Adaptive Lifelong Safe Learning based Intelligent Tracking Control for Nonlinear System under Unstructured Environment with Non-stationary Tasks

Shawon Dey and Hao Xu

NAECON 2024 - IEEE National Aerospace and Electronics Conference | 979-8-3503-6762-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/NAECON61878.2024.10670653

***Abstract*— In this article, a novel adaptive lifelong safe learning algorithm has been designed for real-time cost-effective tracking control in unstructured environments with non-stationary tasks. Learning sequential non-stationary tasks while retaining memory of previous tasks in real-time control is particularly challenging due to the computational complexity of handling non-stationary tasks. To address this, we develop a dynamic task selection-based adaptive lifelong learning algorithm for safe optimal tracking control and also incorporate a control barrier function (CBF) within the cost function. The learning algorithm includes a novel real-time adaptive lifelong learning-based actor-critic mechanism to achieve safe optimal control in unstructured environments. Specifically, a hybrid offline-online learning framework has been developed, where the probability distributions of offline-trained weights for sequential dynamic tasks are utilized in online training to select the most relevant previous tasks in relation to the current online task. The developed learning framework also includes a fairness term to avoid repeatedly selecting specific tasks. The selected previous tasks are then integrated with a weight consolidation scheme in the designed critic weight update law to achieve adaptive lifelong learning. This method balances catastrophic forgetting with online computational efficiency for optimal safe control. Eventually, the effectiveness of the developed algorithm is validated through numerical simulations.**

I. Introduction

In real-world applications, autonomous systems like unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) have seen widespread use across various tasks. These tasks include battlefield target tracking [1], search and rescue operations [2], fire detection [3], and transportation. Additionally, these autonomous systems are increasingly being deployed in environmental monitoring, disaster response, agricultural automation, and infrastructure inspection, showcasing their versatility and critical role in enhancing efficiency and safety in diverse fields. However, ensuring performance and safety during real-time task execution, particularly in dynamic and unstructured environments, is a challenging endeavor. Numerous studies have explored optimal tracking control in both continuous [4] and discrete [5] time frameworks to enhance system performance. Additionally, significant research has been dedicated to safe control [6] and safe learning [7], [8] systems. Recently, a significant number of studies combined the performance and safety [9], [10], [11] by integrating the control barrier function (CBF) [12] with the cost function to ensure both safety and optimality. However, these existing methods do not address the challenge of autonomous systems executing real-time non-stationary sequential tasks while maintaining performance and safety. A vital capability for executing various tasks in real-time environments is the ability to learn new tasks sequentially without forgetting previously acquired skills. This feature is widely used in humans and animals to continually acquire and transfer skills [13] throughout their lifespan due to synaptic consolidation in the brain and is known as lifelong learning. The lifelong learning [13], [14] has been extensively researched within the neural network and machine learning communities. It emphasizes that intelligent agents need to learn and remember multiple tasks in real-world settings where tasks can switch unpredictably and may not recur frequently. This introduces the problem of catastrophic forgetting [15] in artificial neural networks, where knowledge of previous tasks is lost as new tasks are learned sequentially. To address this issue, [15] introduced an Elastic Weight Consolidation (EWC) algorithm for artificial neural networks. The EWC algorithm slows down learning on certain network weights based on their importance to previously learned tasks, enabling the network to learn new tasks sequentially without forgetting older ones. Recently, this algorithm has been adopted by the control community for the trajectory-tracking problem [16] to manage sequential tasks. However, since EWC considers all the weights of previously learned tasks to compare with the current task and prevent forgetting, it is challenging to implement in real-time systems because of its computational complexity, particularly when the environment is uncertain and contains multiple non-stationary tasks. To adapt to environmental changes by executing sequential tasks, an autonomous agent must strike a balance between optimal and safe decisions and efficiently remembering previous tasks to handle future tasks that are similar to past ones.

To tackle this challenge, a novel adaptive lifelong safe learning-based real-time optimal tracking control algorithm has been developed for unstructured environments. In this context, the goal of the autonomous system is to follow a trajectory and execute multiple non-stationary tasks, which involves avoiding multiple obstacles in dynamic environments. The optimal control problem is formulated by integrating the control barrier function (CBF) to embed safety into the cost function. To solve the optimal safe control problem, the Hamilton-Jacobi-Bellman (HJB) equation must be addressed. Reinforcement learning (RL) [17], [18] and adaptive

The authors are with the Department of Electrical and Biomedical Engineering, University of Nevada, Reno, NV, 89557 USA e-mail: shawondey@nevada.unr.edu; haoxu@unr.edu.

dynamic programming (ADP) [19] techniques have been employed to solve this equation. The proposed algorithm introduces a hybrid offline-online learning framework, that includes a novel real-time adaptive lifelong safe learning-based actor-critic approach, where the critic neural network's learning is modified using a dynamic task selection-based adaptive lifelong learning strategy. Each task is initially trained offline using the actor-critic neural network approach. Subsequently, a real-time neural network-based adaptive lifelong safe learning actor-critic algorithm is proposed. Due to the dynamically changing environment with non-stationary tasks, the offline-trained neural network weights are integrated with a probability distribution function. The learned distribution functions of the weights from previous tasks are compared with the current task in real time to identify tasks that are most similar to the current online task. Additionally, a fairness term is incorporated into the task selection algorithm to prevent repeatedly selecting specific tasks and completely neglecting tasks that are not encountered in the current environment. These selected tasks are then incorporated into the objective function of the critic neural network using elastic weight consolidation (EWC) term to recall the most relevant previous tasks. This reduces computational complexity and facilitates quick, safe decision-making in optimal tracking control. The primary contributions of this work are as follows.:

- A novel dynamic task selection-based adaptive lifelong safe learning algorithm is designed for the real-time safe optimal tracking control problem in the presence of an unstructured environment.
- A novel adaptive lifelong safe learning-based critic neural network is developed, where previous tasks are selected by balancing relevance and fairness. These tasks are then incorporated into the critic algorithm's objective function using an EWC term to perform non-stationary sequential tasks in real-time optimal safe tracking control.

The structure of the paper is as follows: Section II presents the problem formulation. In Section III, the development of the adaptive lifelong learning-based actor-critic algorithm is detailed. The simulation study is covered in Section IV. Finally, Section V offers the conclusion.

II. Problem Formulation

Consider the following differential equation of a nonlinear affine system of an agent $\mathcal{A}$

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

where $x(t) \in \mathbb{R}^n$ represent the system state and $u(t) \in \mathbb{R}^m$ represents the control input. Moreover, The functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ represents the intrinsic dynamics of the system. Next, the external obstacles dynamic can be represented as follows:

$$\dot{z}_q(t) = f_o(z_q(t), u_q) \tag{2}$$

where the notation q represents the index of the obstacle and z_q denotes the system state of the obstacle q. Please note that the obstacle in this framework can be a dynamic or static obstacle. Next, the objective of each agent $\mathcal{A}$ is to track a predefined trajectory while avoiding collision with the external obstacles in the environment. Now, the cost function of the agent $\mathcal{A}$ is defined as follows:

$$\mathcal{V}(x,u) = \int_0^\infty [L(x(t),u(t)) + h(x,z)]dt \tag{3}$$

where the first term is defined as $L(x(t),u(t)) = \|e\|_Q^2 + \|u\|_R^2$, captures the state error and control input's quadratic normals weighted by Q and R. Here, $e(t) = x - x_d$ represents the tracking error of the agent. The term x_d represents the reference trajectory. Then, the tracking error dynamic can be defined as follows:

$$\begin{aligned} de(t) &= dx(t) - dx_d(t) \\ &= [f_r(e) + g_r(e)u]dt \end{aligned} \tag{4}$$

with $f_r(e) = f(e + x_d) - (dx_d/dt)$ and $g_r(e) = g(e + x_d)$. The second term $h(x,z)$ represents a control barrier function. Here, the CBF has been incorporated into the cost function to ensure safety with optimality. Please note that, a continuously differentiable smooth function $h(x,z) : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ is known as CBF for the safe set $\mathcal{S} = \{x \subset \mathbb{R}^n : b(x,z) \geq 0\}$ if there exists a locally Lipschitz class $\mathcal{K}$ functions β_1, β_2 and β_3 such that the following condition is satisfied [9]:

$$\frac{1}{\beta_1(b(x,z))} \leq h(x,z) \leq \frac{1}{\beta_2(b(x,z))} \tag{5}$$

$$\dot{h}(x,z) \leq \beta_3(b(x,z)) \tag{6}$$

Next, Considering continuous dynamics of agent $\mathcal{A}$ given in (1) and the cost function in (3), an admissible control needs to be evaluated to achieve the optimal cost function in (3). According to the optimal control [9] and Bellman's optimality principle [9], the Hamiltonian is:

$$\begin{aligned} H(x,u) = L(x(t),u(t)) + h(x,z) + \partial_x \mathcal{V}(x,u) \\ [f(x) + g(x)u(t)] \end{aligned} \tag{7}$$

with $f(x)$ and $g(x)$ are nonlinear functions and $h(x,u)$ represents the barrier function. Now, the optimal control the agent $\mathcal{A}$ is evaluated as follows:

$$u(t) = -\frac{1}{2}\, R^{-1} g^T(x) \partial_x \mathcal{V}(x,u) \tag{8}$$

Next, the corresponding HJB equation can be achieved by substituting the optimal control into the Hamiltonian defined in equation (7)

$$\begin{aligned} &\|x\|_Q^2 + h(x,z) + \partial_x \mathcal{V}(x,u) f(x) - 1/4\; R^{-1} g^T(x) \partial_x \\ &\mathcal{V}(x,u) g^T(x) \partial_x \mathcal{V}(x,u) = 0 \end{aligned} \tag{9}$$

Next, the agent $\mathcal{A}$ has been assigned to execute M number of non-stationary tasks during its maneuvering period. Additionally, the index i is used to denote the current task, while j represents the previous task. Now, addressing the optimal tracking control problem involves tackling the Hamilton-Jacobi-Bellman (HJB) equation to derive the optimal value function. Furthermore, the agent needs to efficiently remember previous tasks without increasing computational

complexity, while also solving the HJB equation to ensure optimal and safe maneuvering in the current task, which presents significant challenges. To tackle this challenge, An adaptive lifelong safe learning-based actor-critic algorithm with sequential dynamic tasks selection mechanism is developed in the following section.

III. Adaptive Lifelong Safe Learning based Actor-Critic Algorithm

In this part of the study, the agent aims to learn the optimal value function for the current task while retaining knowledge of previous tasks similar to the current one, without increasing computational complexity. Additionally, the agent must ensure that its strategy remains safe and optimal. Now, the ideal value function of the critic neural network for the current task i is defined as:

$$\mathcal{V}_i(x,u) = \mathcal{W}_{i,\mathcal{V}}^T \phi_{i,\mathcal{V}}(x,u) + \varepsilon_{\text{HJB}}(x) \tag{10}$$

In this context, $\mathcal{W}_{i,\mathcal{V}}$ represents the neural network weights, $\phi_{i,\mathcal{V}}(x,u)$ is the activation function for the i-th task, and $\varepsilon_{\text{HJB}}(x)$ denotes the reconstruction error. Next, by substituting the ideal value function (10) into the HJB equation (9), the effect of the reconstruction error is considered:

$$\begin{aligned}
&\|x\|_Q^2 + h(x,z) + \partial_x[\mathcal{W}_{i,\mathcal{V}}^T \phi_{i,\mathcal{V}}(x,u) + \varepsilon_{\text{HJB}}(x)] - \frac{1}{4}\partial_x \\
&[\mathcal{W}_{i,\mathcal{V}}^T \phi_{i,\mathcal{V}}(x,u) + \varepsilon_{\text{HJB}}(x)]^T + \mathcal{D}(x)\partial_x[\mathcal{W}_{i,\mathcal{V}}^T \phi_{i,\mathcal{V}}(x,u) \\
&+ \varepsilon_{\text{HJB}}(x)] = 0 \\
&\|x\|_Q^2 + h(x,z) + \mathcal{W}_{i,\mathcal{V}}^T \partial_x \phi_{i,\mathcal{V}}(x,u) f(x) - 1/4\ \mathcal{W}_{i,\mathcal{V}}^T \partial_x \\
&\phi_{i,\mathcal{V}}(x,u)\mathcal{D}(x)\mathcal{W}_{i,\mathcal{V}}^T \partial_x \phi_{i,\mathcal{V}}(x,u) + \varepsilon_{\text{HJBa}} = 0
\end{aligned} \tag{11}$$

with $\mathcal{D}(x) = g(x)R^{-1}g^T(x)$ and $\varepsilon_{\text{HJBa}}$ is defined as follows:

$$\begin{aligned}
&\varepsilon_{\text{HJBa}} = -1/2\mathcal{W}_{i,\mathcal{V}}^T \partial_x \phi_{i,\mathcal{V}}(x,u)\mathcal{D}(x)\partial_x \varepsilon_{\text{HJB}}(x) - 1/4\ \partial_x \\
&\varepsilon_{\text{HJB}}(x)\mathcal{D}(x)\partial_x \varepsilon_{\text{HJB}}(x) + \partial_x \varepsilon_{\text{HJB}}(x) f(x)
\end{aligned} \tag{12}$$

Next, the approximated value function for the current task i is defined as follows:

$$\hat{\mathcal{V}}_i(x,u) = \hat{\mathcal{W}}_{i,\mathcal{V}}^T \phi_{i,\mathcal{V}}(x,u) \tag{13}$$

Now, the HJB equation error is defined as follows by inserting the estimated value function from the equation (13) in the HJB equation (9):

$$\begin{aligned}
&e_{\text{HJB}} = \|x\|_Q^2 + \hat{h}(x,z) + \hat{\mathcal{W}}_{i,\mathcal{V}}^T \partial_x \phi_{i,\mathcal{V}}(x,u) f(x) - 1/4 \\
&\hat{\mathcal{W}}_{i,\mathcal{V}}^T \partial_x \phi_{i,\mathcal{V}}(x,u)\mathcal{D}(x)\hat{\mathcal{W}}_{i,\mathcal{V}}^T \partial_x \phi_{i,\mathcal{V}}(x,u)
\end{aligned} \tag{14}$$

Next, the objective function of the neural network training for the i-th task can be derived as follows:

$$E_i(\hat{\mathcal{W}}_{i,\mathcal{V}}) = \frac{1}{2} e_{\text{HJB}}^T e_{\text{HJB}} \tag{15}$$

Then, the normalized weight update law [17] of the neural network is defined as:

$$\dot{\hat{\mathcal{W}}}_{i,\mathcal{V}} = -\alpha_{i,\mathcal{V}} \frac{\sigma_{i,\mathcal{V}}}{1 + \sigma_{i,\mathcal{V}}^T \sigma_{i,\mathcal{V}}} e_{\text{HJB}}^T \tag{16}$$

with, $\sigma_{i,\mathcal{V}} = \partial_x \phi_{i,\mathcal{V}}(x,u) f(x) - \frac{1}{2}\partial_x \phi_{i,\mathcal{V}}(x,u)\mathcal{D}(x)\partial_x \phi_{i,\mathcal{V}}(x,u)\hat{\mathcal{W}}_{i,\mathcal{V}}$.

Remark 1: Please note that this update law only incorporates training for the current i-th task and does not include the weights of previous tasks. Using this update law leads to catastrophic forgetting of previous tasks. To efficiently deploy optimal safe control for the agent across various non-stationary tasks, information from previously executed tasks is required. To address this, we propose a novel dynamic task selection-based adaptive lifelong learning approach to prevent the forgetting of previous tasks efficiently.

Dynamic Sequential Tasks Selection Mechanism: Let, the total number tasks can be selected as M. The goal of the agent $\mathcal{A}$ is to execute these sequential non-stationary tasks while reducing computation in safe optimal tracking control. Next, the actor-critic neural network has been trained for each of the tasks offline. Since there are M tasks, each task has its own activation function, and the set of activation functions can be defined as $\phi_{\mathcal{V}} = \phi_{1,\mathcal{V}}, \phi_{2,\mathcal{V}}, ..., \phi_{M,\mathcal{V}}$. Next, the set of learned weight distribution functions from offline learning is represented as $\rho(\mathcal{W}_{\mathcal{V}}) = \rho_1(\mathcal{W}_{1,\mathcal{V}}), \rho_2(\mathcal{W}_{2,\mathcal{V}}), ..., \rho_M(\mathcal{W}_{M,\mathcal{V}})$. Using this information from offline learning, a novel cost function is defined to select a certain number of tasks that are similar to the current i-th task from the set of M tasks, as described below:

$$\begin{aligned}
J_{\text{tsel}} = w_{ac}[&\mathbb{E}\{\rho_i(\mathcal{W}_{i,\mathcal{V}})\}\phi_{i,\mathcal{V}} - \mathbb{E}\{\rho_j(\mathcal{W}_{j,\mathcal{V}})\}\phi_{j,\mathcal{V}}] + w_d \\
&[\frac{1}{2}\{\int_{-\infty}^{\infty} \rho_i(\mathcal{W}_{i,\mathcal{V}}) \log(\frac{\rho_i(\mathcal{W}_{i,\mathcal{V}})}{\mathcal{M}(\mathcal{W})})dx\} + \frac{1}{2}\{\int_{-\infty}^{\infty} \rho_j \\
&(\mathcal{W}_{j,\mathcal{V}}) \log(\frac{\rho_j(\mathcal{W}_{j,\mathcal{V}})}{\mathcal{M}(\mathcal{W})})dx\}] + w_c F_c
\end{aligned} \tag{17}$$

with, $\mathcal{M}(\mathcal{W}) = \frac{1}{2}(\rho_i(\mathcal{W}_{i,\mathcal{V}}) + \rho_j(\mathcal{W}_{j,\mathcal{V}}))$. Also, j defined the index of the previous task and w_{ac}, w_d, and w_c are the weight terms. Moreover, $\mathbb{E}\{\rho_i(\mathcal{W}_{i,\mathcal{V}})\}$ and $\mathbb{E}\{\rho_j(\mathcal{W}_{j,\mathcal{V}})\}$ represent the mean value of the probability distribution of the weights from task i and j, respectively. The second term is included in the cost function to measure the statistical distance between two probability distributions. We use the well-known KL divergence [20] to measure this distance. Lastly, the final term is incorporated to ensure fairness in task selection, preventing the same task from being selected repeatedly. The function F_c is defined as follows:

$$F_c = \begin{cases} (\frac{S_{j,i}}{\text{task}_i})^2 & \text{if} \quad M_{\text{sel}} \geq i \\ (\frac{S_{j,i}}{\text{task}_M})^2 & \text{if} \quad i > M_{\text{sel}} \end{cases} \tag{18}$$

with, M_{sel} is a predefined maximum selected tasks threshold number and $S_{j,i}$ total instances of the selection of jth task before all sequential tasks of current task i. Also, the term task_i and task_M are defined as follows.

$$\begin{aligned}
\text{task}_i &= (i-1) + (i-2) + ... + (i-i) \\
&= i^2 - (1 + 2 + ... + i)
\end{aligned} \tag{19}$$

$$\begin{aligned}
\text{task}_M &= M_{\text{sel}}^2 - (1 + 2 + ... + M_{\text{sel}}) + (i - 1 - M_{\text{sel}})M_{\text{sel}} \\
&= iM_{\text{sel}} - (1 + 2 + ... + 2M_{\text{sel}})
\end{aligned} \tag{20}$$

Next, a set μ_{sel} is chosen, defining the selected previous tasks based on the minimum values of the cost function presented

in equation (17). Next, the new critic NN cost function for adaptive lifelong learning is defined as follows:

$$E_{L,i} = E_i(\hat{\mathcal{W}}_{i,\mathcal{V}}) + \sum_{j\in\mu_{\text{sel}}}\sum_{p}\frac{1}{2}\lambda\mathcal{F}_{j,p}\|\hat{\mathcal{W}}_{i,\mathcal{V},p} - \hat{\mathcal{W}}_{j,\mathcal{V},p}\|^2 \tag{21}$$

The second term in this new objective function represents the adaptive EWC cost function. Here, λ indicates the importance of the previous task, p represents the parameters of the corresponding weight vectors, and $\mathcal{F}$ denotes the Fisher information matrix [15]. Next, the adaptive lifelong learning-based critic weight update is defined as follows:

$$\dot{\hat{\mathcal{W}}}_{L,i,\mathcal{V}} = -\alpha_{i,\mathcal{V}}\frac{\sigma_{i,\mathcal{V}}}{1+\sigma_{i,\mathcal{V}}^T\sigma_{i,\mathcal{V}}}e_{\text{HJB}}^T - \alpha_{i,\mathcal{V}}\sum_{j\in\mu_{\text{sel}}}\sum_{p}\lambda\mathcal{F}_{j,p}$$
$$\|\hat{\mathcal{W}}_{i,\mathcal{V},p} - \hat{\mathcal{W}}_{j,\mathcal{V},p}\| \tag{22}$$

Next, the ideal function for the actor neural network during the time of ith task execution is defined as follows:

$$u_i(x) = \mathcal{W}_{i,u}^T\phi_{i,u}(x) + \varepsilon_u(x) \tag{23}$$

Here, $\mathcal{W}_{i,u}$ is the actor neural network weight, $\phi_{i,u}(x)$ is the activation function of the actor NN, and $\varepsilon_u(x)$ represents the reconstruction error. Now, by substituting the ideal function into the optimal control equation defined in (8), we obtain the following equation

$$\mathcal{W}_{i,u}^T\phi_{i,u}(x) + \varepsilon_u(x) + \frac{1}{2}R^{-1}g^T(x)\mathcal{W}_{i,u}^T\partial_x\phi_{i,u}(x) = 0 \tag{24}$$

The estimation of the control input is defined as follows:

$$\hat{u}_i(x) = \hat{\mathcal{W}}_{i,u}^T\hat{\phi}_{i,u}(x) \tag{25}$$

The residual error of the actor neural network is now evaluated by inserting the estimated control into the equation (8)

$$e_{i,u} = \hat{\mathcal{W}}_{i,u}^T\hat{\phi}_{i,u}(x) + \frac{1}{2}R^{-1}g^T(x)\hat{\mathcal{W}}_{i,u}^T\partial_x\hat{\phi}_{i,u}(x) \tag{26}$$

The residual error including the reconstruction error is now achieved by combining equations (24) and (26)

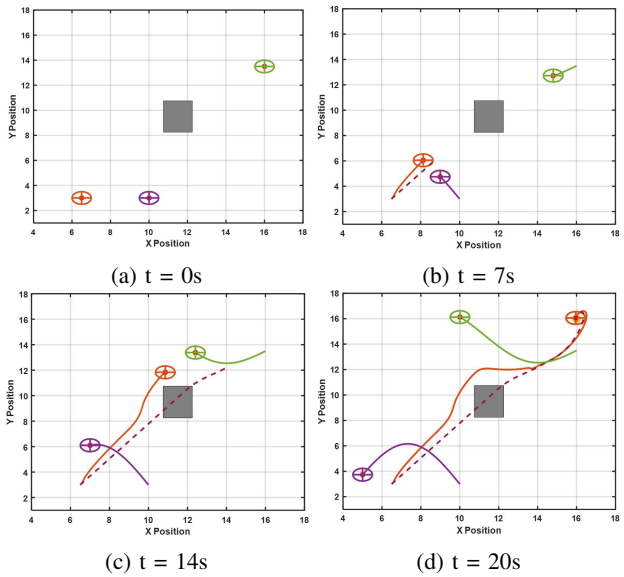
$$e_u = -\tilde{\mathcal{W}}_{i,u}^T\hat{\phi}_{i,u}(x) - \mathcal{W}_{i,u}^T\tilde{\phi}_{i,u}(x) - \frac{1}{2}R^{-1}g^T(x)$$
$$\partial_x\tilde{\mathcal{V}}_i(x,u) - \varepsilon_u(x) \tag{27}$$

Now, the normalized weight update law of the actor neural network is defined as follows:

$$\dot{\hat{\mathcal{W}}}_{i,u} = -\alpha_{i,u}\frac{\hat{\phi}_{i,u}}{1+\hat{\phi}_{i,u}^T\hat{\phi}_{i,u}}e_{i,u}^T \tag{28}$$

IV. Simulation Results

In this section of the simulation study, we implement an adaptive lifelong learning-based safe tracking control algorithm for a local autonomous unmanned vehicle (UAV). This UAV operates in an environment with both static and dynamic obstacles, including other UAVs. The objective of the local UAV is to follow a reference trajectory while completing sequential tasks, ensuring safe navigation by avoiding collisions with dynamic UAVs and stationary obstacles like buildings. The initial state of the local UAV is chosen as $x = \begin{bmatrix}x_1 & x_2\end{bmatrix}^T = \begin{bmatrix}6.5 & 3\end{bmatrix}^T$. Moreover, the initial states of the dynamic external UAV-1 and UAV-2 are selected as $z_1 = \begin{bmatrix}10 & 3\end{bmatrix}^T$ and $z_2 = \begin{bmatrix}16 & 13.5\end{bmatrix}^T$, respectively. Also, a rectangular static obstacle is positioned along the path of the reference trajectory. The center of the obstacle is selected as $z_3 = \begin{bmatrix}11.5 & 9.5\end{bmatrix}^T$ with width$= 1.5m$ and height$= 2.5m$. Next, the local UAV intrinsic dynamic is defined as:



(a) t = 0s (b) t = 7s

(c) t = 14s (d) t = 20s

Fig. 1: The trajectory tracking of the unmanned autonomous vehicle (UAV) in the presence of dynamic and static obstacles. The red dashed line depicts the path of the reference trajectory. The orange curve represents the local UAV trajectory. Also, the magenta and green curves depict the UAV-1 and UAV-2 trajectories, respectively. The circle shapes with different colors represent the UAVs. The static obstacle is represented with a grey-colored rectangular shape. (a) the initial states of the UAVs and obstacles at time $t = 0s$. (b) (c) the UAVs state at time $t = 7s$ and $t = 14s$. (d) The final states of the UAVs at time $t = 20s$.

$$f(x) = \begin{bmatrix}-x_1 + \frac{1}{2}x_2^2 \\ -0.6x_2^2\end{bmatrix} \;;\; g(x) = \begin{bmatrix}0 \\ 1.2\end{bmatrix} \tag{29}$$

Here, $x = \begin{bmatrix}x_1 & x_2\end{bmatrix}^T$. Also, the dynamic functions of the external UAVs are selected as follows:

$$f_o(z_1) = \begin{bmatrix}z_{1,1} - 0.5t \\ z_{2,1} + 3sin(0.3t) + 0.03t\end{bmatrix} \tag{30}$$

and,

$$f_o(z_2) = \begin{bmatrix}z_{1,2} - 0.6t \\ z_{2,2} - 1.5sin(0.4t) + 0.15t\end{bmatrix} \tag{31}$$

with, $z_1 = \begin{bmatrix}z_{1,1} & z_{2,1}\end{bmatrix}^T$ and $z_2 = \begin{bmatrix}z_{1,2} & z_{2,2}\end{bmatrix}^T$.

Now, an actor-critic neural network is trained offline for 10 sequential tasks, with 7 tasks focusing on avoiding dynamic UAVs and 3 tasks dedicated to avoiding static obstacles. The

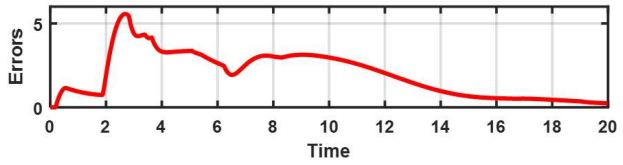


Fig. 2: Tracking error of the local unmanned aerial vehicle (UAV).

learning rate of the critic NN for each task is selected as $\alpha_{\mathcal{V}} = 1 \times 10^{-6}$ and the error threshold is selected as $\delta_{\text{HJB}} = 1 \times 10^{-8}$. Additionally, for the actor neural network, the learning rates and error thresholds are set to $\alpha_u = 1 \times 10^{-5}$ and $\delta_u = 1 \times 10^{-7}$, respectively. Subsequently, the learned weights for all 10 tasks are stored for use in online learning. The parameters of the cost function are selected as $Q = 1$ and $R = 1.5$. Additionally, the activation functions for different tasks in the adaptive lifelong critic neural network are chosen from a polynomial expansion with the formula $\sum_{\beta=1}^{P}(\sum_{j=1}^{n} z_j)^{\beta}$, with n represents the input dimension and P is a constant value. Also, the weight parameters for the task selection cost function are selected as $w_{ac} = 0.4$, $w_d = 0.3$, and $w_c = 0.3$. Again, the learning rate of the adaptive lifelong learning based critic NN for each task is selected as $\alpha_{i,\mathcal{V}} = 1 \times 10^{-3}$ and the error threshold is selected as $\delta_{i,\text{HJB}} = 1 \times 10^{-1}$. Additionally, for the actor neural network, the learning rates and error thresholds are defined as $\alpha_u = 1 \times 10^{-3}$ and $\delta_u = 1 \times 10^{-2}$, respectively. The task selection threshold is chosen as $M_{\text{sel}} = 2$.

Next, a local UAV is deployed in an unstructured environment alongside other UAVs and static obstacle structures. Given the differing nature of dynamic and static obstacles, avoiding these barriers is treated as separate tasks for the UAVs. This mirrors real-world scenarios, as UAVs frequently encounter these challenges in dynamic environments. When faced with a new task, executing it online while ensuring safety and performance can be challenging. However, if the UAV possesses prior knowledge of the tasks, learned from the environment in a manner similar to humans, it can make quick decisions to ensure safety and performance across various tasks.

We illustrate the effectiveness of the developed adaptive lifelong safe learning-based tracking algorithm with a series of figures. The maneuvering of the deployed UAV using the developed algorithm is demonstrated in Figure 1. In this figure, a reference trajectory is provided to the local UAV. Here, the red dashed line represents the curve of the reference trajectory. Also, the local UAV is represented by an orange circle, and its trajectory is illustrated with an orange curve. However, during the time of trajectory tracking, the local UAV encounters other dynamic UAVs and static obstacles. This figure demonstrates how the local UAV performs different tasks by ensuring safety and performance during its trajectory tracking using the adaptive lifelong learning-based non-stationary task execution algorithm. The external UAVs are depicted in magenta and green circles, with their respective trajectories shown in the same colors. Additionally, a static obstacle represented by a grey rectangle is placed along the reference trajectory's path. The initial position of the UAVs and static obstacle is shown in Figure 1(a) at the time $t = 0s$. Then, the motion of the UAVs at time $t = 7s$ is depicted in Figure 1(b). In this figure, the local UAV begins its movement from the left lower corner, which is the starting position of the reference trajectory. Simultaneously, the other external UAVs also start their motion from the different parts of the figure. At that initial stage, the local UAV encounters its first task. From this figure, we can see that the UAV-1 approaches the local UAV. To avoid a collision with the approaching UAV, the local UAV executes its first task by making a slight left turn from the reference trajectory. Then in the figure 1(c) at time $t = 14s$, the second task execution is depicted. Here, the local UAV encounters a static obstacle placed in the path of the reference trajectory. To ensure safety while tracking the trajectory, the local UAV makes a left turn to avoid the static obstacle. Subsequently, UAV-2 enters the path of the local UAV. In figure 1(c) at time $20s$, it is shown that the local UAV successfully executes the third task by avoiding the potential collision with UAV-2 and continues to track the reference trajectory. Since the third task is similar to the first, the UAV performs better in terms of safety and trajectory tracking due to its prior experience with the first task.

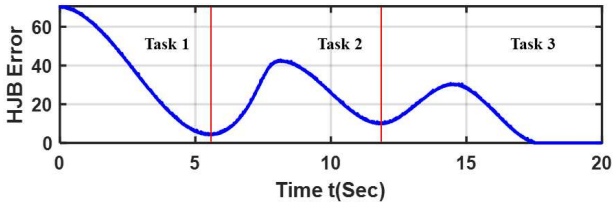


Fig. 3: Adaptive lifelong learning based critic neural network HJB error.

Next, the tracking performance of the local UAV is illustrated in Figure 2. Since the local UAV must track the trajectory while also avoiding collisions, the tracking error increases in certain instances to efficiently execute the given tasks. Specifically, higher tracking errors are observed during the execution of tasks 1 and 2. However, during task 3, as the UAV gains more experience from the previous tasks, the tracking error approaches zero.

Finally, the HJB error from the adaptive lifelong learning-based critic neural network is depicted in Figure 3. This error is shown for the entire simulation duration of 20s. During this period, the local UAV performs three assigned tasks. Initially, the first task assigned to the UAV is to avoid a collision with external UAV-1. At this stage, the local UAV incorporates 2 offline trained weights, similar to the given tasks, using the task selection cost function from equation (17). During the Task-1 period, the UAV's HJB error approaches zero, indicating the successful execution of Task-1. However, it is important to note that the error does not converge to zero perfectly during Task 1. When a new task is assigned,

the HJB error rises. In this period, the UAV avoids a static obstacle and utilizes weights from offline training to recall the relevant task weights. For Task-2, the HJB error also approaches zero. Finally, a new third task is assigned, which is similar to the first task. During this period, the UAV uses weights from both offline and online learning, as a similar instance occurred during the UAV's online maneuvering. Since the UAV has more information related to Task 3 than before, the HJB error converges to zero more efficiently. In summary, the simulation results presented in this section demonstrate the effectiveness of the proposed algorithm.

V. Conclusion

This study has developed a novel adaptive lifelong safe learning-based real-time optimal tracking control algorithm in the presence of an unstructured environment with non-stationary tasks. Here, the optimal problem is formulated with control barrier function (CBF) to ensure the safety and performance of the system. Specifically, the developed algorithm introduces a hybrid offline-online learning framework with a real-time adaptive lifelong safe learning-based actor-critic method. Here, the critic neural network is modified with a dynamic task selection-based adaptive lifelong learning strategy. To adapt to the dynamic environment, the offline neural network weights associated with specific tasks are integrated with a probability distribution and fed into the online algorithm. A novel task selection cost function is provided to compare the relevant previous tasks and efficiently select them to avoid computational complexity. Additionally, a fairness term is incorporated into the algorithm to prevent the repeated selection of previous tasks. Then, the objective function of the critic neural network is reformulated with the selected tasks using the elastic weight consolidation method. Finally, the effectiveness of the algorithm is validated through a series of simulation studies. In the future, more tasks will be added in the online part to validate the efficiency of the proposed algorithm.

References

[1] Y. Liu, Q. Wang, H. Hu, and Y. He, "A novel real-time moving target tracking and path planning system for a quadrotor uav in unknown unstructured outdoor scenes," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2362–2372, 2018.

[2] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini uav," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 89–110, 2008.

[3] C. Yuan, Z. Liu, and Y. Zhang, "Uav-based forest fire detection and tracking using image processing techniques," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2015, pp. 639–643.

[4] S. Dey and H. Xu, "Hierarchical game theoretical distributed adaptive control for large scale multi-group multi-agent system," *IET Control Theory & Applications*, vol. 17, no. 17, pp. 2332–2352, 2023.

[5] T. Dierks and S. Jagannathan, "Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 6750–6755.

[6] S. Dey and H. Xu, "Decentralized adaptive tracking control for large-scale multi-agent systems under unstructured environment," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2022, pp. 900–907.

[7] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[8] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3387–3395.

[9] Z. Marvi and B. Kiumarsi, "Safe reinforcement learning: A control barrier function optimization approach," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 6, pp. 1923–1940, 2021.

[10] M. H. Cohen and C. Belta, "Approximate optimal control for safety-critical systems with control barrier functions," in *2020 59th IEEE conference on decision and control (CDC)*. IEEE, 2020, pp. 2062–2067.

[11] 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*, 2023.

[12] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[13] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural networks*, vol. 113, pp. 54–71, 2019.

[14] Z. Chen and B. Liu, *Lifelong machine learning*. Springer, 2018, vol. 1.

[15] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[16] B. Farzanegan, R. Moghadam, S. Jagannathan, and P. Natarajan, "Optimal adaptive tracking control of partially uncertain nonlinear discrete-time systems using lifelong hybrid learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[17] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[18] S. Dey, H. Xu, and M. S. Fadali, "Adaptive distributed formation control for multi-group large-scale multi-agent systems: A hybrid game approach," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5482–5487, 2023.

[19] S. Dey and H. Xu, "Distributed adaptive flocking control for large-scale multiagent systems," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[20] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 4. IEEE, 2007, pp. IV–317.