

Research Article

Open Access



# Continual online learning-based optimal tracking control of nonlinear strict-feedback systems: application to unmanned aerial vehicles

Irfan Ganie, Sarangapani Jagannathan

Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA.

**Correspondence to:** Irfan Ganie, Electrical and Computer Engineering, Missouri University of Science and Technology, 141 Emerson Electric Co. Hall, 301 W. 16th St., Rolla, MO 65409, USA. E-mail: iag76b@mst.edu

**How to cite this article:** Ganie I, Jagannathan S. Continual online learning-based optimal tracking control of nonlinear strict-feedback systems: application to unmanned aerial vehicles. *Complex Eng Syst* 2024;4:4. <http://dx.doi.org/10.20517/ces.2023.35>

**Received:** 8 Oct 2023 **First Decision:** 6 Dec 2023 **Revised:** 4 Jan 2024 **Accepted:** 23 Jan 2024 **Published:** 20 Feb 2024

**Academic Editors:** Ding Wang, Hamid Reza Karimi **Copy Editor:** Fangling Lan **Production Editor:** Fangling Lan

## Abstract

A novel optimal trajectory tracking scheme is introduced for nonlinear continuous-time systems in strict feedback form with uncertain dynamics by using neural networks (NNs). The method employs an actor-critic-based NN backstepping technique for minimizing a discounted value function along with an identifier to approximate unknown system dynamics that are expressed in augmented form. Novel online weight update laws for the actor and critic NNs are derived by using both the NN identifier and Hamilton-Jacobi-Bellman residual error. A new continual lifelong learning technique utilizing the Fisher Information Matrix via Hamilton-Jacobi-Bellman residual error is introduced to obtain the significance of weights in an online mode to overcome the issue of catastrophic forgetting for NNs, and closed-loop stability is analyzed and demonstrated. The effectiveness of the proposed method is shown in simulation by contrasting the proposed with a recent method from the literature on an underactuated unmanned aerial vehicle, covering both its translational and attitude dynamics.

**Keywords:** Continual lifelong learning, optimal control, neural networks, unmanned aerial vehicles, strict-feedback systems



© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.



## 1. INTRODUCTION

Optimal control of nonlinear dynamical systems with known and uncertain dynamics is an important field of study due to numerous practical applications. Traditional optimal control methods<sup>[1,2]</sup> for nonlinear continuous-time (CT) systems with known dynamics often require the solution to a partial differential equation, referred to as Hamilton-Jacobi-Bellman (HJB) equation, which cannot be solved analytically. To address this challenge, actor-critic designs (ACDs) combined with approximate dynamic programming (ADP) have been proposed as an online method<sup>[3,4]</sup>. Numerous optimal adaptive control (OAC) techniques for nonlinear CT systems using strict-feedback structure have emerged, leveraging backstepping design as outlined in<sup>[5,6]</sup>. These approaches, however, require predefined knowledge of the system dynamics. In real-world industrial settings, where system dynamics might be partially or completely unknown, the application of neural network (NN)-based optimal tracking for uncertain nonlinear CT systems in strict feedback form has been demonstrated in<sup>[5,7]</sup>, utilizing the policy/value iterations associated with ADP. However, these policy/value iteration methods often require an extensive number of iterations within each sampling period to solve the HJB equation and ascertain the optimal control input, leading to a significant computational challenge.

The optimal trajectory tracking of nonlinear CT systems involves obtaining a time-varying feedforward term to ensure precise tracking and a feedback term to stabilize the system dynamics. Recent optimal tracking efforts<sup>[7,8]</sup>, have utilized a backstepping-based approach with completely known or partially unknown system dynamics, but the design of feedforward term while minimizing a cost function has not been addressed. Instead, a linear term is used to design the control input. A more recent study<sup>[8,9]</sup> employed a positive function for obtaining simple weight update laws of the actor and critic NN, which also relaxes the persistency of excitation (PE) condition. However, finding such a function for the time-varying trajectory tracking problem of a nonlinear CT system will be challenging by using an explicit time-dependent value function and HJB equation at each stage of backstepping design since the Hamiltonian is nonzero along the optimal trajectory<sup>[10]</sup>. In simplified and optimized backstepping control schemes were developed for a class of nonlinear strict feedback systems<sup>[8,11,12]</sup>. These approaches are different from the one proposed in<sup>[5]</sup>. However, they either require complete knowledge of the system dynamics or do not assume that the system dynamics are completely unknown.

Moreover, all control techniques rooted in NN-based learning, whether aimed at regulation or tracking, routinely face the issues of catastrophic forgetting<sup>[13]</sup>. This is understood as the system's ability to lose previously acquired knowledge while assimilating new information<sup>[13,14]</sup>. Continual lifelong learning (CLL) is conceived as the sustained ability of a nonlinear system to acquire, assimilate, and retain knowledge over prolonged periods without the interference of catastrophic forgetting. This concept is particularly critical when delving into the realm of online NN control strategies for nonlinear CT systems, as these systems are often tasked with navigating and managing complex processes within dynamic and varying environments and conditions. Nonetheless, the lifelong learning (LL) methodologies shown in<sup>[13,15]</sup> operate in an offline mode and have not been applied to real-time NN control scenarios yet. This scenario offers a promising direction to leverage the advantage of LL in online control systems, addressing catastrophic forgetting and thus enhancing the efficacy of the control system progressively. Implementing LL-oriented strategies in online NN control enables persistent learning and adaptation without discarding prior knowledge, thereby improving its overall performance. By developing an LL-based NN trajectory tracking scheme, it is possible to continuously learn and track trajectories of interest without losing information about previous tasks.

This paper presents an optimal backstepping control approach that incorporates reinforcement learning (RL) to design the controller. The proposed method utilizes an augmented system to address the tracking problem, incorporating both feedforward and feedback controls, which sets it apart from prior work such as<sup>[8,16]</sup>. This approach uses a trajectory generator to generate the trajectories and hence deals with the non-stationary condition in the HJB equation that arises in optimal tracking problems due to the time-varying reference trajectory. In addition, the proposed weight update laws are direct error driven based, obtained using Hamiltonian and

control input error, in contrast to where the weight update laws are obtained using some positive functions<sup>[8,16]</sup>. Furthermore, the control scheme incorporates an identifier where the approximation error is bounded above by system states to approximate the unknown system dynamics, as opposed to prior work, such as<sup>[8,16]</sup>, where the system dynamics are either completely known or partially known. Additionally, the utilization of an HJB equation at each step of the backstepping process is intended to ensure that the entire sequence of steps is optimized.

The paper also examines the impacts of LL and catastrophic forgetting on control systems and proposes strategies for addressing these challenges in control system-based applications. Specifically, the proposed method employs a weight velocity attenuation (WVA)-based LL scheme in an online manner, in contrast to prior work, such as<sup>[13,15]</sup>, which utilizes offline learning. Additionally, the proposed method demonstrates the stability of the LL scheme via Lyapunov analysis in contrast to offline-based learning<sup>[13,15]</sup>, where the weight convergence is not addressed. To validate the effectiveness of the proposed method, an unmanned aerial vehicle (UAV) application is considered, and the proposed method is contrasted with the existing approach. Lyapunov stability shows the uniform ultimate boundedness (UUB) of the overall closed-loop continual lifelong RL (LRL) scheme.

The contributions include

- (1) A novel optimal trajectory tracking control formulation is presented, utilizing an augmented system approach for nonlinear strict-feedback systems within an ADP-based framework, offering a novel perspective.
- (2) An NN-based identifier is employed, wherein the reconstruction error is presumed to be upper-bounded by the norm of the state vector, providing an enhanced approximation of the system dynamics. The new weight update laws are introduced, incorporating Hamiltonian and the NN identifier within an actor-critic framework at each step of the backstepping process.
- (3) An online LL method is developed in the critic NN weight update law, mitigating both catastrophic forgetting and gradient explosion, with the significance of weights for NN layers obtained using Fisher Information Matrix (FIM) determined by the Bellman error, as opposed to offline LL-based methods with targets.
- (4) Lyapunov stability analysis is undertaken for the entire closed-loop tracking system, involving the identifier NN and the LL-based actor-critic NN framework to show the UUB of the closed-loop system.

## 2. CONTINUAL LIFELONG OPTIMAL CONTROL FORMULATION

In this section, we provide the problem formulation and the development of our proposed LRL approach for uncertain nonlinear CT systems in strict feedback form.

### 2.1 System description

Consider the following strict feedback system

$$\begin{aligned}\dot{x}_1(t) &= f_1(x_1) + g_1(x_1)x_2, \\ \dot{x}_2(t) &= f_2(\bar{x}_2) + g_2(\bar{x}_2)u,\end{aligned}\tag{1}$$

where  $\bar{x}_2(t) = [x_1(t), x_2(t)]^T \in \mathbb{R}^2$  is the system state,  $u \in \mathbb{R}$  is the control input, and  $f_i(\cdot)$ ,  $g_i(\cdot)$ ,  $i = 1, 2$  are unknown yet Lipschitz continuous functions on  $\Omega_x$  and satisfying  $f_i(0) = 0$ . The standard following assumptions are stated to proceed.

**Assumption 1** (<sup>[17]</sup>) *The nonlinear CT system is controllable and observable. In addition, the control coefficient matrix satisfies  $\|g_i(\cdot)\| \leq g_{iM}$ , where  $g_{iM}$  is an unknown positive constant.*

**Assumption 2** ([4]) *The state vector is considered measurable. The desired trajectory  $\xi_r \in \mathbb{R}$  is bounded, and there exists a Lipschitz continuous command generator function  $h_d(\xi_r(t))$  such that  $\dot{\xi}_r(t) = h_d(\xi_r(t))$  and  $h_d(0) = 0$ .*

Next, the LRL control design is introduced. The goal of the LRL control scheme is to achieve satisfactory tracking and maintain the boundedness of all closed-loop system signals while minimizing the control effort and addressing the issue of catastrophic forgetting.

The design of the control system begins by implementing an optimal backstepping approach using augmented system-based actor-critic architecture and then using an online LL to mitigate catastrophic forgetting.

## 2.2 Optimal backstepping control

To develop optimal control using the backstepping technique, first, a new augmented system is expressed in terms of tracking error as follows. Define the tracking error as  $e_{tr1}(t) = x_1(t) - \xi_r(t)$ . Taking the time derivative and using the value of  $\dot{x}_1$  to obtain

$$\dot{e}_{tr1}(t) = f_1(x_1) + g_1(x_1)x_2 - h_d(\xi_r(t)), \quad (2)$$

where  $x_2(t)$  is the virtual control.

In order to get both the feedforward and feedback part of the controller, the tracking problem is changed to a regulation problem by defining a new augmented state as  $z_1 = [e_{tr1}^\top \quad \xi_r^\top]^\top$ . Then, we can write

$$\dot{z}_1(t) = \mathcal{F}_{s1}(z_1) + \mathcal{G}_{s1}(z_1)\alpha_1, \quad (3)$$

where  $\mathcal{F}_{s1}(z_1) = \begin{bmatrix} f_1(x_1) - h_d(\xi_r(t)) \\ h_d(\xi_r(t)) \end{bmatrix}$  and  $\mathcal{G}_{s1}(z_1) = \begin{bmatrix} g_1(x_1) \\ 0 \end{bmatrix}$ , with  $z_1(0) = [e_{tr1}^\top(0) \quad \xi_r^\top(0)]^\top$  and  $\alpha_1$  is virtual control. From Assumption 1, it follows that  $0 < \|\mathcal{G}_{si}(\cdot)\| \leq \bar{G}_i$ , where  $\bar{G}_i > 0$  for  $i = 1, 2$ . The design is presented in two steps.

Step 1: For the first backstepping step, let  $\alpha_1$  and  $\alpha_1^*$  be the virtual and optimal virtual control inputs, respectively. The optimal performance index function  $J_1^*(z_1)$  is defined as

$$J_1^*(z_1) = \min_{\alpha_1 \in \Psi(\Omega_{z_1})} \left( \int_t^\infty e^{-\gamma_1(s-t)} h_1(z_1, \alpha_1) ds \right) = \int_t^\infty e^{-\gamma_1(s-t)} h_1(z_1, \alpha_1^*) ds, \quad (4)$$

where  $\Psi(\Omega_{z_1})$  denotes the set of admissible control policies over a compact set  $\Omega_{z_1}$ ,  $J_1(z_1) = \left( \int_t^\infty e^{-\gamma_1(s-t)} h_1(z_1, \alpha_1) ds \right)$ , and  $h_1(z_1, \alpha_1) = z_1(s)^\top q_1 z_1(s) + \alpha_1(z_1)^\top r_1 \alpha_1(z_1)$  with  $q_1$  is a positive definite,  $r_1 > 0$ ,  $\gamma_1$  is the discount factor,  $\gamma_1 > 0$ .

**Remark 1** *Generally, addressing trajectory tracking control problems poses considerable challenges, particularly when dealing with a system characterized by nonlinear dynamics and a trajectory that evolves over time. In such instances, a prevalent approach is to employ a discounted cost function, denoted as (4), to render the cost index,  $J_1^*$ , finite. The control input, consisting of both feedforward and feedback components, is obtained simultaneously by minimizing the performance function (4) along the trajectories of the augmented system (3). In addition, the performance function is not explicitly dependent on time.*

By taking the time derivative on both sides of the optimal performance function (4), the tracking Bellman equation is obtained as

$$j_1^*(z_1) = \int_t^\infty \frac{\partial}{\partial t} e^{-\gamma_1(s-t)} h_1(z_1, \alpha_1^*) ds - h_1(z_1, \alpha_1^*) \tag{5}$$

By noting that the first term of (5) is  $\gamma_1 J_1$ , therefore, (5) can be rewritten as

$$h_1(z_1, \alpha_1^*) + \frac{dJ_1^{*\top}(z_1)}{dz_1} \dot{z}_1(t) - \gamma_1 J_1^* = 0. \tag{6}$$

Therefore, the tracking HJB equation is generated as

$$\begin{aligned} H_1\left(z_1, \alpha_1^*, \frac{dJ_1^*}{dz_1}\right) &= h_1(z_1, \alpha_1^*) + \frac{dJ_1^{*\top}(z_1)}{dz_1} \dot{z}_1(t) - \gamma_1 J_1^* \\ &= \frac{dJ_1^{*\top}(z_1)}{dz_1} (\mathcal{F}_{s1}(z_1) + \mathcal{G}_{s1}(z_1)\alpha_1^*) + Q_1 + \bar{\alpha}_1 - \gamma_1 J_1^* = 0. \end{aligned} \tag{7}$$

where  $H_1\left(z_1, \alpha_1^*, \frac{dJ_1^*}{dz_1}\right)$  is the Hamiltonian function for the first step,  $Q_1 = z_1(s)^\top q_1 z_1(s)$ ,  $\bar{\alpha}_1 = \alpha_1^{*\top}(z_1) r_1 \alpha_1^*(z_1)$ . The optimal control solution  $\alpha_1^*$  can be obtained by solving  $\frac{\partial H_1}{\partial \alpha_1^*} = 0$ . This equation represents the condition for finding the optimal control that minimizes the performance index and satisfies the HJB equation as

$$\alpha_1^* = -\frac{1}{2} r_1^{-1} \mathcal{G}_{s1}^\top(z_1) \frac{dJ_1^*(z_1)}{dz_1}. \tag{8}$$

It is well known that NNs have universal function approximation abilities and can approximate a nonlinear continuous function  $\mathcal{H}(z) : \mathbb{R}^n \rightarrow \mathbb{R}^m$  on a compact set  $\Omega_z$  as  $\mathcal{H}(z) = W^\top \sigma(z)$ , where  $W \in \mathbb{R}^{p \times m}$  is the weight matrix,  $p$  is the number of neurons, and  $\sigma(z)$  is the basis function vector.

Since  $J_1^*$  is unknown, an NN and its derivative will be used to approximate it on a compact set as

$$J_1^* = W_{c1}^{*\top} \sigma_{c1}(z_1) + \varepsilon_{c1}(z_1) \tag{9}$$

$$\frac{dJ_1^*(z_1)}{dz_1} = \nabla \sigma_{c1}^\top(z_1) W_{c1}^* + \nabla \varepsilon_{c1}^\top(z_1), \tag{10}$$

where  $W_{c1}^*$ ,  $\sigma_{c1}$ , and  $\varepsilon_{c1}$  are the target weights, basis function vector, and the NN reconstruction error, respectively,  $\nabla \sigma_{c1}$  and  $\nabla \varepsilon_{c1}$  are the partial derivative of  $\sigma_{c1}$  and  $\varepsilon_{c1}$ , respectively, with respect to input  $z_1$ . Substituting (8) and (9) into the HJB equation (6) to get

$$\begin{aligned} H_1^* &= \frac{1}{4} (W_{c1}^{*\top} \nabla \sigma_{c1} + \nabla \varepsilon_{c1}) \wedge_1 (\nabla \sigma_{c1}^\top W_{c1}^* + \nabla \varepsilon_{c1}^\top) - \gamma_1 (W_{c1}^{*\top} \sigma_{c1}(z_1) + \varepsilon_{c1}) \\ &+ (W_{c1}^{*\top} \nabla \sigma_{c1}(z_1) + \nabla \varepsilon_{c1}) \times \left( \mathcal{F}_{s1}(z_1) - \left( \frac{1}{2} \wedge_1 (\nabla \sigma_{c1}^\top(z_1) W_{c1}^* + \nabla \varepsilon_{c1}^\top) \right) \right) + Q_1, \end{aligned} \tag{11}$$

where optimal Hamiltonian function  $H_1^* = H_1\left(z_1, \alpha_1^*, \frac{dJ_1^*}{dz_1}\right)$ , and  $\wedge_1 = \mathcal{G}_{s1} r_1^{-1} \mathcal{G}_{s1}^\top$ . Since the target weight matrix,  $W_{c1}^*$ , is unknown, an actor-critic NN and its derivative will be designed to find the solution as follows

$$\begin{aligned} \hat{J}_1 &= \hat{W}_{c1}^\top(t) \sigma_{c1}(z_1) \\ \frac{d\hat{J}_1(z_1)}{dz_1} &= \nabla \sigma_{c1}^\top(z_1) \hat{W}_{c1}, \end{aligned} \tag{12}$$

where  $\hat{W}_{c1}$  is the estimate of the weight matrices for critic NN, and  $\sigma_{c1}(z_1)$  is the basis function vector. Next, using (12), we can rewrite (11) as

$$\hat{H}_1 = \frac{1}{4} \hat{W}_{c1}^\top \nabla \sigma_{c1} \hat{\lambda}_1 \nabla \sigma_{c1}^\top \hat{W}_{c1} - \gamma_1 \hat{W}_{c1}^\top \sigma_{c1}(z_1) + (\hat{W}_{c1}^\top \nabla \sigma_{c1}) \left( \hat{\mathcal{F}}_{s1}(z_1) - \left( \frac{1}{2} \hat{\lambda}_1 \nabla \sigma_{c1}^\top \hat{W}_{c1} \right) \right) + \mathcal{Q}_1, \quad (13)$$

where the estimated Hamiltonian function  $\hat{H}_1 = H_1 \left( z_1, \hat{\alpha}_1, \frac{dJ_1}{dz_1} \right)$ ,  $\hat{\lambda}_1 = \hat{\mathcal{G}}_{s1} r_1^{-1} \hat{\mathcal{G}}_{s1}^\top$  and  $\hat{\mathcal{F}}_{s1}(z_1)$ ,  $\hat{\mathcal{G}}_{s1}$  are the estimate of augmented internal dynamics and input dynamics to be generated in the subsequent section by using an NN identifier. From (8),  $\alpha_1$  is unknown, continuous on a compact set  $\Omega_{z_1}$ , and an NN is used to approximate it over a compact set, as shown in the subsequent steps. We can write

$$-\frac{1}{2} r_1^{-1} \mathcal{G}_{s1}^\top(z_1) \frac{dJ_1^*(z_1)}{dz_1} = -\frac{1}{2} \mathcal{V}_1^*(z_1), \quad (14)$$

where  $\mathcal{V}_1^*(z_1) = r_1^{-1} \mathcal{G}_{s1}^\top(z_1) \frac{dJ_1^*(z_1)}{dz_1}$ . where  $\mathcal{G}_{s1}(z_1) \in \mathbb{R}^2$ , and  $\frac{dJ_1^*(z_1)}{dz_1} \in \mathbb{R}^2$  which makes  $\mathcal{V}_1^*(z_1)$  a scalar. To achieve the tracking, decompose the term  $\mathcal{V}_1^*$  into following two parts

$$\mathcal{V}_1^* = 2\beta_1 z_1 + \mathcal{V}_1^0, \quad (15)$$

where  $\mathcal{V}_1^0 = -2\beta_1 z_1 + \mathcal{V}_1^*$ , where  $\beta_1$  is a  $[\bar{\beta}_1, 0]$  i.e  $\beta_1 \in \mathbb{R}^{1 \times 2}$ ,  $\bar{\beta}_1 > 0$ ,  $z_1 \in \mathbb{R}^2$ . The optimal virtual controller  $\alpha_1^*$  can be written as

$$\alpha_1^* = -\beta_1 z_1 - \frac{1}{2} \mathcal{V}_1^0. \quad (16)$$

Therefore,  $\mathcal{V}_1^0$  can be approximated using NN,

$$r_1^{-1} \mathcal{G}_{s1}^\top(z_1) \frac{dJ_1^*(z_1)}{dz_1} = 2\beta_1 z_1 + (W_{a1}^{*\top} \sigma_{a1}(z_1) + \varepsilon_1), \quad (17)$$

where  $W_{a1}^* \in \mathbb{R}^{p \times m}$  are the target bounded unknown weights, and  $\varepsilon_1 \in \mathbb{R}^m$  denotes the function reconstruction error;  $\sigma_{a1}(z_1)$  with  $z_1 \in \Omega_{z_1}$  is the basis function vector. Hence, the virtual controller (8) can be written as

$$\alpha_1^* = -\beta_1 z_1 - \frac{1}{2} (W_{a1}^{*\top} \sigma_{a1}(z_1) + \varepsilon_1(z_1)), \quad (18)$$

where  $W_{a1}^* \in \mathbb{R}^{p \times m}$  are the target bounded unknown weights, and  $\varepsilon_1 \in \mathbb{R}^m$  denotes the function reconstruction error;  $\sigma_{a1}(z_1)$  with  $z_1 \in \Omega_{z_1}$  is the basis function vector. Since the target weights are unknown, the actor NN will be designed to estimate the optimal virtual control as

$$\hat{\alpha}_1 = -\beta_1 z_1 - \frac{1}{2} \hat{W}_{a1}^\top \sigma_{a1}(z_1), \quad (19)$$

where  $\hat{W}_{a1}$ ,  $\sigma_{a1}(z_1)$ ,  $z_1 \in \Omega_{z_1}$  are the estimated weights and the basis function of actor NN. According to the HJB equation (11) and its approximation (13), define the HJB residual error  $e_1(t)$  as

$$e_1(t) = \hat{H}_1 - H_1^* = \hat{H}_1, \quad (20)$$

since the optimal Hamiltonian value is zero. Notice that the estimated Hamiltonian,  $\hat{H}_1$ , requires the estimate of unknown dynamics from the identifier. The second step is discussed next.

**Remark 2** The subsequent section will leverage the HJB residual error (20) to formulate the weight update laws for the critic NN. Additionally, the control input error  $u_e$ , delineated in Lemma 1, will facilitate the derivation of the actor NN weight update laws.

Employing the HJB residual error and the control input error for formulating the weight update laws is pivotal as it enables efficient optimization of the weights in NN. This methodology ensures more accurate and reliable learning processes, allowing the network to better approximate the desired functions or policies, thereby enhancing the overall performance and robustness of the system.

**Step 2:** This is the second step, and the actual controller  $\hat{u}$  will be derived. Define  $e_{ir2}(t) = x_2(t) - \hat{\alpha}_1$ ; then using (1), the error dynamics are written as

$$\dot{e}_{ir2}(t) = f_2(\bar{x}_2) + g_2(\bar{x}_2)u - \dot{\hat{\alpha}}_1. \tag{21}$$

Since  $\hat{\alpha}_1 = -\beta_1 z_1 - \hat{W}_{a1}^\top \sigma_{z_1}$ , on using the values from step 1,  $\dot{\hat{\alpha}}_1$  can be calculated by using a virtual generator  $h_d(\hat{\alpha}_1)$ . Let  $z_2 = [e_{ir2}^\top, \hat{\alpha}_1^\top]^\top \in \mathbb{R}^2$ . One can write

$$\dot{z}_2(t) = \mathcal{F}_{s2}(z_2) + \mathcal{G}_{s2}(z_2)u, \tag{22}$$

where  $\mathcal{F}_{s2}(z_2) = \begin{bmatrix} f_2(\bar{x}_2) - h_d(\hat{\alpha}_1) \\ h_d(\hat{\alpha}_1) \end{bmatrix}$  and  $\mathcal{G}_{s2}(z_2) = \begin{bmatrix} g_2(\bar{x}_2) \\ 0 \end{bmatrix}$ . Letting  $u^*$  be the optimal control, the optimal integral cost function is defined as

$$J_2^*(z_2) = \min_{u \in \Psi(\Omega_{z2})} \left( \int_t^\infty e^{-\gamma_2(s-t)} h_2(z_2(s), u(z_2)) ds \right) = \int_t^\infty e^{-\gamma_2(s-t)} h_2(z_2(s), u^*(z_2)) ds, \tag{23}$$

where  $h_2 = z_2(s)^\top q_2 z_2(s) + u^*(z_2)^\top r_2 u^*(z_2)$  is the cost function. The user-defined penalties,  $q_2$  and  $r_2$ , are positive definite and  $\gamma_2 > 0$ .

The HJB equation for step 2 is given by

$$H_2 \left( z_2, u^*, \frac{dJ_2^*}{dz_2} \right) = Q_2 + \bar{u} + \frac{dJ_2^{*\top}}{dz_2} (\mathcal{F}_{s2}(z_2) + \mathcal{G}_{s2}(z_2)u^*) - \gamma_2 J_2^* = 0, \tag{24}$$

where  $H_2 \left( z_2, u^*, \frac{dJ_2^*}{dz_2} \right)$  is the Hamiltonian function for the second step,  $Q_2 = z_2(s)^\top q_2 z_2(s)$ ,  $\bar{u} = u^*(z_2)^\top r_2 u^*(z_2)$ . Similar to previous steps, solving  $(\partial H_2 / \partial u^*) = 0$  yields

$$u^* = -\frac{1}{2} r_2^{-1} \mathcal{G}_{s2}^\top(z_2) \frac{dJ_2^*}{dz_2}. \tag{25}$$

Since  $J_2^*$  is unknown, an NN will be used to approximate it on a compact set given by

$$J_2^* = W_{c2}^{*\top} \sigma_{c2}(z_2) + \varepsilon_{c2}(z_2). \tag{26}$$

Therefore,  $\frac{dJ_2^*(z_2)}{dz_2} = \nabla \sigma_{c2}^\top(z_2) W_{c2}^* + \nabla \varepsilon_{c2}^\top$ , where  $W_{c2}^*$ ,  $\sigma_{c2}$ ,  $\varepsilon_{c2}$  are the NN weights, activation function, and the reconstruction error, respectively. Substituting (25) and (26) into the HJB equation (24) gets

$$\begin{aligned} H_2^* &= \frac{1}{4} (W_{c2}^{*\top} \nabla \sigma_{c2} + \nabla \varepsilon_{c2}) \wedge_2 (\nabla \sigma_{c2}^\top W_{c2}^* + \nabla \varepsilon_{c2}^\top) + Q_2 - \gamma_2 (W_{c2}^{*\top} \sigma_{c2}(z_2) + \varepsilon_{c2}) \\ &+ (W_{c2}^{*\top} \nabla \sigma_{c2}(z_2) + \nabla \varepsilon_{c2}) \left( \mathcal{F}_{s2}(z_2) - \frac{1}{2} \wedge_2 (\nabla \sigma_{c2}^\top(z_2) W_{c2}^* + \nabla \varepsilon_{c2}^\top) \right), \end{aligned} \tag{27}$$

where  $H_2^* = H_2 \left( z_2, \alpha_2^*, \frac{dJ_2^*}{dz_2} \right)$  is the optimal Hamiltonian for the second step,  $\Lambda_2 = \mathcal{G}_{s2} r_2^{-1} \mathcal{G}_{s2}^\top$ . Since the target weight matrix,  $W_{c2}^*$ , is unknown, an actor-critic NN will be designed to find the solution as follows

$$\begin{aligned} \hat{J}_2 &= \hat{W}_{c2}^\top(t) \sigma_{c2}(z_2) \\ \frac{d\hat{J}_2(z_2)}{dz_2} &= \nabla \sigma_{c2}^\top(z_2) \hat{W}_{c2}, \end{aligned} \quad (28)$$

where  $\hat{W}_{c2}$  is the estimate of the weight matrices for critic NN, and  $\sigma_{c2}(z_2)$ ,  $z_2 \in \Omega_{z2}$  is the basis function vector. Next, we can rewrite (27) as follows

$$\hat{H}_2 = \frac{1}{4} \hat{W}_{c2}^\top \nabla \sigma_{c2} \hat{\Lambda}_2 \nabla \sigma_{c2}^\top \hat{W}_{c2} + Q_2 - \gamma_2 \hat{W}_{c2}^\top \sigma_{c2}(z_2) + (\hat{W}_{c2}^\top \nabla \sigma_{c2}) \left( \hat{\mathcal{F}}_{s2}(z_2) - \left( \frac{1}{2} \hat{\Lambda}_2 \nabla \sigma_{c2}^\top \hat{W}_{c2} \right) \right), \quad (29)$$

where  $\hat{H}_2 = H_2 \left( z_2, \hat{\alpha}_2, \frac{d\hat{J}_2}{dz_2} \right)$  is the estimated Hamiltonian,  $\hat{\Lambda}_2 = \hat{\mathcal{G}}_{s2} r_2^{-1} \hat{\mathcal{G}}_{s2}^\top$  and  $\hat{\mathcal{F}}_{s2}(z_2)$ ,  $\hat{\mathcal{G}}_{s2}$  are the estimate of augmented internal and input dynamics to be generated in the subsequent section by using the NN identifier.

Since  $\mathcal{G}_{s2}(z_2)$  and  $\frac{dJ_2^*(z_2)}{dz_2}$  are unknown, continuous on a compact set  $\Omega_{z2}$ , an NN is used to approximate them over a compact set, we can write

$$-\frac{1}{2} r_2^{-1} \mathcal{G}_{s2}^\top(z_2) \frac{dJ_2^*(z_2)}{dz_2} = -\frac{1}{2} \mathcal{V}_2^*(z_2), \quad (30)$$

where  $\mathcal{V}_2^*(z_2) = r_2^{-1} \mathcal{G}_{s2}^\top(z_2) \frac{dJ_2^*(z_2)}{dz_2}$ . To achieve tracking, decompose the term  $\mathcal{V}_2^*$  into following two parts

$$\mathcal{V}_2^* = 2\beta_2 z_2 + \mathcal{V}_2^0, \quad (31)$$

where  $\mathcal{V}_2^0 = -2\beta_2 z_2 + \mathcal{V}_2^*$ ,  $\beta_2 = [\bar{\beta}_2, 0]$  i.e  $\beta_2 \in \mathbb{R}^{1 \times 2}$ ,  $\bar{\beta}_2 > 0$ ,  $z_2 \in \mathbb{R}^2$ . The optimal controller  $u^*$  can be written as  $u^* = -\beta_2 z_2 - \frac{1}{2} \mathcal{V}_2^0$ . Therefore, we can write

$$r_2^{-1} \frac{\mathcal{G}_{s2}^\top(z_2)}{2} \frac{dJ_2^*(z_2)}{dz_2} = 2\beta_2 z_2 + (W_{a2}^{*\top} \sigma(z_2) + \varepsilon_2), \quad (32)$$

where  $W_{a2}^* \in \mathbb{R}^{p \times m}$  are the target bounded unknown weights, and  $\varepsilon_2 \in \mathbb{R}^m$  denotes the function reconstruction error;  $\sigma(z_2)$  with  $z_2 \in \Omega_{z2}$  is the basis function vector. Therefore, the actual controller (16) can be written as

$$u^* = -\beta_2 z_2 - \frac{1}{2} (W_{a2}^{*\top} \sigma_{a2}(z_2) + \varepsilon_2(z_2)). \quad (33)$$

Similarly, the actor NN will be designed to estimate the optimal control as

$$\hat{u} = -\beta_2 z_2 - \frac{1}{2} \hat{W}_{a2}^\top \sigma_{a2}(z_2), \quad (34)$$

where  $\hat{W}_{a2}$ ,  $\sigma_{a2}(z_2)$ ,  $z_2 \in \Omega_{z2}$  are the estimated weights and the basis function of actor NN. According to the HJB equation (27) and its approximation (29), define the HJB residual error  $e_2(t)$  as follows

$$e_2(t) = \hat{H}_2 - H_2^* = \hat{H}_2, \quad (35)$$

since the optimal Hamiltonian for the second step is zero, notice that the estimated Hamiltonian,  $\hat{H}_2$ , requires the estimate of unknown dynamics from the identifier. Similar to the two steps, the proposed method can be extended to  $n$ th order.

**Remark 3** The optimal control input can be obtained by utilizing the gradient of the optimal value function (26) and an NN identifier. As a result, the critic NN outlined in (26) may be utilized to determine the actor input without the need for an additional NN. However, for the purpose of simplifying the derivation of weight update rules and subsequent stability analysis, separate NNs are employed for the actor and critic.

Next, an NN identifier will be used to approximate the unknown dynamics given by (3) and (22).

### 2.3 NN identifier

A single-layer NN is used to approximate both the nonlinear functions  $\mathcal{F}_{sj}$  and  $\mathcal{G}_{sj}$  where  $j = 1, 2$ , that govern the augmented dynamics<sup>[17]</sup>. Then, by using  $\hat{\mathcal{G}}_{sj}$ , which is obtained from the NN identifier, the estimated control policy is applied to the nonlinear system. The approximations are represented as  $\mathcal{F}_{sj}(z_j) = V_{F_j}^\top \sigma_{F_j}(z_j) + \varepsilon_{F_j}(z_j)$  and  $\mathcal{G}_{sj}(z_j) = V_{G_j}^\top \sigma_{G_j}(z_j) + \varepsilon_{G_j}(z_j)$ , where  $V_{F_j}$  and  $V_{G_j}$  are the NN's weight matrices,  $\sigma_{F_j}(z_j)$  and  $\sigma_{G_j}(z_j)$  are the activation functions, and  $\varepsilon_{F_j}(z_j)$  and  $\varepsilon_{G_j}(z_j)$  are the NN's reconstruction errors respectively. The estimated values of the internal dynamics and control coefficient matrix of the augmented system are given by  $\hat{\mathcal{F}}_{sj}(z_j) = \hat{V}_{F_j}^\top \sigma_{F_j}(z_j)$  and  $\hat{\mathcal{G}}_{sj}(z_j) = \hat{V}_{G_j}^\top \sigma_{G_j}(z_j)$ , where  $\hat{V}_{F_j}$  and  $\hat{V}_{G_j}$  are the estimated NN's weight matrices. Define

$$\dot{z}_j(t) = Z_j^\top \sigma(\xi_j) \bar{\alpha}_j + \varepsilon_I(z_j), \tag{36}$$

where  $\bar{\alpha}_j = [1, \alpha_j]^\top$ , is the augmented control input,  $\alpha_j$  is given by (19), (34),  $Z_j = \begin{bmatrix} V_{F_j}^\top & V_{G_j}^\top \end{bmatrix}^\top \in \mathbb{R}^{2l \times n}$  represents the augmented NN identifier weights, and  $\sigma(\xi_j) = \text{diag}\{\sigma_{F_j}(z_j), \sigma_{G_j}(z_j)\}$  denotes the augmented activation function for the NN identifier. The reconstruction error of the NN identifier, denoted by  $\varepsilon_I(z_j)$ , is defined as  $\varepsilon_I(z_j) = (\varepsilon_{F_j}(z_j) + \varepsilon_{G_j}(z_j)\alpha_j)$ . Next, the following assumption is stated.

**Assumption 3** <sup>[17]</sup> The NN identifier is of single-layer, and its reconstruction error is bounded above such that  $\|\varepsilon_I(z_1)\|^2 \leq b_0 \|z_1\|^2$  and  $\|Z_1\| \leq Z_{m_1}$ .

**Remark 4** Because  $\varepsilon_I(z_j)$  depends on input  $\alpha_j$  and the system state  $z_j(t)$ , it is assumed to be bounded above by the norm of the state vector unlike<sup>[18]</sup>, where  $\varepsilon_I(z_j)$  is bounded by a constant value.

Define the dynamics of the NN identifier as

$$\dot{\hat{z}}_j(t) = \hat{Z}_j^\top \sigma(\xi_j) \bar{\alpha}_j + K(z_j - \hat{z}_j), \tag{37}$$

where  $\hat{z}_j(t)$  represents the estimated augmented state vector,  $K$  is a user defined constant gain matrix, and  $\hat{Z}_j = \begin{bmatrix} \hat{V}_{F_j}^\top & \hat{V}_{G_j}^\top \end{bmatrix}^\top$  represents the augmented NN identifier estimated weights. The state estimation error is defined as  $e_{i_j} = z_j - \hat{z}_j$ . The weight update law for the NN identifier is given by

$$\dot{\hat{Z}}_j = -\alpha_{vj} \hat{Z}_j + \sigma(\xi_j) \bar{\alpha}_j e_{i_j}^\top, \tag{38}$$

where  $\alpha_{vj} > 0$  is a tuning parameter.

**Remark 5** The NN identifier weights are tuned by using both augmented state estimation error and the input vector. The boundedness of the control input is needed to show the convergence of the NN identifier if proof is shown separately, whereas this assumption is relaxed when the identifier is combined with the LRL control scheme, as shown in the next section.

**Remark 6** Since the control input error and the HJB error used to tune the actor-critic NN weights require the system dynamics which are uncertain, the NN identifier is used to approximate the unknown dynamics of an

augmented system. The estimated values from the identifier are used in the actor-critic weight update laws to tune the NN weights, as shown in the subsequent section.

## 2.4 Actor critic NN weight tuning

In this section, the actor-critic weight update laws are obtained using the gradient descent method to the Hamiltonian-based performance function. The following Lemma is stated.

**Lemma 1** Consider a system (1), transformed system (3), NN identifier weight update laws (38), the update laws for the critic NN (12), (26) and actor NN (19), (34). They can be written as

$$\dot{\hat{W}}_{cj} = \gamma_{cj} \hat{\psi}_j(t) \left( e_j^\top(t) - \frac{1}{4} \hat{W}_{cj}^\top \nabla \sigma_{cj} \hat{\lambda}_j \nabla \sigma_{cj}^\top \hat{W}_{cj} - \hat{W}_{cj}^\top \nabla \sigma_{cj} \hat{\mathcal{F}}_{sj} \right) - \sigma_{1j} \hat{W}_{cj} \quad (39)$$

$$\dot{\hat{W}}_{aj}(t) = \beta_{aj} (S_{aj} u_{ej}^\top) - \sigma_{2j} \hat{W}_{aj} \quad (40)$$

where  $j$  denotes the number of steps in backstepping,  $\gamma_{cj} = \frac{\beta_{cj} \hat{\psi}_j}{\|\hat{\psi}_j\|^2 + 1}$ ,  $\beta_{cj}, \beta_j, \beta_{aj} > 0$  are the learning rates,  $S_{aj} = \sigma_{aj}(z_j)$ ,  $S_{cj} = \sigma_{cj}(z_j)$ , and  $\hat{\psi}_j(t) = \gamma_j S_{cj} - \nabla \sigma_{cj} (\hat{\mathcal{F}}_{sj} + \hat{\mathcal{G}}_{sj} \hat{\alpha}_j)$ , where  $\sigma_{1j} > 0, \sigma_{2j} > 0$  are the design constants,  $u_{ej}$  is the error between estimated control and actual control and is given by  $u_{ej} = -\beta_j z_j - \frac{1}{2} \hat{W}_{aj}^\top \sigma_{aj}(z_j) + \frac{1}{2} r_j^{-1} \hat{\mathcal{G}}_{sj}^\top \frac{d\hat{j}_j}{dz_j}$ ,  $e_j$  is the HJB error for each step given by (20), (35),  $\hat{\lambda}_j = \hat{\mathcal{G}}_{sj} r_j^{-1} \hat{\mathcal{G}}_{sj}^\top$ , and  $\hat{\mathcal{F}}_{sj}, \hat{\mathcal{G}}_{sj}$  are approximated by using NN identifier.

**Proof:** The weight update laws for critic NN in step  $j$  are obtained by defining the performance function as

$$E_j = \frac{1}{2} e_j^2. \quad (41)$$

By using the gradient descent algorithm, the weight update law can be obtained as

$$\dot{\hat{W}}_{cj} = - \frac{\beta_{cj} \hat{\psi}_j}{(1 + \hat{\psi}_j^\top \hat{\psi}_j)} \frac{\partial E_j(t)}{\partial \hat{W}_{cj}}. \quad (42)$$

On simplifying (42), we will get the weight update law for critic NN, as shown in Lemma. The weight update law for actor NN is obtained by defining the performance function as

$$E_{aj} = \frac{1}{2} u_{ej}^2. \quad (43)$$

By using the gradient descent approach, the weight update law for an actor NN is obtained as

$$\dot{\hat{W}}_{aj}(t) = - \frac{\partial E_{aj}(t)}{\partial \hat{W}_{aj}}. \quad (44)$$

On further solving and adding the stabilization terms, we will get the weight update law shown in Lemma 1.

**Remark 7** . The weight update laws are obtained using the gradient descent method to the Hamiltonian-based performance function. The weight update equations for the critic and actor have additional terms to ensure stability and facilitate convergence proof. The last term, known as the sigma modification term, relaxes the PE condition needed to ensure weight convergence. It is important to note that the right-hand side terms in the weight update equation can be measured.

The following assumption is stated next.

**Assumption 4** ([4]) *It is assumed that the ideal weights exist and are bounded over a compact set by an unknown positive constant, such that  $\|W_{c_j}^*\| \leq \bar{W}_{M1j}$ ,  $\|W_{a_j}^*\| \leq \bar{W}_{M2j}$  where  $\bar{W}_{M1j}$ ,  $\bar{W}_{M2j}$  are unknown constant values. The basis function  $\sigma(\cdot)$ , function reconstruction error  $\varepsilon_j(\cdot)$ , and their derivatives with respect to their arguments are assumed to be bounded over a compact set, with unknown bounds as  $\|\sigma\| \leq \bar{\sigma}$ ,  $\|\dot{\sigma}\| \leq \bar{\dot{\sigma}}$ ,  $\varepsilon_j(\cdot) \leq \varepsilon_m$ ,  $\dot{\varepsilon}_j(\cdot) \leq \bar{\varepsilon}_m$ .*

Next, the following theorem is stated.

**Theorem 1** *Consider the nonlinear system in strict-feedback form defined by (1). By using the augmented system (3), consider the optimal virtual control (19) and actual control terms (34), along with the identifier, actor-critic updating laws (38), (39), and (40). Further, assume that the design parameters are selected as per the conditions stated and that Assumptions 1 through 4 hold. If the system input is PE and its initial value,  $u_0$ , is admissible, then tracking errors, augmented state  $z_j$ , the actual state  $x_j$  and NN weight estimation errors,  $\tilde{Z}_j = Z_j - \hat{Z}_j$ ,  $\tilde{W}_{c_j} = W_{c_j} - \hat{W}_{c_j}$ , and  $\tilde{W}_{a_j} = W_{a_j} - \hat{W}_{a_j}$  are guaranteed to be bounded. This, in turn, ensures that the actual system output tracks the desired trajectory and estimated/actual control inputs are bounded close to their optimal values.*

**Proof:** See Appendix.

**Remark 8** *In the proposed optimal backstepping technique, the RL/ADP is employed at every step to obtain the optimal virtual and actual control inputs. We have derived the backstepping for a two-step process; however, it can be implemented up to  $n$  steps using a similar procedure.*

**Remark 9** *The sigma modification term does serve to alleviate the PE condition and assists in the process of forgetting; however, it does not prove effective in multitasking scenarios to minimize forgetting. Subsequently, a novel online LL strategy is presented to address the issue of catastrophic forgetting.*

Next, an online regularization-based approach to LL is introduced.

### 2.5 Continual lifelong learning

To mitigate the issues of catastrophic forgetting<sup>[13]</sup>, a novel technique called WVA was proposed<sup>[15]</sup>. However, WVA has only been used in an offline manner, which cannot be applied to NN-based online techniques.

In contrast, this study introduces a new online LL technique that can be integrated into an online NN-based trajectory tracking control scheme by identifying and safeguarding the most critical parameters during the optimization process. To achieve this, the proposed technique employs a performance function given by

$$L_j(\hat{W}_{c_j}) \approx E_{bj} + \frac{\lambda_j}{2} \left\| \hat{W}_{c_j} - \hat{W}_{c_{jp}}^* \right\|_{\bar{\Omega}_{kj}}^2, \tag{45}$$

where  $E_{bj}$  is the loss function for the current task  $B$  in step  $j$  (41),  $\bar{\Omega}_{kj} = \text{diag}\{\frac{\Omega_{w1}}{\Omega_{w1+1}} \dots \frac{\Omega_{wn}}{\Omega_{wn+1}}\}$ ,  $\Omega_{wi}$  represents the significance of the  $i$ -th weight, where  $i = 1, \dots, n$ , of the NN in step  $j$  after learning from prior tasks, the diagonal elements of the FIM  $\bar{\Omega}_{kj}$ , with  $k$  denoting the task, are estimated using HJB error since targets are unavailable in online learning,  $\lambda_j$  is the design parameter controlling the strength of the regularization,  $\left\| \hat{W}_{c_j} - \hat{W}_{c_{jp}}^* \right\|_{\bar{\Omega}_{kj}} = (\hat{W}_{c_j} - \hat{W}_{c_{jp}}^*) \bar{\Omega}_{kj} (\hat{W}_{c_j} - \hat{W}_{c_{jp}}^*)^\top$ ,  $\hat{W}_{c_{jp}}^*$  is the optimized bounded weight vector of the previous task, and  $\hat{W}_{c_j}$  is the weight vector of the current task that needs to be optimized. The FIM for each

task is calculated by defining the log-likelihood function as:

$$\ell(\hat{W}_{cjp}, z_{jp}) = \log p(e_{jp} | \hat{W}_{cjp}, z_{jp}), \quad (46)$$

where  $e_{jp}$  represents the HJB residual error, as defined by (20) and (35) from the previous task. The term  $p(e_{jp} | \hat{W}_{cjp}, z_{jp})$  denotes the probability density function of the HJB error, given the input  $z_{jp}$  from the previous task and the weights  $\hat{W}_{cjp}$  at step  $j$  from the previous task. Calculate the Jacobian matrix as

$$J(\hat{W}_{cjp}, z_{jp}) = \frac{\partial \ell(\hat{W}_{cjp}, z_{jp})}{\partial \hat{W}_{cjp}}, \quad (47)$$

where  $\frac{\partial \ell(\hat{W}_{cjp}, z_{jp})}{\partial \hat{W}_{cjp}}$  denotes the partial derivative of the log-likelihood function with respect to the weights from the previous task. Therefore, the estimation of FIM is obtained as

$$\bar{\Omega}_{kj} = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} J(\hat{W}_{cjp}, z_{jp}) J(\hat{W}_{cjp}, z_{jp})^T dt, \quad (48)$$

where  $t_0$  denotes the task start time,  $t_1$  denotes the task end time. The elements of the FIM illustrate the extent of information concerning the HJB error that is conveyed by every weight within the network. Sure! Here's a clearer and more structured way to convey the information. For the first task, the FIM is zero. When estimating the FIM for the second task using data from the first task, the estimate remains bounded. This bounded behavior is due to the fact that the closed-loop system associated with task 1 is also bounded, as demonstrated in Theorem 1.

Subsequently, leveraging normalized gradient descent allows us to formulate an additional term in the critic weight update law. This term is derived as follows

$$-\frac{\partial}{\partial \hat{W}_{cj}}(L_j(\hat{W}_{cj})) = -\frac{\partial E_j(t)}{\partial \hat{W}_{cj}} - \lambda_j \bar{\Omega}_{kj} (\hat{W}_{cj} - \hat{W}_{cj}^*). \quad (49)$$

For LL, the terms from (49) are combined with the terms from the previously defined update law that is given in Theorem 1. Next, the following theorem is stated.

**Theorem 2** Consider the hypothesis stated in Theorem 1, and let Assumptions 1 to 4 hold, with the LRL critic NN tuning law for  $j$  step optimal backstepping, given by

$$\dot{\hat{W}}_{cj} = \gamma_{cj} \hat{\psi}_j(t) \left( e_j(t) - \frac{1}{4} \hat{W}_{cj}^T \nabla \sigma_{cj} \hat{\lambda}_j \nabla \sigma_{cj}^T \hat{W}_{cj} - \hat{W}_{cj}^T \nabla \sigma_{cj} \hat{\mathcal{F}}_{sj} \right) - \sigma_{j1} \hat{W}_{cj} - \alpha_j \lambda_j \bar{\Omega}_{kj} (\hat{W}_{cj} - \hat{W}_{cj}^*), \quad (50)$$

where  $j$  denotes the number of steps in backstepping,  $\lambda_j$  is the design parameter, and  $\alpha_j$  are the NN learning rates; then,  $\tilde{W}_{cj}$  and all closed-loop signals, including those in Theorem 1, are UUB.

**Proof:** See Appendix.

**Remark 10** From (49), when the significance of the weights increases,  $\Omega_{w_i}$  can become infinite; therefore,  $\frac{\Omega_{w_i}}{\Omega_{w_i} + 1}$  will become 1; hence, the gradient explosion can be avoided.

**Remark 11** *The first part of the NN weight update law in Theorem 2 is the same as in Theorem 1, whereas the second part includes regularization terms resulting from LL. Notice that the tracking and weight estimation error bounds increase due to the LRL-based control scheme because the bounding constant,  $\bar{C} = \bar{C}_{0j} + \gamma_{penj}$ , includes additional terms that are obtained from the penalty function from LL, where  $\bar{C}_{0j}$  is bound when LL is absent, and  $\gamma_{penj}$  is because of LL in  $j$ th step. The overall stability is unaffected due to LL.*

**Remark 12** *The proposed LL method is scalable to encompass  $n$  tasks. A third task,  $C$ , aims to maintain network weights in alignment with the learned weights from the preceding two tasks, implementable via single or dual penalties, given the quadratic nature of the penalties.*

**Remark 13** *The efficacy of the LL method is pronounced when Tasks 1 and 2 share informational overlap reflected in the weights, facilitating the knowledge transfer for Task 2. However, in the absence of shared weights or knowledge between non-overlapping tasks, the visible enhancement in Task 2 performance might be negligible with online LL, albeit it mitigates the catastrophic forgetting of Task 1, offering long-term benefits when reverting to Task 1.*

### 3. UAV TRACKING SIMULATION OUTCOMES

This section delineates the outcomes of optimal tracking control founded on LRL, applied on an underactuated UAV system.

#### 3.1 Unmanned aerial vehicle (UAV) problem formulation and control design

Consider the UAV model depicted in Figure 1, which is characterized by two reference frames: the inertial frame  $I = \{x_I, y_I, z_I\}$  fixed to the earth and the body-fixed frame  $A = \{x_A, y_A, z_A\}$ . The UAV is steered using four forces  $F_{j=1,2,3,4}$ , each produced by a respective rotor. To negate the yaw drift due to the reactive torque, rotors are arranged in two pairs, with (1, 3) spinning clockwise and (2, 4) spinning counterclockwise.

The quadrotor dynamics can be modeled by two unique equations: (1) translational; and (2) rotational. However, these dynamic equations interrelate via the rotation matrix, rendering them as two cohesive subsystems. A holistic control strategy involves both outer and inner loop controls, corresponding to the two subsystems. The outer loop aims to execute positional control by managing the state variables of  $x$ ,  $y$ , and  $z$  while concurrently generating command inputs for roll and pitch states in the inner loop through a correction block, dependent on a predefined yaw reference signal. The inner loop's objective is to achieve attitude control by managing the state variables of roll, pitch, and yaw.

Define  $\zeta_1(t) = [x(t), y(t), z(t)]^T \in \mathbb{R}^3$  as the UAV's positional state and  $\zeta_2(t) = [\dot{x}(t), \dot{y}(t), \dot{z}(t)]^T \in \mathbb{R}^3$  as its velocity state within the inertial frame  $I$ . A transformation relation exists:  $\zeta_2(t) = R(\eta_1)V$ . Consequently, the translational dynamic equation can be represented as

$$\begin{aligned} \dot{\zeta}_1(t) &= \zeta_2(t) \\ \dot{\zeta}_2(t) &= - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + R(\eta_1) \begin{bmatrix} 0 \\ 0 \\ 1/m \end{bmatrix} u_c. \end{aligned}$$

Given the underactuated nature of UAV translational dynamics, an intermediate control vector

$$V = R(\eta_1) \begin{bmatrix} 0 \\ 0 \\ 1/m \end{bmatrix} u_c$$

is introduced for optimal position control derivation, and the translational dynamic can thus be reformulated

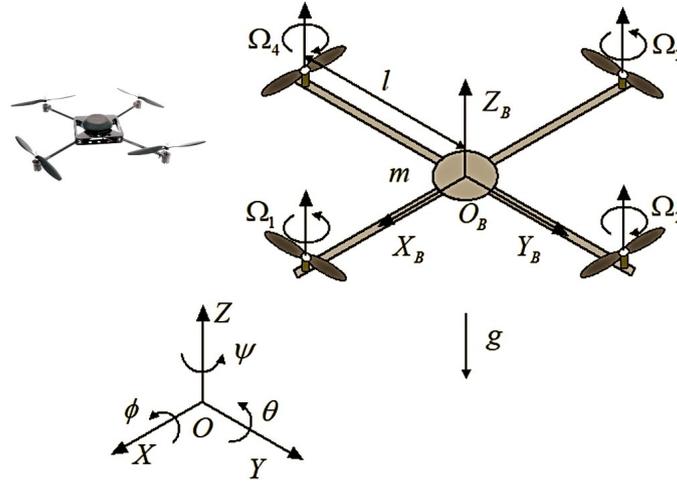


Figure 1. Quadrotor UAV.

as

$$\begin{aligned} \dot{\zeta}_1(t) &= \zeta_2(t) \\ \dot{\zeta}_2(t) &= - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + V. \end{aligned}$$

**Remark 14** The relation between  $V = [V_1, V_2, V_3]^T \in \mathbb{R}^3$  and  $u_c$  is given by

$$\begin{aligned} V_1 &= (\cos(\phi) \sin(\theta) \cos(\psi) + \sin(\phi) \sin(\psi)) \frac{u_c}{m} \\ V_2 &= (\cos(\phi) \sin(\theta) \sin(\psi) - \sin(\phi) \cos(\psi)) \frac{u_c}{m} \\ V_3 &= \cos(\phi) \cos(\theta) \frac{u_c}{m} \end{aligned}$$

Solving yields the control  $u_c$  as

$$u_c = m \left( V_1^2 + V_2^2 + V_3^2 \right)^{\frac{1}{2}}$$

Using the reference trajectory vector  $\zeta_{\text{ref}}(t) = [x_{\text{ref}}(t), y_{\text{ref}}(t), z_{\text{ref}}(t)]^T \in \mathbb{R}^3$ , we define the tracking error variables as  $e_{\zeta_1}(t) = \zeta_1(t) - \zeta_{\text{ref}}(t)$  and  $e_{\zeta_2}(t) = \zeta_2(t) - \alpha_\zeta$ , where  $\alpha_\zeta$  is the virtual control.

For the coordinate transformation of rotational dynamic, the transformation relationship between rotational velocity  $\Omega$  and the Euler angles rate of change  $\eta_2(t) = \dot{\eta}_1(t) = [\dot{\phi}(t), \dot{\theta}(t), \dot{\psi}(t)]^T \in \mathbb{R}^3$  is represented as:

$$\eta_2(t) = \dot{\eta}_1(t) = \Phi(\eta_1) \Omega$$

with

$$\Phi(\eta_1) = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) \sec(\theta) & c(\phi) \sec(\theta) \end{bmatrix}.$$

Applying time derivation to both sides yields the attitude dynamic as:

$$\ddot{\eta}_1(t) = -\Phi(\eta_1) I^{-1} \left( \Phi^{-1}(\eta_1) \eta_2(t) \times I \Phi^{-1}(\eta_1) \eta_2(t) \right) + \dot{\Phi}(\eta_1) \Phi^{-1}(\eta_1) \eta_2(t) + \Phi(\eta_1) I^{-1} \tau.$$

The function  $f(\eta_1, \eta_2)$  is represented as:

$$f(\eta_1, \eta_2) = -\Phi(\eta_1) I^{-1} \left( \Phi^{-1}(\eta_1) \dot{\eta}_2(t) \times I \Phi^{-1}(\eta_1) \eta_2(t) \right) + \dot{\Phi}(\eta_1) \Phi^{-1}(\eta_1) \eta_2(t)$$

So, the attitude dynamic can be rephrased in strict feedback form as

$$\begin{aligned} \dot{\eta}_1(t) &= \eta_2(t) \\ \dot{\eta}_2(t) &= f(\eta_1, \eta_2) + \Phi(\eta_1) I^{-1} \tau. \end{aligned}$$

Reference signals are denoted as  $\eta_{re}(t) = [\phi_{re}(t), \theta_{re}(t), \psi_{re}(t)]^T$ . The yaw command element  $\psi_{re}(t)$  is predefined, and the roll and pitch command angles are derived as

$$\begin{aligned} \phi_{re}(t) &= \arcsin \left( m \frac{V_1 \sin(\psi_{re}) - V_2 \cos(\psi_{re})}{u_c} \right) \\ \theta_{re}(t) &= \arctan \left( \frac{V_1 \cos(\psi_{re}) + V_2 \sin(\psi_{re})}{V_3} \right). \end{aligned}$$

Tracking error variables are designated as  $e_{\eta_1}(t) = \eta_1(t) - \eta_{re}(t)$  and  $e_{\eta_2}(t) = \eta_2(t) - \alpha_{\eta}$ , where  $\alpha_{\eta} \in \mathbb{R}^3$  is the virtual control.

Therefore, using the control law (34) and the weight update laws shown in Theorems 1 and 2 for translational and attitude dynamics will drive the UAV system to track the reference trajectory, as shown in the simulations.

### 3.2 Simulation parameters and results

The desired position trajectory for  $\xi_{re}(t) = [2t - 20, 5, 0]^T$  and, for task 2, is  $\xi_{re}(t) = [5 \sin(t), 5 \cos(t), t]^T$ . The yaw command angle is predefined as  $\psi_{re} = \pi/4$ ; then, the roll and pitch command signals  $\phi_{re}$  and  $\theta_{re}$  can be produced. The design parameter  $\beta_1 = [6, 0]$ ,  $\beta_2 = [4.2, 0]$ ,  $m = 5$ ,  $\sigma_{1j} = 0.75$  and  $\sigma_{2j} = 0.82$ ,  $\beta_{ci} = 0.6$ ,  $\beta_{ai} = 0.6$ ,  $\lambda_w = 2$ ,  $\alpha_w = 0.85$ . The initial values are set as  $z_1(0) = [1.7, 3.8, 0.41]^T$ ,  $z_2(0) = [2.31, 3, 0.4]^T$ . In this configuration, the NN is based on a Random Vector Functional Link (RVFL) architecture. The network is structured with a single layer, where the input layer is directly connected to the output layer, bypassing traditional hidden layers. The interlayer connections are facilitated by a weight matrix  $V$ , which is randomly initialized. This matrix  $V$  effectively represents the neurons in the network. Specifically, the network comprises ten neurons, each corresponding to a column in the  $V$  matrix. These neurons functionally link the input to the output through the RVFL mechanism. The transformation from the input to the output layer is given by  $W^T \sigma(V^T X)$ , where  $\sigma$  denotes the activation function applied element-wise, and  $X$  is the input and  $W$  is tuned using the weight update laws in Theorems 1 and 2. Similarly, seven neurons are used for the identifier; sigmoid is used as an activation function,  $\epsilon_0 = 0.65$ . For attitude control, the initial values are set as  $z_3(0) = [\pi/4, \pi/4, \pi/4]^T$ ,  $z_4(0) = [\pi/3, \pi/3, \pi/3]^T$ .

We consider two task scenarios in which the reference trajectory is changed in each task as if the UAV is moving in a different path or environment. In the simulations, we have shown task 1 again to demonstrate that when the UAV returns to task 1, the LL-based control will help mitigate the catastrophic forgetting. The proposed method is able to drive the UAV to track the reference trajectory accurately, even on changing tasks. Figure 2 shows the performance of the position and attitude tracking; it indicates that using the proposed LRL method shown by the blue color, and the UAV position states can accurately follow the reference trajectory shown by the red color. The attitude tracking performance demonstrates that the UAV attitudes can also follow the reference attitudes better as compared to recent literature<sup>[9]</sup> shown by green color. Figure 3 illustrates the tracking errors, indicating that the tracking performance of the proposed method is superior when compared with recent literature, referred to as 'r-lit'<sup>[9]</sup>. Figure 4 illustrates the position and attitude tracking errors.

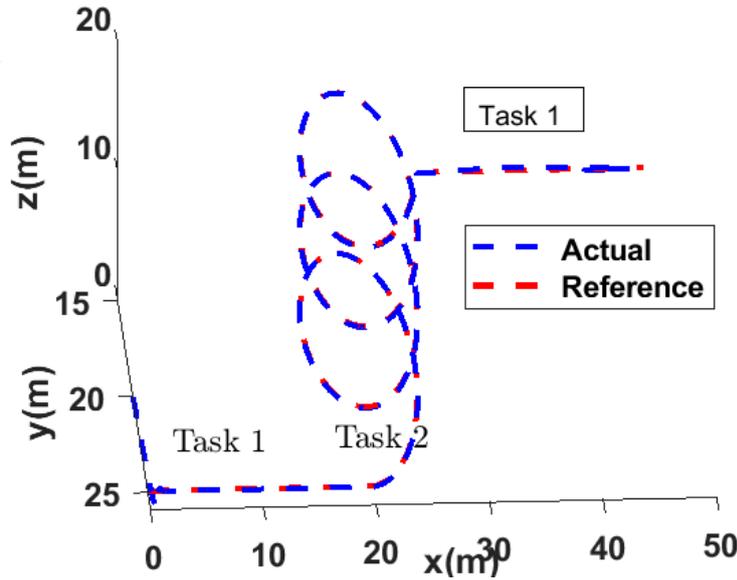


Figure 2. Actual and reference trajectories using the proposed LL-based method.

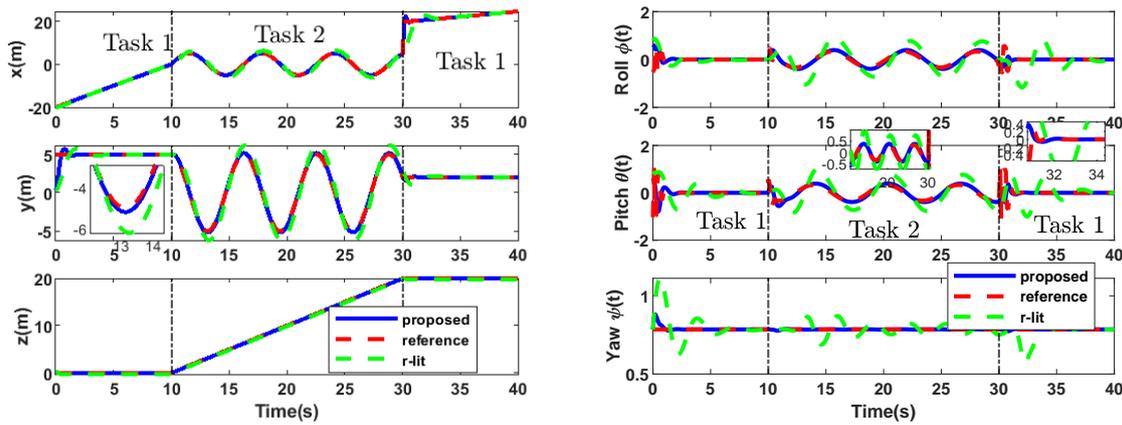


Figure 3. Tracking performance of position and attitude subsystems using LRL and recent literature (r-lit)<sup>[9]</sup> methods.

Both the system state tracking plots and positional error plots in Figure 2 and Figure 3 demonstrate the superior performance of the proposed LL method, represented by blue lines. However, the recent literature<sup>[9]</sup> exhibits higher error, thus showing the need for LL. In contrast, the 'Lit' method, as shown in green, has a higher error rate when compared to other methods. Notably, the total average error shown in Figure 4 is low when the proposed LL method is employed over the 'Lit' method, indicating a substantial enhancement in tracking accuracy.

Figure 5 depicts torque inputs and cumulative costs where it can be seen that the cost of using the proposed method is minimal, and all the closed-loop signals are bounded. The control effort demanded by the 'Lit' method is higher in comparison to the proposed LL-based method. Figure 5 also showcases the cumulative cost. It is observed that the cost associated with Lit (shown in green color) is higher compared to the proposed LL method (represented in blue) during the tasks and as the tasks change.

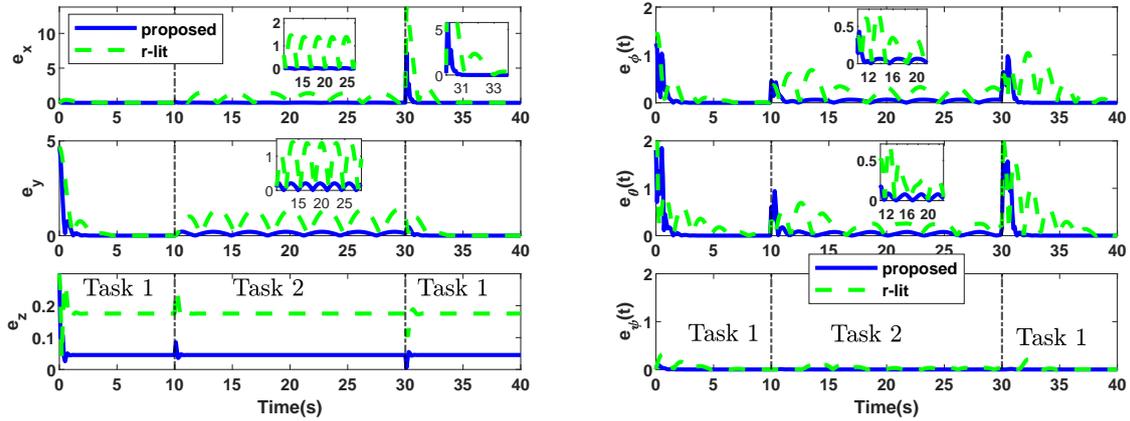


Figure 4. Position and attitude tracking errors using proposed LRL and recent literature (r-lit)<sup>[9]</sup> methods.

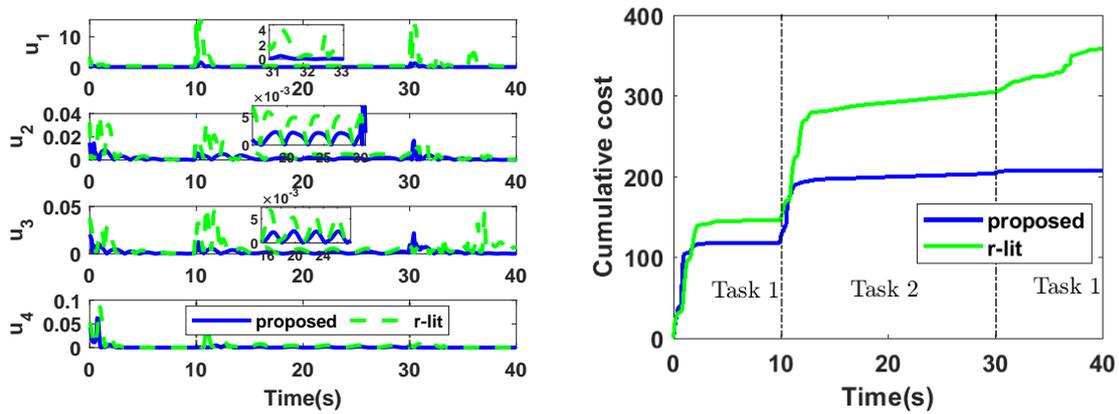


Figure 5. Torque inputs and cumulative cost using proposed LRL and recent literature (r-lit)<sup>[9]</sup> methods.

#### 4. CONCLUSION AND DISCUSSION

This paper proposed an innovative LL tracking control technique for uncertain nonlinear CT systems in strict feedback form. The method combined the augmented system, trajectory generator, and optimal backstepping approach to design both feedforward and feedback terms of the tracking scheme. By utilizing a combination of actor-critic NN and identifier NN, the method effectively approximated the solution to the HJB equations with unknown nonlinear functions. The use of RL at each step of the backstepping process allows for the development of virtual and actual optimal controllers that can effectively handle the challenges posed by uncertain, strict feedback systems. The proposed work highlighted the significance of considering catastrophic forgetting in online controller design and developed a new method to address this issue. Simulation results on a UAV tracking a desired trajectory show acceptable performance. The proposed approach can be extended by using deep NNs for better approximation. In addition, the integral RL (IRL)-based approach can relax the drift dynamics. Dynamic surface control can be included to minimize the number of NNs used.

#### DECLARATIONS

##### Authors' contributions

Made substantial contributions to the conception and design of the study: Ganie I, Jagannathan S  
 Made contributions in writing, reviewing, editing, and methodology: Ganie I, Jagannathan S

### Availability of data and materials

Not applicable.

### Financial support and sponsorship

The project or effort undertaken was or is sponsored by the Office of Naval Research Grant N00014-21-1-2232 and Army Research Office Cooperative Agreements W911NF-21-2-0260 and W911NF-22-2-0185.

### Conflicts of interest

Both authors declared that there are no conflicts of interest.

### Ethical approval and consent to participate

Not applicable.

### Consent for publication

Not applicable.

### Copyright

© The Author(s) 2024.

## REFERENCES

1. Abu-Khalaf M, Lewis FL. Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. *Automatica* 2005;41:779–91. [DOI](#)
2. McLain TW, Beard RW. Successive galerkin approximations to the nonlinear optimal control of an underwater robotic vehicle. In Proceedings of the 1998 IEEE international conference on robotics and automation (Cat. No.98CH36146). Leuven, Belgium. 20-20 May 1998. [DOI](#)
3. Vrabie D, Pastravanu O, Abu-Khalaf M, Lewis FL. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica* 2009;45:477–84. [DOI](#)
4. Modares H, Lewis FL. Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning. *Automatica* 2014;50:1780–92. [DOI](#)
5. Gao W, Jiang ZP. Learning-Based adaptive optimal tracking control of strict-feedback nonlinear systems. *IEEE Trans Neural Netw Learn Syst* 2018;29:2614–24. [DOI](#)
6. Zargarzadeh H, Dierks T, Jagannathan S. Optimal control of nonlinear continuous-time systems in strict-feedback form. *IEEE Trans Neural Netw Learn Syst* 2015;26:2535–49. [DOI](#)
7. Huang Z, Bai W, Li T, et al. Adaptive reinforcement learning optimal tracking control for strict-feedback nonlinear systems with prescribed performance. *Inf Sci* 2023;621:407–23. [DOI](#)
8. Wen G, Chen CLP, Ge SS. Simplified optimized backstepping control for a class of nonlinear strict-feedback systems with unknown dynamic functions. *IEEE Trans Cybern* 2021;51:4567–80. [DOI](#)
9. Wen G, Hao W, Feng W, Gao K. Optimized backstepping tracking control using reinforcement learning for quadrotor unmanned aerial vehicle system. *IEEE Trans Syst Man Cybern Syst* 2022;52:5004–15. [DOI](#)
10. Bryson AE. Applied optimal control: optimization, estimation and control. New York: Routledge; 1975. p. 496. [DOI](#)
11. Wen G, Ge SS, Tu F. Optimized backstepping for tracking control of strict-feedback systems. *IEEE Trans Neural Netw Learn Syst* 2018;29:3850–62. [DOI](#)
12. Wu J, Wang W, Ding S, Xie X, Yi Y. Adaptive neural optimized control for uncertain strict-feedback systems with unknown control directions and pre-set performance. *Commun Nonlinear Sci Numer Simul* 2023;126:107506. [DOI](#)
13. Kirkpatrick J, Pascanu R, Rabinowitz NC, et al. Overcoming catastrophic forgetting in neural networks. *Proc Natl Acad Sci USA* 2017;114:3521–6. [DOI](#)
14. Ganie I, Jagannathan S. Adaptive control of robotic manipulators using deep neural networks. *IFAC-PapersOnLine* 2022;55:148–53. [DOI](#)
15. Kutalev A, Lapina A. Stabilizing elastic weight consolidation method in practical ML tasks and using weight importances for neural network pruning. *ArXiv* 2021. Available from: <https://arxiv.org/abs/2109.10021> [Last accessed on 2 Feb 2024].
16. Liu Y, Zhu Q, Wen G. Adaptive tracking control for perturbed strict-feedback nonlinear systems based on optimized backstepping technique. *IEEE Trans Neural Netw Learn Syst* 2022;33:853–65. [DOI](#)
17. Moghadam R, Jagannathan S. Optimal adaptive control of uncertain nonlinear continuous-time systems with input and state delays. *IEEE Trans Neural Netw Learn Syst* 2023;34:3195–204. [DOI](#)
18. Mishra A, Ghosh S. Simultaneous identification and optimal tracking control of unknown continuous-time systems with actuator constraints. *Int J Control* 2022;95:2005–23. [DOI](#)

## APPENDICES

### Proof of Theorem 1

**Step1:** Consider the Lyapunov function as follows

$$L_1 = J_1(z_1) + \frac{1}{2}e_{i1}^T e_{i1} + \frac{1}{2}tr\{\tilde{Z}_1^T \tilde{Z}_1\} + \frac{1}{2}tr\{\tilde{W}_{c1} \tilde{W}_{c1}^T\} + \frac{1}{2}tr\{\tilde{W}_{a1} \tilde{W}_{a1}^T\}. \quad (51)$$

The time derivative of  $L_1$  is

$$\dot{L}_1 = \dot{J}_1(z_1) + tr\{e_{i1}^T \dot{e}_{i1}\} + tr\{\tilde{Z}_1^T \dot{\tilde{Z}}_1\} + tr\{\tilde{W}_{c1}^T \dot{\tilde{W}}_{c1}\} + tr\{\tilde{W}_{a1}^T \dot{\tilde{W}}_{a1}\}. \quad (52)$$

Where  $tr$  denotes the trace operator. Let  $\dot{L}_1 = \dot{L}_{11} + \dot{L}_{12} + \dot{L}_{13} + \dot{L}_{14} + \dot{L}_{15}$ , where  $L_{11} = J_1(z_1)$ ,  $L_{12} = \frac{1}{2}e_{i1}^T e_{i1}$ ,  $L_{13} = \frac{1}{2}tr\{\tilde{Z}_1^T \tilde{Z}_1\}$ ,  $L_{14} = \frac{1}{2}tr\{\tilde{W}_{c1} \tilde{W}_{c1}^T\}$ ,  $L_{15} = \frac{1}{2}tr\{\tilde{W}_{a1} \tilde{W}_{a1}^T\}$ .

Considering the first term of (51), we can write it as  $\dot{J}_1(z_1) = \nabla J_1 \dot{z}_1$ .

Substituting (3), (9) in (52) gives

$$\dot{J}_1(z_1) = (W_{c1}^T \nabla \sigma_{c1} + \nabla \varepsilon_{c1})(\mathcal{F}_{s1}(z_1) + \mathcal{G}_{s1}(z_1)\alpha_1). \quad (53)$$

Substituting the value of  $\alpha$  from (8) in (53) leads to

$$\dot{J}_1(z_1) = (W_{c1}^T \nabla \sigma_{c1} + \nabla \varepsilon_{c1})(\mathcal{F}_{s1}(z_1) - \frac{1}{2}\mathcal{G}_{s1}(z_1)r^{-1}\hat{\mathcal{G}}_{s1}^T(z_1)(\nabla \sigma_{c1}^T \hat{W}_{c1})). \quad (54)$$

Using  $\hat{W}_{c1} = W_{c1} - \tilde{W}_{c1}$  in (54), and simplifying, one will get

$$\dot{J}_1(z_1) = (W_{c1}^T \nabla \sigma_{c1} + \nabla \varepsilon_{c1})\left(\mathcal{F}_{s1}(z_1) - \frac{1}{2}\mathcal{G}_{s1}(z_1)r^{-1}\hat{\mathcal{G}}_{s1}^T(z_1)\nabla \sigma_{c1}^T(W_{c1} - \tilde{W}_{c1})\right) \quad (55)$$

Separating the terms in (55) w.r.t actual NN weights and the terms w.r.t weight estimation error gives

$$\begin{aligned} \dot{J}_1(z_1) &= W_{c1}^T \nabla \sigma_{c1} \mathcal{F}_{s1}(z_1) - \frac{1}{2}W_{c1}^T \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1)r^{-1}\hat{\mathcal{G}}_{s1}^T(z_1)\nabla \sigma_{c1}^T W_{c1} \\ &+ \frac{1}{2}W_{c1}^T \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1)r^{-1}\hat{\mathcal{G}}_{s1}^T(z_1)\nabla \sigma_{c1}^T \tilde{W}_{c1} + \nabla \varepsilon_{c1} \mathcal{F}_{s1}(z_1) \\ &- \frac{\nabla \varepsilon_{c1}}{2}\mathcal{G}_{s1}(z_1)r^{-1}\hat{\mathcal{G}}_{s1}^T(z_1)\nabla \sigma_{c1}^T W_{c1} + \frac{\nabla \varepsilon_{c1}}{2}\mathcal{G}_{s1}(z_1)r^{-1}\hat{\mathcal{G}}_{s1}^T(z_1)\nabla \sigma_{c1}^T \tilde{W}_{c1}. \end{aligned} \quad (56)$$

Substituting  $\hat{\mathcal{G}}_{s1} = \mathcal{G}_{s1} - \tilde{\mathcal{G}}_{s1}$  in (56) leads to

$$\begin{aligned} \dot{J}_1(z_1) &= W_{c1}^T \nabla \sigma_{c1} \mathcal{F}_{s1}(z_1) - \frac{1}{2}W_{c1}^T \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1)r^{-1}(\mathcal{G}_{s1} - \tilde{\mathcal{G}}_{s1})^T \nabla \sigma_{c1}^T W_{c1} + \frac{1}{2}W_{c1}^T \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1)r^{-1} \\ &(\mathcal{G}_{s1} - \tilde{\mathcal{G}}_{s1})^T \nabla \sigma_{c1}^T \tilde{W}_{c1} + \nabla \varepsilon_{c1} \mathcal{F}_{s1}(z_1) - \frac{\nabla \varepsilon_{c1}}{2}\mathcal{G}_{s1}(z_1)r^{-1}(\mathcal{G}_{s1} - \tilde{\mathcal{G}}_{s1})^T \nabla \sigma_{c1}^T W_{c1} \\ &+ \frac{\nabla \varepsilon_{c1}}{2}\mathcal{G}_{s1}(z_1)r^{-1}(\mathcal{G}_{s1} - \tilde{\mathcal{G}}_{s1})^T \nabla \sigma_{c1}^T \tilde{W}_{c1}. \end{aligned} \quad (57)$$

One can further simplify (57), as follows

$$\begin{aligned}
 j_1(z_1) = & \operatorname{tr}\{W_{c1}^\top \nabla \sigma_{c1} \mathcal{F}_{s1}(z_1) - \frac{1}{2} W_{c1}^\top \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1) r^{-1} \mathcal{G}_{s1}^\top \nabla \sigma_{c1}^\top W_{c1} \\
 & + \frac{1}{2} W_{c1}^\top \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1) r^{-1} \mathcal{G}_{s1}^\top \nabla \sigma_{c1}^\top \tilde{W}_{c1} + \nabla \varepsilon_{c1} \mathcal{F}_{s1}(z_1) \\
 & - \frac{\nabla \varepsilon_{c1}}{2} \mathcal{G}_{s1}(z_1) r^{-1} \mathcal{G}_{s1}^\top \nabla \sigma_{c1}^\top W_{c1} + \frac{\nabla \varepsilon_{c1}}{2} \mathcal{G}_{s1}(z_1) r^{-1} \mathcal{G}_{s1}^\top \nabla \sigma_{c1}^\top \tilde{W}_{c1} \\
 & + \frac{1}{2} W_{c1}^\top \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1) r^{-1} \tilde{\mathcal{G}}_{s1}^\top \nabla \sigma_{c1}^\top W_{c1} - \frac{1}{2} W_{c1}^\top \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1) r^{-1} \tilde{\mathcal{G}}_{s1}^\top \nabla \sigma_{c1}^\top \tilde{W}_{c1} \\
 & + \frac{\nabla \varepsilon_{c1}}{2} \mathcal{G}_{s1}(z_1) r^{-1} \tilde{\mathcal{G}}_{s1}^\top \nabla \sigma_{c1}^\top W_{c1} - \frac{\nabla \varepsilon_{c1}}{2} \mathcal{G}_{s1}(z_1) r^{-1} \tilde{\mathcal{G}}_{s1}^\top \nabla \sigma_{c1}^\top \tilde{W}_{c1}\}.
 \end{aligned} \tag{58}$$

Using (11), we have  $W_{c1}^\top \nabla \sigma_{c1} \mathcal{F}_{s1}(z_1) - \frac{1}{2} W_{c1}^\top \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1) r^{-1} \mathcal{G}_{s1}(z_1) W_{c1} \nabla \sigma_{c1} \leq -k \|z_1\|^2$ . Additionally, it is also assumed that the effect of the NN approximation error in the system dynamics is bounded above, i.e.,  $\nabla \varepsilon_{c1} \mathcal{F}_{s1}(z_1) - \frac{1}{2} \nabla \varepsilon_{c1} \mathcal{G}_{s1}(z_1) r^{-1} \mathcal{G}_{s1}(z_1) W_{c1} \nabla \sigma_{c1} \leq -\bar{\varepsilon} \|z_1\|^2$ . One can write (58), as follows

$$j_1(z_1) = -(k + \bar{\varepsilon}) z_1^2 + k_1 \tilde{W}_{c1} + k_2 \tilde{W}_{c1} + k_3 \tilde{\mathcal{G}}_{s1} + k_4 \tilde{\mathcal{G}}_{s1} + k_5 \tilde{W}_{c1} \tilde{\mathcal{G}}_{s1} + k_6 \tilde{W}_{c1} \tilde{\mathcal{G}}_{s1}. \tag{59}$$

which can be further written as

$$j_1(z_1) = -\bar{k}_1 z_1^2 + \bar{k}_2 \tilde{W}_{c1} + \bar{k}_3 \tilde{\mathcal{G}}_{s1} + \bar{k}_4 \tilde{W}_{c1} \tilde{\mathcal{G}}_{s1}, \tag{60}$$

where  $\frac{1}{2} W_{c1}^\top \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1) r^{-1} \mathcal{G}_{s1} \nabla \sigma_{c1} \leq k_1$ ,  $\frac{\nabla \varepsilon_{c1}}{2} \mathcal{G}_{s1}(z_1) r^{-1} \mathcal{G}_{s1} \nabla \sigma_{c1} \leq k_2$ ,  $\frac{1}{2} W_{c1}^\top \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1) r^{-1} W_{c1} \nabla \sigma_{c1} \leq k_3$ ,

$\frac{\nabla \varepsilon_{c1}}{2} \mathcal{G}_{s1}(z_1) r^{-1} W_{c1} \nabla \sigma_{c1} \leq k_4$ ,  $\frac{1}{2} W_{c1}^\top \nabla \sigma_{c1} \mathcal{G}_{s1}(z_1) r^{-1} \nabla \sigma_{c1} \leq k_5$ ,  $\frac{\nabla \varepsilon_{c1}}{2} \mathcal{G}_{s1}(z_1) r^{-1} \nabla \sigma_{c1} \leq k_6$ . Consider the second and third term of (52)

$$\dot{L}_{12} + \dot{L}_{13} = \operatorname{tr}\{e_{i1}^\top \dot{e}_{i1}\} + \operatorname{tr}\{\tilde{Z}_1^\top \dot{\tilde{Z}}_1\} \tag{61}$$

On substituting the value of  $\dot{e}_{i1}$  and  $\dot{\tilde{Z}}_1$  from Theorem 1, one can write RHS of (83) as follows

$$\operatorname{tr}\left\{e_{i1}^\top (-K e_{i1} + \tilde{Z}_1 \sigma(\xi) \hat{u} + \varepsilon_I(z_1)) + \tilde{Z}_1^\top (-\alpha_v \tilde{Z}_1 - \sigma(\xi) \hat{u} e_{i1}^\top + \alpha_v Z_1)\right\}. \tag{62}$$

which can be further simplified by using the cyclic property of traces as

$$\operatorname{tr}\{e_{i1}^\top (-K e_{i1} + \varepsilon_I(z_1)) + \tilde{Z}_1^\top (-\alpha_v \tilde{Z}_1 + \alpha_v Z_1)\}. \tag{63}$$

To simplify, we have

$$\dot{L}_{12} + \dot{L}_{13} \leq -K \|e_{i1}\|^2 - \alpha_v \|\tilde{Z}_1\|^2 + \|e_{i1}\|^2 + \|z_1\|^2 + \|\alpha_v \tilde{Z}_1^\top Z_1\| \tag{64}$$

Consider the fourth term of (52),

$$\dot{L}_{14} = \operatorname{tr}\left\{\tilde{W}_{c1}^\top \left(\gamma_c \hat{\psi}_1(t) (-e_1(t) + \frac{1}{4} \hat{W}_{c1}^\top \nabla \sigma_{c1} \wedge \nabla \sigma_{c1}^\top \hat{W}_{c1} + \hat{W}_{c1}^\top \nabla \sigma_{c1} \mathcal{F}_{s1}) + \sigma_{11} \hat{W}_{c1}\right)\right\} \tag{65}$$

Therefore, one can further simplify (65) by using Young's inequality in cross-product terms as follows

$$\dot{L}_{14} \leq c_1 \|\tilde{W}_{c1}\|^2 + c_2 \|\tilde{W}_{c1}\| + c_3 \|\tilde{Z}_1\|^2 + c_4 \|\tilde{W}_{c1}\| - c_5 \|\tilde{W}_{c1}\|^2 \tag{66}$$

Consider the fifth term of (52)  $\dot{L}_{15} = tr\{\tilde{W}_{a1}^T \dot{W}_{a1}\}$ . On using the weight update law from Theorem 2, one can write

$$\dot{L}_{15} = tr\left\{\tilde{W}_{a1}^T(\beta_a S_3(\beta_1 z_1 + \frac{1}{2}\hat{W}_{a1}^T \sigma_{a1}(\bar{z}_1) - \frac{1}{2}r^{-1}\hat{\mathcal{G}}_{s1}^T \nabla \sigma_{c1}^T \hat{W}_{c1}) + \sigma_{22}\hat{W}_{a1})\right\} \quad (67)$$

Using  $\tilde{W}_{c1} = W_{c1} - \hat{W}_{c1}, \tilde{W}_{a1} = W_{a1} - \hat{W}_{a1}, \tilde{\mathcal{G}}_{s1} = \mathcal{G}_{s1} - \hat{\mathcal{G}}_{s1}$

$$\dot{L}_{15} = tr\left\{\tilde{W}_{a1}^T(\beta_a S_3(\beta_1 z_1 + \frac{1}{2}(W_{a1} - \hat{W}_{a1})^T \sigma_{a1}(\bar{z}_1) - \frac{1}{2}r^{-1}(\mathcal{G}_{s1} - \hat{\mathcal{G}}_{s1})^T \nabla \sigma_{c1}^T (W_{c1} - \hat{W}_{c1}) + \sigma_{22}\tilde{W}_{a1}(W_{a1} - \hat{W}_{a1}))\right\} \quad (68)$$

Considering a last term, we can write

$$\tilde{W}_{a1}(W_{a1} - \hat{W}_{a1}) = \tilde{W}_{a1}W_{a1} - \tilde{W}_{a1}^2$$

Using Young's inequality in the cross product terms, we can write

$$\dot{L}_{15} = c_5\|\tilde{W}_{a1}\|^2 + c_6\|z_1\|^2 + c_7\|\tilde{Z}_1\|^2 + c_8\|W_{c1}\|^2 + c_9\|\tilde{Z}_1\| + c_{10}\|\tilde{W}_{c1}\| + c_{11}\|\tilde{W}_{a1}\| - c_{12}\|\tilde{W}_{a1}\|^2 \quad (69)$$

Combining (60), (64),(66) and (69) and simplifying, we have

$$\dot{L}_1 \leq -\bar{c}_1\|z_1\|^2 + \bar{k}_2\|\tilde{W}_{c1}\| + \bar{k}_3\|\tilde{Z}_1\| - \bar{c}_2\|\tilde{W}_{c1}\|^2 - \bar{c}_3\|\tilde{Z}_1\|^2 - \bar{c}_5\|e_{i1}\|^2 + c_0\|\tilde{W}_{a1}\| - \bar{c}_4\|\tilde{W}_{a1}\|^2, \quad (70)$$

where  $\bar{c}_1 = \bar{k}_1 - c_{13}, \bar{k}_1 > c_{13}, \bar{c}_2 = c_4 - \bar{k}_4, c_4 > \bar{k}_4, \bar{c}_3 = \alpha_v - k_4, \alpha_v > \bar{k}_4, \bar{c}_4 = c_{12} - c_5, c_{12} > c_5$ . We can simplify it further as follows

$$\dot{L}_1 < -\bar{c}_1\|z_1\|^2 - \bar{c}_2(\|\tilde{W}_{c1}\| - \frac{\bar{k}_2}{\bar{c}_2})^2 - \bar{c}_3(\|\tilde{Z}_1\| - \frac{\bar{k}_3}{\bar{c}_3})^2 - \bar{c}_4(\|\tilde{W}_{a1}\| - \frac{c_0}{\bar{c}_4})^2 - \bar{c}_5\|e_{i1}\|^2 + \bar{C}_{01}, \quad (71)$$

where  $\bar{C}_{01} = \frac{k_2^2}{c_2} + \frac{k_3^2}{c_3} + \frac{c_0^2}{c_4}$ . Therefore, we can write

$$\begin{cases} \|z_1\| > \sqrt{\frac{\bar{C}_{01}}{c_1}} \text{ or } \|\tilde{W}_{c1}\| > \sqrt{\frac{\bar{C}_{01}}{c_2}} + \frac{k_2}{c_2} \text{ or} \\ \|\tilde{W}_{a1}\| > \sqrt{\frac{\bar{C}_{01}}{c_4}} + \frac{c_0}{c_4} \text{ or } \|\tilde{Z}_1\| > \sqrt{\frac{\bar{C}_{01}}{c_3}} + \frac{k_3}{c_3} \text{ or } \|e_{i1}\| > \sqrt{\frac{\bar{C}_{01}}{c_5}}. \end{cases} \quad (72)$$

This demonstrates that the overall closed-loop system is bounded. Since  $\tilde{G}$  is a function of the weight estimation error of the NN identifier, or  $\tilde{Z}$ , and from (72),  $\tilde{Z}$  is bounded; as a consequence,  $\hat{G}$  is bounded. From (72)  $\tilde{W}$ , identification error and system state are bounded. As a result, the control input error becomes bounded.

### Step 2:

This is the final step. Consider the Lyapunov function as follows

$$L_2 = L_1 + J_2(z_2) + \frac{1}{2}e_{i2}^T e_{i2} + \frac{1}{2}tr\{\tilde{Z}_2^T \tilde{Z}_2\} + \frac{1}{2}tr\{\tilde{W}_{c2} \tilde{W}_{c2}^T\} + \frac{1}{2}tr\{\tilde{W}_{a2} \tilde{W}_{a2}^T\}. \quad (73)$$

The time derivative of  $L_2$  is

$$\dot{L}_2 = \dot{L}_1 + J_2(z_2) + tr\{e_{i2}^T \dot{e}_{i2}\} + tr\{\tilde{Z}_2^T \dot{\tilde{Z}}_2\} + tr\{\tilde{W}_{c2}^T \dot{\tilde{W}}_{c2}\} + tr\{\tilde{W}_{a2}^T \dot{\tilde{W}}_{a2}\}. \quad (74)$$

Let  $\dot{L}_2 = L_1 + \dot{L}_{21} + \dot{L}_{22} + \dot{L}_{23} + \dot{L}_{24} + \dot{L}_{25}$ . where  $L_{21} = J_2(z_2)$ ,  $L_{22} = \frac{1}{2}e_{i2}^\top e_{i2}$ ,  $L_{23} = \frac{1}{2}tr\{\tilde{Z}_2^\top \tilde{Z}_2\}$ ,  $L_{24} = \frac{1}{2}tr\{\tilde{W}_{c2}\tilde{W}_{c2}^\top\}$ ,  $L_{25} = \frac{1}{2}tr\{\tilde{W}_{a2}\tilde{W}_{a2}^\top\}$ .

Considering the second term of (74), substituting (3) in  $J_2(z_2)$ , it gives

$$J_2(z_2) = (W_{c2}^\top \nabla \sigma_{c2} + \nabla \varepsilon_{c2})(\mathcal{F}_{s2}(z_2) + \mathcal{G}_{s2}(z_2)u). \quad (75)$$

which on further solving leads to

$$J_2(z_2) = (W_{c2}^\top \nabla \sigma_{c2} + \nabla \varepsilon_{c2})(\mathcal{F}_{s2}(z_2) - \frac{1}{2}\mathcal{G}_{s2}(z_2)r_2^{-1}(\hat{\mathcal{G}}_{s2}^\top(z_2)(\nabla \sigma_{c2}^\top \hat{W}_{c2}))). \quad (76)$$

Using  $\hat{W}_{c2} = W_{c2} - \tilde{W}_{c2}$  in (76), and simplifying, one will get

$$J_2(z_2) = (W_{c2}^\top \nabla \sigma_{c2} + \nabla \varepsilon_{c2})(\mathcal{F}_{s2}(z_2) - \frac{1}{2}\mathcal{G}_{s2}(z_2)r_2^{-1}\mathcal{G}_{s2}^\top(z_1)\nabla \sigma_{c2}^\top(W_{c2} - \tilde{W}_{c2})). \quad (77)$$

Separating the terms in (77) w.r.t actual NN weights and the terms w.r.t weight estimation error gives

$$\begin{aligned} J_2(z_2) &= W_{c2}^\top \nabla \sigma_{c2} \mathcal{F}_{s2}(z_2) - \frac{1}{2}W_{c2}^\top \nabla \sigma_{c2} \mathcal{G}_{s2}(z_2)r_2^{-1}\hat{\mathcal{G}}_{s2}^\top(z_2)\nabla \sigma_{c2}^\top W_{c2} \\ &+ \frac{1}{2}W_{c2}^\top \nabla \sigma_{c2} \mathcal{G}_{s2}(z_2)r_2^{-1}\hat{\mathcal{G}}_{s2}^\top(z_2)\nabla \sigma_{c2}^\top \tilde{W}_{c2} + \nabla \varepsilon_{c2} \mathcal{F}_{s2}(z_2) \\ &- \frac{\nabla \varepsilon_{c2}}{2}\mathcal{G}_{s2}(z_2)r_2^{-1}\hat{\mathcal{G}}_{s2}^\top(z_2)\nabla \sigma_{c2}^\top W_{c2} + \frac{\nabla \varepsilon_{c2}}{2}\mathcal{G}_{s2}(z_2)r_2^{-1}\hat{\mathcal{G}}_{s2}^\top(z_2)\nabla \sigma_{c2}^\top \tilde{W}_{c2}. \end{aligned} \quad (78)$$

Substituting  $\hat{\mathcal{G}}_{s2} = \mathcal{G}_{s2} - \tilde{\mathcal{G}}_{s2}$  in (78) leads to

$$\begin{aligned} J_2(z_2) &= W_{c2}^\top \nabla \sigma_{c2} \mathcal{F}_{s2}(z_2) - \frac{1}{2}W_{c2}^\top \nabla \sigma_{c2} \mathcal{G}_{s2}(z_2)r_2^{-1}(\mathcal{G}_{s2} - \tilde{\mathcal{G}}_{s2})^\top \nabla \sigma_{c2}^\top W_{c2} \\ &+ \frac{1}{2}W_{c2}^\top \nabla \sigma_{c2} \mathcal{G}_{s2}(z_2)r_2^{-1}(\mathcal{G}_{s2} - \tilde{\mathcal{G}}_{s2})^\top \nabla \sigma_{c2}^\top \tilde{W}_{c2} + \nabla \varepsilon_{c2} \mathcal{F}_{s2}(z_2) - \frac{\nabla \varepsilon_{c2}}{2}\mathcal{G}_{s2}(z_2)r_2^{-1}(\mathcal{G}_{s2} - \tilde{\mathcal{G}}_{s2})^\top \nabla \sigma_{c2}^\top W_{c2} \\ &+ \frac{\nabla \varepsilon_{c2}}{2}\mathcal{G}_{s2}(z_2)r_2^{-1}(\mathcal{G}_{s2} - \tilde{\mathcal{G}}_{s2})^\top \nabla \sigma_{c2}^\top \tilde{W}_{c2}. \end{aligned} \quad (79)$$

One can further simplify (79), as follows

$$\begin{aligned} J_2(z_2) &= tr\{W_{c2}^\top \nabla \sigma_{c2} \mathcal{F}_{s2}(z_2) - \frac{1}{2}W_{c2}^\top \nabla \sigma_{c2} \mathcal{G}_{s2}(z_2)r_2^{-1}\mathcal{G}_{s2}^\top \nabla \sigma_{c2}^\top W_{c2} \\ &+ \frac{1}{2}W_{c2}^\top \nabla \sigma_{c2} \mathcal{G}_{s2}(z_2)r_2^{-1}\mathcal{G}_{s2}^\top \nabla \sigma_{c2}^\top \tilde{W}_{c2} + \nabla \varepsilon_{c2} \mathcal{F}_{s2}(z_2) \\ &- \frac{\nabla \varepsilon_{c2}}{2}\mathcal{G}_{s2}(z_2)r_2^{-1}\mathcal{G}_{s2}^\top \nabla \sigma_{c2}^\top W_{c2} + \frac{\nabla \varepsilon_{c2}}{2}\mathcal{G}_{s2}(z_2)r_2^{-1}\mathcal{G}_{s2}^\top \nabla \sigma_{c2}^\top \tilde{W}_{c2} \\ &+ \frac{1}{2}W_{c2}^\top \nabla \sigma_{c2} \mathcal{G}_{s2}(z_2)r_2^{-1}\tilde{\mathcal{G}}_{s2}^\top \nabla \sigma_{c2}^\top W_{c2} - \frac{1}{2}W_{c2}^\top \nabla \sigma_{c2} \mathcal{G}_{s2}(z_2)r_2^{-1}\tilde{\mathcal{G}}_{s2}^\top \nabla \sigma_{c2}^\top \tilde{W}_{c2} \\ &+ \frac{\nabla \varepsilon_{c2}}{2}\mathcal{G}_{s2}(z_2)r_2^{-1}\tilde{\mathcal{G}}_{s2}^\top \nabla \sigma_{c2}^\top W_{c2} - \frac{\nabla \varepsilon_{c2}}{2}\mathcal{G}_{s2}(z_2)r_2^{-1}\tilde{\mathcal{G}}_{s2}^\top \nabla \sigma_{c2}^\top \tilde{W}_{c2}\}. \end{aligned} \quad (80)$$

Using (11), we have  $W_{c2}^\top \nabla \sigma_{c2} \mathcal{F}_{s2}(z_2) - \frac{1}{2}W_{c2}^\top \nabla \sigma_{c2} \mathcal{G}_{s2}(z_2)r_2^{-1}\mathcal{G}_{s2}(z_2)W_{c2}\nabla \sigma_{c2} \leq -k\|z_2\|^2$ . Additionally, it is also assumed that the effect of the NN approximation error in the system dynamics is bounded above, i.e.,  $\nabla \varepsilon_{c2} \mathcal{F}_{s2}(z_2) - \frac{1}{2}\nabla \varepsilon_{c2} \mathcal{G}_{s2}(z_2)r_2^{-1}\mathcal{G}_{s2}(z_2)W_{c2}\nabla \sigma_{c2} \leq -\bar{\epsilon}\|z_2\|^2$ . One can write (80), as follows

$$J_2(z_2) = -(k + \bar{\epsilon})z_2^2 + k_1\tilde{W}_{c2} + k_2\tilde{W}_{c2} + k_3\tilde{\mathcal{G}}_{s2} + k_4\tilde{\mathcal{G}}_{s2} + k_5\tilde{W}_{c2}\tilde{\mathcal{G}}_{s2} + k_6\tilde{W}_{c2}\tilde{\mathcal{G}}_{s2}. \quad (81)$$

which can be further written as

$$j_2(z_2) = -\bar{k}_1 z_2^2 + \bar{k}_2 \tilde{W}_{c2} + \bar{k}_3 \tilde{\mathcal{G}}_{s2} + \bar{k}_4 \tilde{W}_{c2} \tilde{\mathcal{G}}_{s2}, \tag{82}$$

where  $\frac{1}{2} W_{c2}^T \nabla \sigma \mathcal{G}_{s2}(z_2) r_2^{-1} \mathcal{G}_{s2} \nabla \sigma \leq k_1$ ,  $\frac{\nabla \varepsilon}{2} \mathcal{G}_{s2}(z_2) r_2^{-1} \mathcal{G}_{s2} \nabla \sigma \leq k_2$ ,  $\frac{1}{2} W_{c2}^T \nabla \sigma \mathcal{G}_{s2}(z_2) r_2^{-1} W_{c2} \nabla \sigma \leq k_3$ ,  $\frac{\nabla \varepsilon}{2} \mathcal{G}_{s2}(z_2) r_2^{-1} W_{c2} \nabla \sigma \leq k_4$ ,  $\frac{1}{2} W_{c2}^T \nabla \sigma \mathcal{G}_{s2}(z_2) r_2^{-1} \nabla \sigma \leq k_5$ ,  $\frac{\nabla \varepsilon}{2} \mathcal{G}_{s2}(z_2) r_2^{-1} \nabla \sigma \leq k_6$ .

Consider the second and third term of (52)

$$\dot{L}_{22} + \dot{L}_{23} = tr\{e_{i2}^T \dot{e}_{i2}\} + tr\{\tilde{Z}_2^T \dot{\tilde{Z}}_2\} \tag{83}$$

On substituting the value of  $\dot{e}_{i2}$  and  $\dot{\tilde{Z}}_2$  from Theorem 1, one can write (83) as follows

$$\dot{L}_{22} + \dot{L}_{23} = tr\left\{e_{i2}^T(-K e_{i2} + \tilde{Z}_2 \sigma(\xi) \hat{u} + \varepsilon_I(\zeta)) + \tilde{Z}_2^T(-\alpha_v \tilde{Z}_2 - \sigma(\xi) \hat{u} e_{i2}^T + \alpha_v Z_2)\right\}. \tag{84}$$

which can be further simplified by using the cyclic property of traces as

$$\dot{L}_{22} + \dot{L}_{23} = tr\{e_{i2}^T(-K e_{i2} + \varepsilon_I(\zeta)) + \tilde{Z}_2^T(-\alpha_v \tilde{Z}_2 + \alpha_v Z_2)\}. \tag{85}$$

To simplify, we have  $\varepsilon_I^2(z_2) \leq \|z_2\|^2$ . Therefore, on using Young's inequality in  $e_{i2} z_2$ , we have

$$\dot{L}_{22} + \dot{L}_{23} \leq -K \|e_{i2}\|^2 - \alpha_v \|\tilde{Z}_2\|^2 + \|e_{i2}\|^2 + \|z_2\|^2 + \alpha_v \|\tilde{Z}_2^T Z_2\| \tag{86}$$

Consider the fourth term of (52),

$$\dot{L}_{24} = tr\left\{\tilde{W}_{c2}^T \left(\gamma_c \hat{\psi}_2(t)(-e_2(t) + \frac{1}{4} \hat{W}_{c2}^T \nabla \sigma_{c2} \wedge \nabla \sigma_{c2}^T \hat{W}_{c2} + \hat{W}_{c2}^T \nabla \sigma_{c2} \mathcal{F}_{s2}) + \sigma_{21} \hat{W}_{c2}\right)\right\} \tag{87}$$

Therefore, one can further simplify (87) by using Young's inequality in cross-product terms as follows

$$\dot{L}_{24} \leq c_1 \|\tilde{W}_{c2}\|^2 + c_2 \|\tilde{W}_{c2}\| + c_3 \|\tilde{Z}_2\|^2 + c_4 \|\tilde{W}_{c2}\| - c_5 \|\tilde{W}_{c2}\|^2 \tag{88}$$

Consider the fifth term of (52)  $\dot{L}_{25} = tr\{\tilde{W}_{a2}^T \dot{W}_{a2}\}$ . On using the weight update law from Theorem 2, one can write

$$\dot{L}_{25} = tr\left\{\tilde{W}_{a2}^T (\beta_a S_3(\beta_1 z_2 + \frac{1}{2} \hat{W}_{a2}^T \sigma_{a2}(\bar{z}_2) - \frac{1}{2} r_2^{-1} \hat{\mathcal{G}}_{s2}^T \nabla \sigma_{c2}^T \hat{W}_{c2}) + \sigma_{22} \hat{W}_{a2})\right\} \tag{89}$$

Using  $\tilde{W}_{c2} = W_{c2} - \hat{W}_{c2}$ ,  $\tilde{W}_{a2} = W_{a2} - \hat{W}_{a2}$ ,  $\tilde{\mathcal{G}}_{s2} = \mathcal{G}_{s2} - \hat{\mathcal{G}}_{s2}$

$$\dot{L}_{25} = tr\left\{\tilde{W}_{a2}^T (\beta_a S_3(\beta_1 z_2 + \frac{1}{2} (W_{a2} - \tilde{W}_{a2})^T \sigma_{a2}(\bar{z}_2) - \frac{1}{2} r_2^{-1} (\mathcal{G}_{s2} - \tilde{\mathcal{G}}_{s2})^T \nabla \sigma_{c2}^T (W_{c2} - \tilde{W}_{c2}) + \sigma_{22} \tilde{W}_{a2} (W_{a2} - \tilde{W}_{a2}))\right\} \tag{90}$$

Considering a last term, we can write

$$\tilde{W}_{a2} (W_{a2} - \tilde{W}_{a2}) \leq \|\tilde{W}_{a2} W_{a2}\| - \|\tilde{W}_{a2}\|^2$$

Using Young's inequality in the cross product terms, we can write

$$\dot{L}_{25} \leq c_5 \|\tilde{W}_{a2}\|^2 + c_6 \|z_2\|^2 + c_7 \|\tilde{Z}_2\|^2 + c_8 \|W_{c2}\|^2 + c_9 \|\tilde{Z}_2\| + c_{10} \|\tilde{W}_{c2}\| + c_{21} \|\tilde{W}_{a2}\| - c_{22} \|\tilde{W}_{a2}\|^2 \quad (91)$$

Combining (82), (86),(88) and (91) and simplifying, we have

$$\begin{aligned} \dot{L} &\leq -c_{z1} \left( \|z_1\| - \frac{k_{11}}{c_{z1}} \right)^2 - c_{z2} \left( \|z_2\| - \frac{k_{21}}{c_{z2}} \right)^2 \\ &\quad - C_{c1} \left( \|\tilde{W}_{c1}\| - \frac{k_{12}}{C_{c1}} \right)^2 - C_{c2} \left( \|\tilde{W}_{c2}\| - \frac{k_{22}}{C_{c2}} \right)^2 \\ &\quad - C_{a1} \left( \|\tilde{W}_{a1}\| - \frac{k_{13}}{C_{a1}} \right)^2 - C_{a2} \left( \|\tilde{W}_{a2}\| - \frac{k_{23}}{C_{a2}} \right)^2 \\ &\quad - C_{f1} \left( \|\tilde{W}_{f1}\| - \frac{k_{f1}}{C_{f1}} \right)^2 - C_{f2} \left( \|\tilde{W}_{f2}\| - \frac{k_{f2}}{C_{f2}} \right)^2 + \bar{C}_{02}, \end{aligned} \quad (92)$$

where  $\bar{C}_{02} = \frac{k_{11}^2}{c_{z1}} + \frac{k_{21}^2}{c_{z2}} + \frac{k_{12}^2}{C_{c1}} + \frac{k_{13}^2}{C_{a1}} + \frac{k_{23}^2}{C_{a2}} + \frac{k_{f1}^2}{C_{f1}} + \frac{k_{f2}^2}{C_{f2}}$ . Therefore, from (92), the bounds for  $\|z_k\|$ ,  $\|\tilde{W}_{ck}\|$ ,  $\|\tilde{W}_{ak}\|$ ,  $\|\tilde{W}_{fk}\|$  can be obtained as

$$\left\{ \begin{array}{l} \|z_1\| > \sqrt{\frac{\bar{C}_{01}}{c_{z1}} + \frac{k_{11}}{c_{zk}}} \quad or \quad \|z_2\| > \sqrt{\frac{\bar{C}_{02}}{c_{z2}} + \frac{k_{21}}{c_{z2}}} \quad or \\ \|\tilde{W}_{c1}\| > \sqrt{\frac{\bar{C}_{01}}{C_{c1}} + \frac{k_{12}}{C_{c1}}} \quad or \quad \|\tilde{W}_{c2}\| > \sqrt{\frac{\bar{C}_{02}}{C_{c2}} + \frac{k_{22}}{C_{c2}}} \quad or \\ \|\tilde{W}_{a1}\| > \sqrt{\frac{\bar{C}_{01}}{C_{a1}} + \frac{k_{13}}{C_{a1}}} \quad or \quad \|\tilde{W}_{a2}\| > \sqrt{\frac{\bar{C}_{02}}{C_{a2}} + \frac{k_{23}}{C_{a2}}} \\ or \quad \|\tilde{W}_{f1}\| > \sqrt{\frac{\bar{C}_{01}}{C_{f1}} + \frac{k_{f1}}{C_{f1}}} \quad or \quad \|\tilde{W}_{f2}\| > \sqrt{\frac{\bar{C}_{02}}{C_{f2}} + \frac{k_{f2}}{C_{f2}}}. \end{array} \right. \quad (93)$$

This demonstrates that the overall closed-loop system is bounded. Since  $\hat{\mathcal{G}}_{sj}$  is a function of the weight estimation error of the NN identifier, or  $\tilde{Z}_j$ , and from (93),  $\tilde{Z}_j$  is bounded; as a consequence,  $\hat{\mathcal{G}}_{sj}$  is bounded. From (93)  $\tilde{W}_{aj}$ ,  $\tilde{W}_{cj}$ , identification error and system state are bounded. As a result, the control input error  $u_{ej} = -\beta_j z_j - \frac{1}{2} \hat{W}_{aj} \sigma_{aj}(z_j) + \frac{1}{2} r_j^{-1} \mathcal{G}_{sj} \frac{dJ_j}{dz_j}$  becomes bounded with bound  $e_b$  which can be obtained by using the bounds from (93) and Assumption 4. Therefore, the actual control inputs are bounded close to their optimal values.

### Proof of Theorem 2

The convergence of weights for Task 1 remains in alignment with Theorem 1. For Task 2, an additional term emerges in the Lyapunov proof (92) due to the regularization penalty, denoted as

$$\gamma_{pen1_j} = \tilde{W}_j (\hat{W}_j - W_{A_j}^*), \quad (94)$$

where  $W_{A_j}^*$  is indicative of the optimized weights for the primary task, as verified to be bounded in Theorem 1.

Substituting  $\hat{W}_j = W_j^* - \tilde{W}_j$ , we can rearrange equation (94) to

$$\tilde{W}_j W_j^* - \tilde{W}_j^2 - \tilde{W}_j W_{A_j}^* \leq \|\tilde{W}_j\| \|W_j^*\| - \|\tilde{W}_j\|^2 + \|\tilde{W}_j W_{A_j}^*\|. \quad (95)$$

Employing Young’s inequality to the first and third terms of (95), we get

$$\begin{aligned} \|\tilde{W}_j\| \|W_j^*\| &\leq \frac{\|\tilde{W}_j\|^2}{2} + \frac{\|W_j^*\|^2}{2}, \\ \|\tilde{W}_j W_{A_j}^*\| &\leq \frac{\|\tilde{W}_j\|^2}{2} + \frac{\|W_{A_j}^*\|^2}{2}. \end{aligned}$$

Substituting  $\|\tilde{W}_j\| \|W_j^*\|$ ,  $\|\tilde{W}_j W_{A_j}^*\|$  back into (95) provides

$$\tilde{W}_j W_j^* - \tilde{W}_j^2 - \tilde{W}_j W_{A_j}^* \leq \frac{\|W_{A_j}^*\|^2}{2} + \frac{\|W_j^*\|^2}{2}. \tag{96}$$

Thus, the integration of this term into the proof solely modifies the error bound to  $\gamma_{pen_j} \leq \frac{\|W_j^*\|^2}{2} + \frac{\|W_{A_j}^*\|^2}{2}$ , without impacting the overarching stability of the system.

The aggregate contribution to the error bounds is calculated by adding  $\gamma_{pen_j}$  to (92), resulting in a comprehensive error bound  $C_j = \bar{C}_{0j} + \gamma_{pen_j}$ . It is, therefore, clear from the derived equations that the error bounds experience an increase as the weights deviate from their optimal points; however, the comprehensive stability of the system remains intact.