety (without CBF).



(b) Position trajectory of UAVs in x and y directions with considering the safety using CBF.



(c) Position trajectory of UAVs in x and y directions with considering the robust safety using AI-CBF.

Fig. 11 The position trajectories of a scenario involving a local UAV along with UAV-1 and UAV-2 are depicted. The positions of the UAVs are shown for both the x- and y-axes under three conditions: **a** without control barrier function (CBF), **b** with CBF, and **c** with AI-CBF

the same predefined destination of $x = [12.2 \;\; 9 \;\; 0 \;\; 0]^{\mathrm{T}}$. To simplify and clarify the comparison, only two external UAVs are used. Figure 11a illustrates the positions of the UAVs in the x and y-directions over time without considering the CBF for the local UAV. In this figure, the blue, yellow, and magenta curves represent the trajectories of the local UAV, UAV-1, and UAV-2, respectively, while the red dashed line indicates the predefined destination. It is evident that the local UAV intersects with both UAV-1 and UAV-2 in the x- and y-axes simultaneously, indicating potential collisions with the external UAVs. To clearly illustrate this, a maroon line with circles at both ends is placed at the intersection points of the UAVs' trajectories in the x- and y-axes. Next, Fig. 11b demonstrates the UAV trajectories for the case with CBF. In this scenario, the use of CBF successfully avoids a collision with UAV-2. However, the CBF fails to prevent a collision with UAV-1 because, initially, the learning of the unknown state of this external UAV is not accurate. Using the same scenario, the developed decentralized control design incorporating the multi-NN-based adaptive observer with AI-CBF is implemented. Figure 11c demonstrates that the local UAV successfully avoids collisions with all external UAVs using the developed algorithm. In summary, the simulation results presented here highlight the efficacy of the developed algorithm.

6 Conclusion

This paper has developed a novel safe control approach for local agents in a challenging multi-agent environment where the dynamics of external agents are both uncertain and uncontrollable, and accurate state information is unavailable. The developed method utilizes multiple neural network-based adaptive observers to estimate the states of these external agents. Through the integration of state information from both the external and local agents, an adaptive interplay control barrier function (AI-CBF) has been designed to ensure the local agent's safety in the presence of other external agents. Notably, the AI-CBF guarantees the strict safety of the local agent by maintaining the forward invariance of an approximated safe set. Importantly, this algorithm has been proven to ensure system safety without the need for precise knowledge of the actual safe set. Furthermore, our developed algorithm introduces the AI-CBF for safety constraints and the control Lyapunov function (CLF) for stability. This AI-CBF and CLF combination is used for safe control development, which guides the local agent to a predefined destination point while guaranteeing safety and stability. The neural network-based observer stability and the overall system stability have been rigorously demonstrated through Lyapunov stability analyses. Finally, a simulation study is conducted to demonstrate the efficiency and practical applicability of the developed algorithm.

Funding This work was supported by the National Science Foundation under Grant 2144646 and FA945324CX018.

Data availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Zhou S, Guo K, Yu X et al (2021) Fixed-time observer based safety control for a quadrotor UAV. IEEE Trans Aerosp Electron Syst 57(5):2815–2825
2. Dey S, Xu H (2024) Distributed adaptive flocking control for large-scale multiagent systems. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2023.3343666
3. Guiochet J, Machin M, Waeselynck H (2017) Safety-critical advanced robots: a survey. Robot Auton Syst 94:43–52
4. Koopman P, Wagner M (2017) Autonomous vehicle safety: an interdisciplinary challenge. IEEE Intell Transp Syst Mag 9(1):90–96
5. Olofsson B, Nielsen L (2020) Using crash databases to predict effectiveness of new autonomous vehicle maneuvers for lane-departure injury reduction. IEEE Trans Intell Transp Syst 22(6):3479–3490
6. Dey S, Xu H (2022) Decentralized adaptive tracking control for large-scale multi-agent systems under unstructured environment. In: 2022 IEEE symposium series on computational intelligence (SSCI), IEEE, pp 900–907
7. Lamport L (1977) Proving the correctness of multiprocess programs. IEEE Trans Softw Eng 2:125–143
8. Hewing L, Wabersich KP, Menner M et al (2020) Learning-based model predictive control: toward safe learning in control. Ann Rev Control Robot Auton Syst 3:269–296
9. Brunke L, Greeff M, Hall AW et al (2022) Safe learning in robotics: from learning-based control to safe reinforcement learning. Ann Rev Control Robot Auton Syst 5:411–444
10. Cheng R, Orosz G, Murray RM et al (2019) End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In: Proceedings of the AAAI conference on artificial intelligence pp 3387–3395
11. Ames AD, Coogan S, Egerstedt M, et al (2019) Control barrier functions: theory and applications. In: 2019 18th European control conference (ECC), IEEE, pp 3420–3431
12. Ames AD, Xu X, Grizzle JW et al (2016) Control barrier function based quadratic programs for safety critical systems. IEEE Trans Autom Control 62(8):3861–3876
13. Bansal S, Chen M, Herbert S et al (2017) Hamilton–Jacobi reachability: a brief overview and recent advances. In: 2017 IEEE 56th Annual conference on decision and control (CDC), IEEE, pp 2242–2253
14. Chen M, Shih JC, Tomlin CJ (2016) Multi-vehicle collision avoidance via Hamilton–Jacobi reachability and mixed integer

programming. In: 2016 IEEE 55th conference on decision and control (CDC), IEEE, pp 1695–1700
15. Bansal S, Bajcsy A, Ratner E et al (2020) A Hamilton–Jacobi reachability-based framework for predicting and analyzing human motion for safe planning. In: 2020 IEEE international conference on robotics and automation (ICRA), IEEE, pp 7149–7155
16. Marvi Z, Kiumarsi B (2021) Barrier-certified learning-enabled safe control design for systems operating in uncertain environments. IEEE/CAA J Autom Sin 9(3):437–449
17. Cheng R, Khojasteh MJ, Ames AD et al (2020) Safe multi-agent interaction through robust control barrier functions with learned uncertainties. In: 2020 59th IEEE conference on decision and control (CDC), IEEE, pp 777–783
18. Jankovic M, Santillo M (2021) Collision avoidance and liveness of multi-agent systems with CBF-based controllers. In: 2021 60th IEEE conference on decision and control (CDC), IEEE, pp 6822–6828
19. Notomista G, Wang M, Schwager M et al (2020) Enhancing game-theoretic autonomous car racing using control barrier functions. In: 2020 IEEE international conference on robotics and automation (ICRA), IEEE, pp 5393–5399
20. He S, Zeng J, Zhang B et al (2021) Rule-based safety-critical control design using control barrier functions with application to autonomous lane change. In: 2021 American control conference (ACC), IEEE, pp 178–185
21. Ames AD, Grizzle JW, Tabuada P (2014) Control barrier function based quadratic programs with application to adaptive cruise control. In: 53rd IEEE conference on decision and control, IEEE, pp 6271–6278
22. Grandia R, Taylor AJ, Ames AD et al (2021) Multi-layered safety for legged robots via control barrier functions and model predictive control. In: 2021 IEEE international conference on robotics and automation (ICRA), IEEE, pp 8352–8358
23. Long K, Dhiman V, Leok M et al (2022) Safe control synthesis with uncertain dynamics and constraints. IEEE Robot Autom Lett 7(3):7295–7302
24. Wang S, Lyu B, Wen S et al (2022) Robust adaptive safety-critical control for unknown systems with finite-time elementwise parameter estimation. IEEE Trans Syst Man Cybern: Syst 53(3):1607–1617
25. Choi J, Castaneda F, Tomlin CJ et al (2020) Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions. arXiv preprint arXiv:2004.07584
26. Abdollahi F, Talebi HA, Patel RV (2006) A stable neural network-based observer with application to flexible-joint manipulators. IEEE Trans Neural Netw 17(1):118–129
27. Liu D, Huang Y, Wang D et al (2013) Neural-network-observer-based optimal control for unknown nonlinear systems using adaptive dynamic programming. Int J Control 86(9):1554–1566
28. Vamvoudakis KG, Lewis FL (2010) Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. Automatica 46(5):878–888
29. Zeng J, Zhang B, Sreenath K (2021) Safety-critical model predictive control with discrete-time control barrier function. In: 2021 American control conference (ACC), IEEE, pp 3882–3889
30. Nagumo M (1942) Über die lage der integralkurven gewöhnlicher differentialgleichungen. Proc Physico-Math Soc Jpn 3rd Ser 24:551–559
31. Scarselli F, Tsoi AC (1998) Universal approximation using feedforward neural networks: a survey of some existing methods, and some new results. Neural Netw 11(1):15–37
32. Xu H, Jagannathan S (2013) Stochastic optimal controller design for uncertain nonlinear networked control system via neuro dynamic programming. IEEE Trans Neural Netw Learn Syst 24(3):471–484
33. Dragomir SS (2017) A note on young's inequality. Revista de la Real Academia de Ciencias Exactas. Físicas y Naturales Serie A Matemáticas 111:349–354
34. Khalil HK (2015) Nonlinear control. Pearson Education, New York, NY, USA
35. Nocedal J, Wright SJ (2006) Quadratic programming. Numerical optimization. Springer, Berlin, pp 448–492
36. Zhou Z, Xu H (2022) A novel mean-field-game-type optimal control for very large-scale multiagent systems. IEEE Trans Cybern 52(6):5197–5208. https://doi.org/10.1109/TCYB.2020.3028267