01 Jan 2023

# Optimal Adaptive Tracking Control Of Partially Uncertain Nonlinear Discrete-Time Systems Using Lifelong Hybrid Learning

Behzad Farzanegan

Rohollah Moghadam

Sarangapani Jagannathan
*Missouri University of Science and Technology*, sarangap@mst.edu

Pappa Natarajan



## Recommended Citation

B. Farzanegan et al., "Optimal Adaptive Tracking Control Of Partially Uncertain Nonlinear Discrete-Time Systems Using Lifelong Hybrid Learning," *IEEE Transactions on Neural Networks and Learning Systems*, Institute of Electrical and Electronics Engineers, Jan 2023.
The definitive version is available at https://doi.org/10.1109/TNNLS.2023.3301383

# Optimal Adaptive Tracking Control of Partially Uncertain Nonlinear Discrete-Time Systems Using Lifelong Hybrid Learning

Behzad Farzanegan, *Graduate Student Member, IEEE*, Rohollah Moghadam, *Senior Member, IEEE*, Sarangapani Jagannathan, *Fellow, IEEE*, and Pappa Natarajan

***Abstract*— This article addresses a multilayer neural network (MNN)-based optimal adaptive tracking of partially uncertain nonlinear discrete-time (DT) systems in affine form. By employing an actor–critic neural network (NN) to approximate the value function and optimal control policy, the critic NN is updated via a novel hybrid learning scheme, where its weights are adjusted once at a sampling instant and also in a finite iterative manner within the instants to enhance the convergence rate. Moreover, to deal with the persistency of excitation (PE) condition, a replay buffer is incorporated into the critic update law through concurrent learning. To address the vanishing gradient issue, the actor and critic MNN weights are tuned using control input and temporal difference errors (TDEs), respectively. In addition, a weight consolidation scheme is incorporated into the critic MNN update law to attain lifelong learning and overcome catastrophic forgetting, thus lowering the cumulative cost. The tracking error, and the actor and critic weight estimation errors are shown to be bounded using the Lyapunov analysis. Simulation results using the proposed approach on a two-link robot manipulator show a significant reduction in tracking error by 44% and cumulative cost by 31% in a multitask environment.**

***Index Terms*— Discrete-time (DT) concurrent learning, experience replay, hybrid learning, lifelong learning (LL), multilayer neural networks (MNNs), optimal tracking control (OTC).**

## I. Introduction

THE application of optimal control for nonlinear discrete-time (DT) systems has drawn significant interest in various practical engineering systems, including unmanned surface vehicles and robotic manipulators [1], [2], [3], [4] and others. The Hamiltonian–Jacobi–Bellman (HJB) equation is normally used to find an optimal control strategy for a nonlinear system. However, solving the HJB equation to find the optimal solution in closed form is still challenging [5]. Thus, iterative techniques utilizing adaptive dynamic programming (ADP) have been formulated to find the optimal control input [6].

The application of neural networks (NNs) within the ADP framework offers a robust approach to determining the optimal control policies for unknown nonlinear systems in a forward-in-time fashion. Numerous iterative techniques based on ADP have been developed for optimal adaptive control (OAC), primarily focused on regulation tasks [7], [8], [9], [10]. It has been demonstrated that these iterative techniques converge to an optimal solution as the number of iterations approaches infinity [11]. However, this long convergence process can be a limitation when implementing the control scheme in real-time scenarios. Despite this disadvantage, the traditional ADP framework utilizing NNs generates an online approximate optimal input for the regulation [12]. On the other hand, approximate OAC schemes for trajectory tracking require an additional feedforward term to be designed optimally, which appears to be difficult for uncertain systems.

Several studies [7], [8], [13], [14], [15], [16], [17], [18] have explored the use of NNs for approximate optimal trajectory tracking of DT systems in affine form. For instance, in [16], [17], and [18], the nonlinear optimal tracking scheme uses the concept of dynamics inversion to determine the feedforward term and solves the HJB equation to find the feedback one. The feedforward term of the optimal tracking control (OTC) strategy requires full system dynamic knowledge, while the feedback term is based on the optimal value function gradient. Moreover, ADP has been used to develop a time-driven OTC of nonlinear DT systems with uncertain dynamics in the input-affine form using the state vector history in [14], [18], and [19]. In [18], single-layer NN weights are adjusted based on Bellman and control input errors at sampling instants, resulting in a relaxed iterative method, although the feedforward term is not optimal.

Recently, approximate OAC schemes for tracking in continuous [20] and DT [2], [21] have been reported. By using an augmented system approach, which includes the tracking error and the desired trajectory dynamics, the feedforward

Manuscript received 2 May 2022; revised 3 February 2023 and 11 June 2023; accepted 24 July 2023. This work was supported in part by the Office of Naval Research Grant N00014-21-1-2232; in part by the Army Research Office Award under Grant W911NF-21-2-0260; and in part by the Intelligent Systems Center at Missouri University of Science and Technology, Rolla. *(Corresponding author: Rohollah Moghadam.)*

Behzad Farzanegan and Sarangapani Jagannathan are with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: b.farzanegan@mst.edu; sarangap@mst.edu).

Rohollah Moghadam is with the Department of Electrical and Electronic Engineering, California State University-Sacramento, Sacramento, CA 95819 USA (e-mail: moghadam@csus.edu).

Pappa Natarajan is with the Instrumentation Engineering, Madras Institute of Technology, Anna University, Chennai 600044, India (e-mail: npappa@annauniv.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2023.3301383.

Digital Object Identifier 10.1109/TNNLS.2023.3301383

term in an optimal manner has been generated in [2], [21], and [22]. To deal with the feedforward term, the steady-state control strategy associated with the desired trajectory has been employed in [22]. However, the actor and critic NN weight convergence requires to fulfill the persistency of excitation (PE) condition.

Since checking the PE condition in an online manner is complicated, concurrent learning has been adopted for adaptive control and parameter estimation [23], [24], [25], and optimal control of continuous-time (CT) systems [26]. This approach utilizes both current and recorded data stored in a stack, instead of relying on external noise, to satisfy the PE condition. Although OAC techniques using concurrent learning were presented in [25], [26], [27], and [28] for nonlinear CT and [29] DT systems without convergence and stability proof, no concurrent learning-based tracking schemes were introduced for nonlinear DT systems.

In addition, the controller must maintain robustness in the face of changes in the operating conditions when performing a variety of tasks. To accomplish this, the learning scheme should be lifelong, allowing the acquisition of new information without causing forgetting or interference. Lifelong learning (LL) has been widely studied in machine learning and NN literature [30], [31], [32], [33], [34], [35] using offline training. For instance, the elastic weight consolidation (EWC) approach [35] is a method for addressing the problem of catastrophic forgetting, a common issue when training NNs on tasks changing sequentially. The EWC solves catastrophic forgetting by introducing a penalty term function of the difference between the current task parameters and those learned for previous tasks.

The LL feature is necessary for NN-based trajectory tracking of dynamical systems since NN weights can differ significantly with tasks and the cost must be minimized with already executed ones during learning of new tasks. Recently, the EWC-based technique was utilized for the optimal regulation of nonlinear DT systems without stability proof [29]. To the best of our knowledge, the EWC technique has not beet attempted for the optimal adaptive tracking (OAT) of nonlinear DT systems.

This article presents a method for lifelong hybrid learning (LHL)- based OAT for nonlinear DT systems in affine form with unknown internal dynamics. The optimal value function is defined as a cost-to-go function of both the augmented state variable and control input. The proposed scheme obtains the optimal control input by using the recursive Bellman equation and the stationarity condition, which is defined in terms of the value function. In addition, the proposed OAT control scheme requires only the control coefficient matrix.

The output and hidden layer NN weights of the actor NN are adjusted at the sampling instants through control input errors obtained via the known input dynamics. The critic NN weights are adjusted not only once a sampling instant but also by a finite number of times in an iterative manner within the sampling instants through temporal difference error (TDE). This method of updating the critic NN weights, which includes both time-driven updates at sampling instants and additional iterative updates within sampling instants, is referred to as hybrid learning. The addition of iterative weight tuning within the sampling interval can result in faster convergence of the value functional to its optimal solution. The proposed method ensures closed-loop stability and significantly improves the rate of tuning the controller toward optimality.

To enable the LL controller capabilities, an online version of EWC term is integrated into the critic NN update law, and the overall stability is established. Next, the developed method is extended to include NNs with more than two-hidden layers. The control input and TDE errors are directly utilized to adjust the actor and critic NN weights, respectively, thus eliminating the common problem of vanishing gradients in gradient-based weight updates. In addition, the method relaxes the need for a PE condition by using a concurrent learning approach that utilizes data from both past and current sampling instants online in DT. The effectiveness of the proposed approach is illustrated through the provided simulation results.

This article presents the following contributions.

1) development of a novel OAT control scheme based on multilayer NN (MNN) hybrid learning that is a combination of traditional adaptive control and iterative technique to enhance the convergence rate of the approximated solution of the HJB equation to its optimal value, and enabling faster convergence of tracking error and NN weights, unlike policy/value iteration methods [6], [22], [36];
2) relaxation of the PE condition using replay buffer by storing the history, the addition of concurrent learning term in the critic MNN weight tuning, and verification of PE in an online manner as opposed to [27] and [28];
3) development of a novel online EWC approach using the Jacobian matrix of TDE for the critic MNN to attain LL unlike offline method with targets [35];
4) relaxation of basis function selection by using MNN with $N$ hidden layers and addressing the vanishing gradient problem to enable efficient learning and convergence, in contrast to [2];
5) demonstration of closed-loop stability of nonlinear DT systems using actor–critic MNN control framework with concurrent LHL through the Lyapunov analysis.

## II. Problem Formulation

In this section, the OAT control problem is formulated for an uncertain affine nonlinear DT system described by

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

where $x(k) \in \Re^n$ is the state vector at the sampling instant $k$, $u(k) \in \Re^m$ is the control input vector, $f(x(k)) \in \Re^n$ represents the uncertain internal dynamics, and $g(x(k)) \in \Re^{n\times m}$ denotes the known bounded smooth function, i.e., $\|g(x(k))\|_F \leqslant g_M$. Define the desired trajectory as

$$x_d(k+1) = h(x_d(k)) \tag{2}$$

where $x_d(k) \in \Re^n$ is the state trajectory that is bounded and $h(x_d(k)) \in \Re^n$ is an unknown nonlinear function of the reference trajectory. Utilizing (1) and (2), the tracking error is defined as

$$e(k) = x(k) - x_d(k). \tag{3}$$

The primary aim of the OAT is to determine the optimal control policy $u^*(k)$ minimizing the infinite horizon discounted value function $J(e(k))$ as

$$J(e(k)) = \sum_{j=k}^{\infty} \gamma^{j-k} L(e(j), u(j)) \tag{4}$$

with $\gamma$ as the discount factor. The cost-to-go function is defined as $L(e(k), u(k)) = e(k)^T Q e(k) + u(k)^T R u(k)$, where $Q$ and $R$ are, respectively, positive semidefinite and positive definite matrices.

The tracking error dynamics by using (3) is derived as

$$e(k+1) = f(e(k)+x_d(k)) + g(e(k)+x_d(k))u(k) - h(x_d(k)). \tag{5}$$

Define the new state vector composed of the reference trajectory and tracking error as $X_a(k) = [e(k)^T, x_d(k)^T]^T \in \Re^{2n}$. By utilizing the tracking error dynamics (5) and the reference trajectory (2), the augmented system dynamics become

$$X_a(k+1) = \begin{bmatrix} f(e(k)+x_d(k)) - h(x_d(k)) \\ h(x_d(k)) \end{bmatrix} + \begin{bmatrix} g(e(k)+x_d(k)) \\ 0 \end{bmatrix} u(k) \tag{6}$$

which conveniently can be expressed in the affine form as $X_a(k+1) = F(X_a(k)) + G(X_a(k))u(k)$ with $F(X_a(k)) = \begin{bmatrix} f(e(k)+x_d(k)) - h(x_d(k)) \\ h(x_d(k)) \end{bmatrix}$ and $G(X_a(k)) = \begin{bmatrix} g(e(k)+x_d(k)) \\ 0 \end{bmatrix}$. Then, the discounted cost function (4) can be formulated in terms of the augmented state vector $X_a$ as

$$J(X_a(k)) = \sum_{j=k}^{\infty} \gamma^{j-k} L(X_a(j), u(j)) \tag{7}$$

where $L(X_a(k), u(k)) = X_a(k)^T \bar{Q} X_a(k) + u(k)^T R u(k)$ is the utility function using the augmented state vector, with $\bar{Q} \in \Re^{2n \times 2n}$ defined as $\bar{Q} = \begin{bmatrix} Q & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix}$ a positive semidefinite matrix. Using (7), a recursive Bellman equation can be obtained as

$$J(X_a(k)) = L(X_a(k), u(k)) + \gamma J(X_a(k+1)). \tag{8}$$

Invoking (8) and Bellman's optimality principle give $J^*(X_a(k)) = \min_{u(k)}(L(X_a(k))) + \gamma J^*(F(X_a(k)) + G(X_a(k))u(k))$, the HJB equation is defined as

$$H(X_a, J, u) = \gamma J(X_a(k+1)) - J(X_a(k)) + L(X_a(k), u(k)). \tag{9}$$

Solving the HJB equation gives the optimal value function, $J^*(X_a(k))$. Therefore, the optimal control policy $u^*(X_a(k))$ can be achieved by employing the stationary condition as $\partial H/\partial u(k) = 0$, which leads to

$$u^*(k) = -\frac{\gamma}{2} R^{-1} G(X_a(k))^T \frac{\partial J^*(X_a(k+1))}{\partial (X_a(k+1))}. \tag{10}$$

The feedforward and feedback terms are generated by the optimal control input in (10). Nevertheless, computing the optimal control input is impossible because of the need for the future value of the augmented state variable [15]. Moreover, it is essential to use a discount factor in the value functional to prevent the value functional leading to infinity when the reference trajectory does not go to zero. Note that the feedforward term in the control policy depends on the reference trajectory. This implies that the quadratic term with respect to the control input does not converge to zero over time.

Moreover, since it is challenging to find an analytical solution to the DT HJB equation, this article introduces a novel hybrid learning approach by using a combination of time-driven update at the sampling instants and a fixed number of iterations within the sampling instants to approximate the value function employed to forge the optimal control input. Due to a finite sampling duration, a fixed number of iterations are used. As a consequence, the value function approximation is not as accurate as that of the case of using iterative technique, wherein an infinite number of iterations at each sampling instant is used. It will be shown that this hybrid learning approach is practical and appears to generate an enhanced approximated value function over time and converges faster over the time-driven approach.

As a result, the approximated control policy will converge faster to the optimal value. Increasing the number of iterations within the fixed sampling instants further improves the accuracy of the value functional approximation, whereas there is a tradeoff between practicality and optimality. The hybrid learning scheme development for generating the estimated control policy is given in Section III for enhanced value function estimation. The PE condition is relaxed by using a replay buffer that is normally utilized in concurrent learning. The following fact and the assumption are required to proceed.

*Assumption 1:* The state and desired trajectory vectors are measurable for all tasks. In other words, the augmented state vector, $X_a(k)$, is available [2], [22].

*Fact 1:* When the optimal control input is applied to the augmented system in (6), the resulting closed-loop system is bounded. Specifically, it ensures that $\|F(X_a(k)) + G(X_a(k))u^*(X_a(k))\| \leqslant \bar{k}\|X_a(k)\|$ holds, where $\bar{k}$ is a known constant [15].

The aforementioned fact is not limiting because the admissible control policy $u^*$ guarantees the closed-loop stability for the DT system in (6) [15]. Typically, $\bar{k}$ is found in the stability proof during the demonstration of boundedness.

## III. Lifelong Hybrid OAT

The OAT of a nonlinear DT system (1) by employing MNN is presented in this section. The optimal control input and the value function are estimated using two MNNs in an actor–critic framework. A novel hybrid updating law combined with concurrent learning for the critic NNs is presented to accelerate the convergence rate of the value function and optimal control policy and to relax the need for the PE condition. The hidden layers of the actor and critic NNs generate estimation errors observed in the optimal control policy and must be explicitly taken care of in the design and analysis. Next, the LL aspect is introduced, and the proposed work is extended to $N$ layers.

The value functional represented by (7) is estimated via the critic MNN as

$$J(X_a(k)) = w_c^T \phi_c\big(v_c^T \phi(X_a(k))\big) + \varepsilon_{jk} \tag{11}$$

where $v_c$ and $w_c$ denote the critic NN weights, $\varepsilon_{jk}$ presents the bounded function approximation error, and $\phi_c$ and $\phi$ denote the output and hidden layer activation functions, respectively. Besides, the optimal control input achieved in (10) can be approximated using a two-layer actor NN as

$$u(X_a(k)) = w_a^T \phi_a\big(v_a^T \phi(X_a(k))\big) + \varepsilon_{uk} \tag{12}$$

where $v_a$ denotes the first and $w_a$ represents the second-layer actor NN weights with $\varepsilon_{uk}$ as the NN approximation error. The output and hidden activation functions are $\phi_a$ and $\phi$, respectively.

*Assumption 2:* Assume that there exist positive constants $w_{cM}$, $v_{cM}$, $w_{aM}$, $v_{aM}$, $\varepsilon_{jM}$, $\varepsilon_{uk}$, and $\varepsilon'_{jM}$ such that $\|w_c\| \leqslant w_{cM}$, $\|v_c\| \leqslant v_{cM}$, $\|w_a\| \leqslant w_{aM}$, $\|v_a\| \leqslant v_{aM}$, $|\varepsilon_{jk}| \leqslant \varepsilon_{jM}$, $|\varepsilon_{uk}| \leqslant \varepsilon_{uM}$, and $\|\partial \varepsilon_{jK}/\partial X_a(k+1)\|_F \leqslant \varepsilon'_{jM}$ [2].

In Section III-A, the optimal control input and function value approximations utilizing two-layer NN with the augmented state vector as input are presented.

### A. Value Function Approximation via Concurrent Hybrid Learning

One can approximate the value function in (11) as

$$\hat{J}(X_a(k)) = \hat{w}_c^T \phi_c\big(\hat{v}_c^T \phi(X_a(k))\big) \tag{13}$$

where $\hat{J}(X_a(k))$ denotes the estimated value function and $\hat{w}_c^T$ and $\hat{v}_c^T$ are the actual critic MNN weights. Employing $\hat{J}(X_a(k))$ in (8) gives the TDE as

$$\begin{aligned}\mathcal{E}_{\mathrm{TD}}(k) = {} & L(X_a(k-1), u(X_a(k-1))) \\ & + \hat{w}_c^T \Delta\phi_c(X_a(k-1))\end{aligned} \tag{14}$$

where $\mathcal{E}_{\mathrm{TD}} \in \Re$ and $\Delta\phi_c(X_a(k-1)) = \gamma\phi_c(\hat{v}_c^T \phi(X_a(k))) - \phi_c(\hat{v}_c^T \phi(X_a(k-1)))$. The TDE (14) depends on the augmented state vector, which requires the tracking error and reference trajectory, unlike in the case of regulation, where the TDE depends on the system state alone.

Using (11) in (8) gives $L(X_a(k-1), u(X_a(k-1))) = w_c^T \phi_c(v_c^T \phi(X_a(k-1))) - \gamma w_c^T \phi_c(v_c^T \phi(X_a(k))) - \Delta\varepsilon_{jk}$ where $\Delta\varepsilon_{jk} = \gamma\varepsilon_{jk} - \varepsilon_{jk-1}$. Replacing $L(X_a(k-1), u(X_a(k-1)))$ in (14) yields

$$\begin{aligned}\mathcal{E}_{\mathrm{TD}}(k) = {} & w_c^T \phi_c\big(v_c^T \phi(X_a(k-1))\big) \\ & - \gamma w_c^T \phi_c\big(v_c^T \phi(X_a(k))\big) - \Delta\varepsilon_{jk} \\ & + \gamma \hat{w}_c^T \phi_c\big(\hat{v}_c^T \phi(X_a(k))\big) - \hat{w}_c^T \phi_c\big(\hat{v}_c^T \phi(X_a(k-1))\big).\end{aligned} \tag{15}$$

Adding and subtracting $\gamma w_c^T \phi_c(\hat{v}_c^T(k)\phi(X_a(k)))$ and $w_c^T \phi_c(\hat{v}_c^T(k)\phi(X_a(k-1)))$, and doing a few manipulations, the TDE (14) becomes

$$\begin{aligned}\mathcal{E}_{\mathrm{TD}}(k) = {} & -\tilde{w}_c^T(k)\Delta\phi_c(X_a(k-1)) \\ & + w_c^T\big[\gamma\tilde{\phi}_c(k) + \tilde{\phi}_c(k-1)\big] - \Delta\varepsilon_{jk}\end{aligned} \tag{16}$$

where $\tilde{w}_c = w_c - \hat{w}_c$ denotes the critic weight estimation error. $\tilde{\phi}_c(k) = \phi_c(\hat{v}_c^T(k)\phi(X_a(k))) - \phi_c(v_c^T \phi(X_a(k)))$. Substituting $\Pi(k) = \tilde{\phi}_c(k) + \tilde{\phi}_c(k-1)$ in (16) results in

$$\mathcal{E}_{\mathrm{TD}}(k) = -\tilde{w}_c^T(k)\Delta\phi_c(k-1) + w_c^T \Pi(k) - \Delta\varepsilon_{jk}. \tag{17}$$

We can rewrite (17) as

$$\mathcal{E}_{\mathrm{TD}}(k) = -\tilde{w}_c^T(k)\Delta\phi_c(k-1) + \varepsilon_B(t) \tag{18}$$

where $\varepsilon_B(t) = w_c^T \Pi(k) - \Delta\varepsilon_{jk}$. Since $\Delta\hat{\phi}_{ck-1} \leqslant \phi_M$, $\|\Delta\varepsilon_{jk}\| \leqslant \varepsilon_{JM}$, $w_c \leqslant w_{cM}$, and $\|\Pi(k)\| \leqslant \Pi_M$, we have $\|\varepsilon_B(t)\| < \varepsilon_{B\max}$ on the compact set. It is imperative to fulfill the PE condition to ensure the weight convergence in the critic NN toward their target values. To check the PE condition, the samples are recorded in the memory. The terms $L(x(k), u(k))$ and $\phi_c(k)$ are evaluated at $k_j$ as $L(x(k_j), u(k_j))$ and $\phi_c(k_j)$ to store in the experience replay buffer. Therefore, we have

$$\Delta\phi_{cj} = \gamma\phi_c(k_j) - \phi_c(k_j - 1) \tag{19}$$

and $L_j = L(x(k_j), u(k_j))$. We also define a new performance index, including the recent TDE and the stored TDE to tune the update law. Thus, the TDE at $k_j$ is defined as

$$\mathcal{E}_{\mathrm{TD}j}(k_j) = L_j + \hat{w}_c^T \Delta\phi_{cj}. \tag{20}$$

Now, to update the critic NN, a novel gradient descent-based concurrent learning weight tuning law is given as

$$\begin{cases} \hat{w}_c(k+1) \\ \quad = \hat{w}_c(k) \\ \qquad - \dfrac{\alpha_J \Delta\phi_c\big(\hat{v}_c^T(k)\phi(X_a(k))\big)\mathcal{E}_{\mathrm{TD}}^T(k)}{\Delta\phi_c^T\big(\hat{v}_c^T(k)\phi(X_a(k))\big)\Delta\phi_c\big(\hat{v}_c^T(k)\phi(X_a(k))\big) + 1} \\ \qquad - \alpha_J \displaystyle\sum_{j=1}^{l} \dfrac{\Delta\phi_{cj}}{\Delta\phi_{cj}^T\Delta\phi_{cj} + 1}\mathcal{E}_{\mathrm{TD}j}^T \\ \hat{v}_c(k+1) \\ \quad = \hat{v}_c(k) - \phi(X_a(k))\big(\hat{v}_c^T(k)\phi(X_a(k)) \\ \qquad + B_1 k_v \mathcal{E}_{\mathrm{TD}}(k)\big)^T - \displaystyle\sum_{j=1}^{l}\phi\big(X_a(k_j)\big)\big(B_1 k_v \mathcal{E}_{\mathrm{TD}j}\big)^T \end{cases} \tag{21}$$

where $B_1$ and $k_v$ are constant matrices of proper dimension. The learning rate is denoted as $\alpha_J$, which is a constant value. The experience replay buffer comprises sample data where subscript $j$ indicates the sample index in the buffer, with $j \in \{1, \ldots, l\}$. The experience replay buffer is constructed as

$$\Psi = \big[\Delta\bar{\phi}_{c1}, \ldots, \Delta\bar{\phi}_{cl}\big] \tag{22}$$

where $\Delta\bar{\phi}_{cj} = (\Delta\phi_{cj}/(\Delta\phi_{cj}^T\Delta\phi_{cj} + 1))$. Hence, the data recorded in $\Psi$ comprise linearly independent elements, which are equal to the number of neurons in (13). Thus, the rank of matrix $\Psi$ is equal to $m$. The experience replay buffer contains a fixed and known number of samples, denoted as $l$, such that $l > m$.

The following theorem presents the convergence of the estimated critic NN weights to the actual weights using the concurrent learning-based update law for two-layer NN and without requiring the PE condition.

*Theorem 1:* Given the augmented system (6) with the cost function (7), let the update law for critic NN weights with the concurrent learning be given by (21). Under the assumption that $X_a(k)$ is bounded, if $\Psi$ is full rank, then, $\tilde{w}_c = w_c - \hat{w}_c$ and $\tilde{v}_c = v_c - \hat{v}_c$ converge to the residual set $R_{sw} = \{\tilde{w}_c | \|\tilde{w}_c\| \leq c_w\}$ and $R_{sv} = \{\tilde{v}_c | \|\tilde{v}_c\| \leq c_v\}$, respectively, where $c_w > 0$ and $c_v > 0$ are constants, without requiring PE.

*Proof:* Refer to the Supplementary Materials. ■

It is worth noting that the critic NN output and hidden layer weights are adjusted at specific intervals using TDE. To enhance the convergence rate, the weights of the critic NN are adjusted not only at the sample instants but also multiple times within the intervals through an iterative process called hybrid learning. Only the critic weights are adjusted during the sampling intervals, while other system variables, such as TDE, remain unchanged. Thus, by invoking (21), the critic weights can be tuned iteratively within sampling intervals as

$$\begin{cases} \hat{w}_c^{i+1}(k+1) \\ \quad = \hat{w}_c^{i}(k) \\ \qquad - \dfrac{\alpha_J \Delta\phi_c\big(\hat{v}_c^{iT}(k)\phi(X_a(k))\big)\mathcal{E}_{\mathrm{TD}}^T(k)}{\Delta\phi_c^T\big(\hat{v}_c^{iT}(l)\phi(X_a(k))\big)\Delta\phi_c\big(\hat{v}_c^{iT}(k)\phi(X_a(k))\big)+1} \\ \qquad - \alpha_J \displaystyle\sum_{j=1}^{l} \dfrac{\Delta\phi_{cj}}{\Delta\phi_{cj}^T\Delta\phi_{cj}+1}\mathcal{E}_{\mathrm{TD}j}^T \\ \hat{v}_c^{i+1}(k+1) \\ \quad = \hat{v}_c^{i}(k) \\ \qquad - \phi(X_a(k))\big(\hat{v}_c^{iT}(k)\phi(X_a(k)) + B_1 k_v \mathcal{E}_{\mathrm{TD}}(k)\big)^T \\ \qquad - \displaystyle\sum_{j=1}^{l} \phi\big(X_a(k_j)\big)\big(B_1 k_v \mathcal{E}_{\mathrm{TD}j}\big)^T \end{cases} \tag{23}$$

with $i = 1, \ldots, \mathcal{L}$ being the iteration number and $\mathcal{L}$ as the total number of iterations during sampling intervals. It is assumed that the critic NN weight update law at the sampling instants $k$ is the initial value for the sampling intervals, i.e., $\hat{w}_c^1(k) = \hat{w}_c(k)$ and $\hat{v}_c^1(k) = \hat{v}_c(k)$. The subsequent theorem demonstrates that the estimated value functional is bounded by utilizing the initial admissible control policy, $u_0(k)$, and the novel hybrid weight tuning law in (23).

*Theorem 2:* Consider the augmented nonlinear DT system (6) and the value function (7) expressed in terms of cost-to-go function. Consider the critic NN weights tuning in (21) at the sampling instants and (23) within sampling intervals. Then, under the assumption that $X_a(k)$ is bounded, the approximated value functional (13) is uniformly ultimately bounded (UUB).

*Proof:* Refer to the Supplementary Materials. ■

*Remark 1:* The assumption that the augmented state vector, $X_a(k)$, is bounded will be relaxed in Theorem 3. Also, here, a fixed number of iterations are employed due to the finite sampling interval. However, a varying number of iterations can also be utilized as long as the iterations can be executed within the duration. These finite iterative weight tuning updates result in faster convergence of the approximated value functional toward the optimal solution, as shown through the Lyapunov analysis. An increase in the number of iterative updates improves the convergence rate of the approximated control policy to its optimal solution. The proposed hybrid learning method ensures the closed-loop stability and significantly improves the optimal control performance.

### B. Optimal Control Policy Approximation

To derive the optimal control scheme, a two-layer feedforward NN is utilized as the actor NN. Thus, the estimated control input is formulated as

$$\hat{u}(X_a(k)) = \hat{w}_a^T \phi_a\big(\hat{v}_a^T \phi(X_a(k))\big) \tag{24}$$

where $\hat{w}_a$ an $\hat{v}_a$ are the weights and $\phi_a$ and $\phi$ are the activation functions of the actor NN. Next, define the control input error as the difference between the estimated (24) and actual control inputs as

$$\begin{aligned} \tilde{u}(k) &= \hat{w}^T{}_a \phi_a\big(\hat{v}_a^T \phi(X_a(k))\big) \\ &\quad + \frac{\gamma}{2} R^{-1} G(X_a(k))^T \frac{\partial \phi_c\big(\hat{v}_c^T \phi(X_a(k+1))\big)^T}{\partial X_a(k+1)} \hat{w}_c. \end{aligned} \tag{25}$$

Using (11) and (12) in (10) yields

$$\begin{aligned} & w_a^T \phi_a\big(v_a^T \phi(X_a(k))\big) + \varepsilon_{uk} \\ &\quad = -\frac{\gamma}{2} R^{-1} G(X_a(k))^T \frac{\partial \phi_c\big(v_c^T \phi(X_a(k+1))\big)^T}{\partial X_a(k+1)} w_c \\ &\qquad - \frac{\gamma}{2} R^{-1} G(X_a(k))^T \frac{\partial \varepsilon_{jk+1}}{\partial X_a(k+1)}. \end{aligned} \tag{26}$$

Employing (26) in (25) renders

$$\begin{aligned} \tilde{u}(k) &= \hat{w}^T{}_a \phi_a\big(\hat{v}_a^T \phi(X_a(k))\big) \\ &\quad + \frac{\gamma}{2} R^{-1} G(X_a(k))^T \frac{\partial \phi_c\big(\hat{v}_c^T \phi(X_a(k+1))\big)^T}{\partial X_a(k+1)} \hat{w}_c \\ &\quad - w^T{}_a \phi_a\big(v_a^T \phi(X_a(k))\big) + \varepsilon_{uk} \\ &\quad - \frac{\gamma}{2} R^{-1} G(X_a(k))^T \frac{\partial \phi_c\big(v_c^T \phi(X_a(k+1))\big)^T}{\partial X_a(k+1)} w_c \\ &\quad - \frac{\gamma}{2} R^{-1} G(X_a(k))^T \frac{\partial \varepsilon_{jk+1}}{\partial X_a(k+1)}. \end{aligned} \tag{27}$$

We define the actor weight estimation error as $\tilde{w}_a = w_a - \hat{w}_a$. Adding and subtracting $w_a^T \phi_a(\hat{v}_a^T \phi(X_a(k)))$ and $(\gamma/2) R^{-1} G^T(X_a(k))(\partial \phi_c(\hat{v}_c^T \phi(X_a(k+1)))^T / \partial X_a(k+1)) w_c$ in (27) and after some simplifications, one has

$$\begin{aligned} \tilde{u}(k) &= -\tilde{w}_a^T(k)\phi_a(k) - w_a^T(k)\tilde{\phi}_a(k) \\ &\quad - \frac{\gamma}{2} R^{-1} G^T(X_a(k)) \frac{\partial \phi_c\big(\hat{v}_c^T(k) X_a(k+1)\big)^T}{\partial \phi(X_a(k))} \tilde{w}_c(k) \\ &\quad - \frac{\gamma}{2} R^{-1} G^T(X_a(k)) \frac{\partial \tilde{\phi}_c(k+1)}{\partial X_a(k+1)} w_c(k) - \tilde{\varepsilon}_{uk} \end{aligned} \tag{28}$$

where $\tilde{\phi}_a(k) = \phi_a(v_a^T(k)\phi(X_a(k))) - \phi_a(\hat{v}_a^T(k)\phi(X_a(k)))$, $\phi_a(k) = \phi_a(\hat{v}_a^T(k)\phi(X_a(k)))$ and $\tilde{\varepsilon}_{uk} = \varepsilon_{uk} + (\gamma/2) R^{-1} G^T(X_a(k))(\partial \varepsilon_{jk+1} / \partial X_a(k+1))$.

Since $\tilde{u}(X_a(k))$ can be measured, the actor NN weight updating laws are established as

$$\begin{cases} \hat{w}_a(k+1) \\ \quad = \hat{w}_a(k) - \dfrac{\alpha_u \phi_a\left(\hat{v}_a^T(k)\phi(X_a(k))\right)\tilde{u}^T(k)}{\left({\phi_a}^T\left(\hat{v}_a^T(k)\phi(X_a(k))\right)\phi_a\left(\hat{v}_a^T(k)\phi(X_a(k))\right)+1\right)} \\ [15pt]\hat{v}_a(k+1) \\ \quad = \hat{v}_a(k) + \phi(X_a(k))\left({\hat{v}^T}_a(k)\phi(X_a(k)) + B_2 k_v \tilde{u}(k)\right)^T \end{cases} \tag{29}$$

where $0 < \alpha_u < 1$. $B_2$ and $k_v$ are a positive learning rate parameter and matrix of suitable dimensions, respectively. Using (29), the weight estimation error dynamics for the actor NN is obtained as

$$\begin{cases} \tilde{w}_a(k+1) \\ \quad = \tilde{w}_a(k) - \dfrac{\alpha_u \phi_a\left(\hat{v}_a^T(k)\phi(X_a(k))\right)\tilde{u}^T(k)}{\left({\phi_a}^T\left(\hat{v}_a^T(k)\phi(X_a(k))\right)\phi_a\left(\hat{v}_a^T(k)\phi(X_a(k))\right)+1\right)} \\ [5pt]\tilde{v}_a(k+1) \\ \quad = \tilde{v}_a(k) + \phi(X_a(k))\left({\hat{v}^T}_a(k)\phi(X_a(k)) + B_2 k_v \tilde{u}(k)\right)^T. \end{cases} \tag{30}$$

*Remark 2:* Note that the control input error can be measured, provided that the control coefficient matrix $G(X_a(k))$ is known. To relax this requirement for the actor NN weight update law (30), an additional identifier NN can be used [29].

Note the actor NN weights are only adjusted at the sampling instant, unlike the critic NN. This implies that the control policy is updated once per sampling instant and applied to the nonlinear system. Next, the following theorem ensures that the actual control input remains close to the optimal solution.

*Theorem 3:* Given the augmented system (6) with the cost function (7), consider the critic NN weights tuning in (21) at the sampling instants and (23) within sampling intervals, whereas the actor NN weights are adjusted using (29) with PE condition that holds for the actor NN. Under initial admissible control input, $u_0(k)$, there exist constants $\alpha_u > 0$ and $\alpha_J > 0$ such that the augmented state $X_a(k)$, the critic NN estimation weight errors ($\tilde{w}_c$ and $\tilde{v}_c$), and the actor estimation weight errors ($\tilde{v}_a$ and $\tilde{w}_a$) are all UUB. The upper bounds for these estimation errors are given by $\|\tilde{w}_c\| \le b'_{w_c}$, $\|\tilde{v}_c\| \le b'_{v_c}$, $\|\tilde{w}_a\| \le b'_{w_a}$, and $\|\tilde{v}_a\| \le b'_{v_a}$, where $b'_{w_c}$, $b'_{v_c}$, $b'_{w_a}$, and $b'_{v_a}$ are positive constants. This assurance guarantees that the estimated control policy remains close to its optimal value.

*Proof:* Refer to the Supplementary Materials. ■

*Remark 3:* For critic NN weight convergence, the PE condition is essential and it is ensured by using concurrent learning. The actor NN also needs PE, which is satisfied by adding random noise with the estimated control input.

While the approach presented so far has merit, catastrophic forgetting of knowledge occurs when different tasks are executed causing a significant change in the system dynamics resulting in a change in NN weights. Since for an online adaptive NN control, the NN weights change incrementally, this assumption is not satisfied in the presence of changing tasks/operating conditions leading to a degradation in tracking performance unless it is mitigated. This happens with all online NN learning methods. In the following, the LL method, which has the capability of learning from a continuous stream of information, is presented to address this issue.

### C. LHL -Based Tracking

In this section, the LHL-based OAT control approach is proposed. The block diagram of the proposed method is shown in Fig. 1. Sharing information across tasks is one way to avoid forgetting previous knowledge. Since the critic NN evaluates the approximated cost function, which is then utilized to generate the optimal control input, the LL is exclusively applied to the critic NN. In this context, the output and hidden layer weights in the critic NN are adjusted to minimize the subsequent performance index

$$\mathcal{P} = \frac{1}{2}\mathcal{E}_{\mathrm{TD}}(k)^2 + \frac{\lambda_w}{2}||\hat{w}_c - \hat{w}_c^\star||^2_{F_w} + \frac{\lambda_v}{2}||\hat{v}_c - \hat{v}_c^\star||^2_{F_v} \tag{31}$$

where $\hat{w}_c^\star$ and $\hat{v}_c^\star$ are the constant weight matrices obtained at the end of the prior tasks, $\lambda_w$ and $\lambda_v$ determine the importance of the previous task to the recent one, and $F_w$ and $F_v$ are the Fisher information matrix (FIM) [32]. The regularizer terms $||\hat{w}_c - \hat{w}_c^\star||^2_{F_w}$ and $||\hat{v}_c - \hat{v}_c^\star||^2_{F_v}$ are defined as

$$\begin{aligned} ||\hat{w}_c - \hat{w}_c^\star||^2_{F_w} &= \left(\hat{w}_c - \hat{w}_c^\star\right)^\top F_w\left(\hat{w}_c - \hat{w}_c^\star\right) \\ ||\hat{v}_c - \hat{v}_c^\star||^2_{F_v} &= \sum_i\sum_j F_{v,ij} v_{ij}^2 \end{aligned}$$

where $v_{ij} = \hat{v}_{c,ij} - \hat{v}^\star_{c,ij}$, and $F_{v,ij}$, $\hat{v}_{c,ij}$, and $\hat{v}^\star_{c,ij}$ are, respectively, the $i$th and $j$th element of $F_v$, $\hat{v}_c$, and $\hat{v}_c^\star$. The FIM acts as a valuable metric to assess the informational value of a specific set of sample data, denoted as $\mathcal{D}$, in terms of certain parameters. Consider $p(\vartheta|\mathcal{D})$ as the probability density function and $L(\vartheta|\mathcal{D}) = \log(p(\vartheta|\mathcal{D}))$ as the log-likelihood function, with $\vartheta$ representing the relevance of the parameter in relation to the training data $\mathcal{D}$ of a given task. Thus, the FIM can be computed as [37]

$$F_\theta = \mathbb{E}\left[\frac{\partial L(\boldsymbol{\theta} \mid \mathcal{D})}{\partial \boldsymbol{\theta}}\left(\frac{\partial L(\boldsymbol{\theta} \mid \mathcal{D})}{\partial \boldsymbol{\theta}}\right)^\top\right].$$

In contrast, this article presents an innovative online technique to compute the FIM for a new task by utilizing input samples $\mathcal{D}$ from previous tasks. The estimation of FIM involves the computation of the Jacobian matrix, which is based on the input samples obtained from the previous tasks. More precisely, the FIM for the upcoming task is computed as $F_w = \mathbb{E}(\mathbf{J}\,\mathbf{J}^\top)$, where $\mathbf{J}$ is the Jacobian matrix that is equal to the derivative of TDE in (14) with respect to the critic NN weight $\hat{w}_c$ obtained as $\mathbf{J} = \Delta\phi_c(X_a(k-1))$. Therefore, the FIM can be calculated as

$$F_w = \mathbb{E}\left(\Delta\phi_c(X_a(k-1))\Delta\phi_c(X_a(k-1))^\top\right).$$

Similarly, for the hidden layer, we have

$$F_{v,ij} = \mathbb{E}\left(\left(\left(\gamma\left(\hat{w}_{c,i}\phi\left(X_{a,i}(k)\right)\right)\phi_c\left(\sum_j\left[\hat{v}_{c,ji}\phi\left(X_{a,j}(k)\right)\right]\right)\right)\right) - \left(\hat{w}_{c,i}\phi\left(X_{a,i}(k-1)\right)\right)\phi_c\left(\sum_j\left[\hat{v}_{c,ji}\phi\left(X_{a,j}(k-1)\right)\right]\right)\right)^2\right)$$

where $\hat{w}_{c,i}$ and $X_{a,i}(k)$ are the $i$th element of $\hat{w}_c$ and $X_a(k)$, respectively. This technique enables the efficient calculation of the FIM for a new task by utilizing input samples from previous tasks, resulting in improved computational efficiency.

The second and third terms in (31) are combined with the performance index to mitigate significant fluctuations in the estimated critic NN weights during the tuning process of the next task. Thus, the modified tuning law is written as

$$\begin{aligned}
&\hat{w}_c(k+1) \\
&= \hat{w}_c(k) \\
&\quad - \frac{\alpha_J \Delta\phi_c\left(\hat{v}_c^T(k)\phi(X_a(k))\right)}{\Delta\phi_c^T\left(\hat{v}_c^T(k)\phi(X_a(k))\right)\Delta\phi_c\left(\hat{v}_c^T(k)\phi(X_a(k))\right)+1}\mathcal{E}_{\mathrm{TD}}^T(k) \\
&\quad - \alpha_J\sum_{j=1}^{l}\frac{\Delta\phi_{cj}}{\Delta\phi_{cj}^T\Delta\phi_{cj}+1}\mathcal{E}_{\mathrm{TD}j}^T - \alpha_J\lambda_w F_w\left(\hat{w}_c(k)-\hat{w}_c^\star\right) \\
&\hat{v}_c(k+1) \\
&= \hat{v}_c(k) - \phi(X_a(k))\left(\hat{v}_c^T(k)\phi(X_a(k)) + B_1 k_v \mathcal{E}_{\mathrm{TD}}(k)\right)^T \\
&\quad - \sum_{j=1}^{l}\phi\left(X_a(k_j)\right)\left(B_1 k_v \mathcal{E}_{\mathrm{TD}j}\right)^T \\
&\quad - \lambda_v F_v\left(\hat{v}_c(k)-\hat{v}_c^\star\right)
\end{aligned} \tag{32}$$

and within sampling instants $i = 1, \ldots, \mathcal{L}$ as

$$\begin{aligned}
&\hat{w}_c^{i+1}(k+1) \\
&= \hat{w}_c^i(k) \\
&\quad - \frac{\alpha_J \Delta\phi_c\left(\hat{v}_c^{iT}(k)\phi(X_a(k))\right)\mathcal{E}_{\mathrm{TD}}^T(k)}{\Delta\phi_c^T\left(\hat{v}_c^{iT}(l)\phi(X_a(k))\right)\Delta\phi_c\left(\hat{v}_c^{iT}(k)\phi(X_a(k))\right)+1} \\
&\quad - \alpha_J\sum_{j=1}^{l}\frac{\Delta\phi_{cj}}{\Delta\phi_{cj}^T\Delta\phi_{cj}+1}\mathcal{E}_{\mathrm{TD}j}^T - \alpha_J\lambda_w F_w\left(\hat{w}_c^i(k)-\hat{w}_c^\star\right) \\
&\hat{v}_c^{i+1}(k+1) \\
&= \hat{v}_c^i(k) \\
&\quad - \phi(X_a(k))\left(\hat{v}_c^{iT}(k)\phi(X_a(k)) + B_1 k_v \mathcal{E}_{\mathrm{TD}}(k)\right)^T \\
&\quad - \sum_{j=1}^{l}\phi\left(X_a(k_j)\right)\left(B_1 k_v \mathcal{E}_{\mathrm{TD}j}\right)^T - \lambda_v F_v\left(\hat{v}_c^i(k)-\hat{v}_c^\star\right).
\end{aligned} \tag{33}$$

*Theorem 4:* Consider the augmented nonlinear DT system (6) and the value function (7) expressed in terms of cost-to-go function. Let $u_0(k)$ be any initial admissible control input and the modified critic NN with the concurrent weight update law and the LL terms in (32) and (33) be used to generate the actual control input. If the rank of the history stack matrix $m$ is equal to the number of neurons in (13), then $\tilde{w}_c = w_c - \hat{w}_c$ and $\tilde{v}_c = v_c - \hat{v}_c$ exponentially converge to the residual set $R_{sw} = \{\tilde{w}_c | \|\tilde{w}_c\| \le c_w\}$ and $R_{sv} = \{\tilde{v}_c | \|\tilde{v}_c\| \le c_v\}$, respectively, where $c_w > 0$ and $c_v > 0$ are constants. Then, there exist constants $\alpha_u > 0$ and $\alpha_J > 0$ such that all signals, including the augmented state $X_a(k)$, the estimation errors of the critic NN weights ($\tilde{w}_c$ and $\tilde{v}_c$), and the estimation errors of the actor weights ($\tilde{v}_a$ and $\tilde{w}_a$), are UUB. This assurance guarantees that the estimated control policy remains close to its optimal value.

*Proof:* Refer to the Supplementary Materials. ■

*Remark 4:* The LL improves the performance of an adaptive NN control in the presence of changing tasks or trajectories, which requires significant NN weight changes, by using a penalty term at each layer. The proposed LL approach is not limited to two or three tasks and can be utilized for any number of tasks. The first part of the critic NN weight update law is the same as the one from Section III-B, whereas the additional term has been included for LL. Though the bounds increase due to LL, simulation results verify the effectiveness of LL.

The extension of the proposed method to NN with more than two layers is provided next.

### D. Extension to Multilayer NN

Consider a critic MNN as

$$\hat{J}_k(X_a(k)) = \hat{w}_c^T\phi_c\left(\sum_{i=1}^{N-1}\hat{v}_{ci}^T\phi_{ci}(X_a(k))\right) \tag{34}$$

where $\hat{v}_{ci}$ represents the weights of the $i$th layer of the critic NN. The critic MNN weight update law is defined as

$$\begin{cases}
\hat{w}_c^{j+1}(k+1) \\
\quad = \hat{w}_c^j(k) \\
\quad - \dfrac{\alpha_J \Delta\phi_c\left(\sum_{i=1}^{N-1}\hat{v}_{ci}^{jT}(k)\phi_{ci}(k)\right)\mathcal{E}_{\mathrm{TD}}^T(k)}{\Delta\phi_c^T\left(\sum_{i=1}^{N-1}\hat{v}_{ci}^{jT}(k)\phi_{ci}(k)\right)\Delta\phi_c\left(\sum_{i=1}^{N-1}\hat{v}_{ci}^{jT}(k)\phi_{ci}(k)\right)+1} \\
\quad - \alpha_J\sum_{k=1}^{l}\dfrac{\Delta\phi_{ck}}{\Delta\phi_{ck}^T\Delta\phi_{ck}+1}\mathcal{E}_{\mathrm{TD}k}^T - \alpha_J\lambda_w F_w\left(\hat{w}_c^j(k)-\hat{w}_c^\star\right) \\
\hat{v}_{ci}^{j+1}(k+1) \\
\quad = \hat{v}_{ci}^j(k) - \phi_{ci}(k)\left(\hat{v}^{jT}{}_{ci}(k)\phi_{ci}(k)\right. \\
\qquad \left. + B_{ci}k_v\mathcal{E}_{\mathrm{TD}}(k)\right)^T - \sum_{k=1}^{l}\phi(X_a(t_k))(B_1 k_v \mathcal{E}_{\mathrm{TD}k})^T \\
\qquad - \lambda_v F_v\left(\hat{v}_{ci}{}^j(k)-\hat{v}_{ci}^\star\right)
\end{cases} \tag{35}$$

where $i = 1, \ldots, N-1$ and $j = 1, \ldots, \mathcal{L}$. $\phi_{ci}$ denotes the NN activation functions, $B_{ci}$ is an appropriate dimensional matrix, and $v_{ci}$ represents the critic NN weights of the $i$th layer. Using MNNs, the estimated control strategy can be formulated as

$$\hat{u}(k) = \hat{w}_a^T\phi_a\left(\sum_{i=1}^{N-1}\hat{v}_{ai}^T\phi_{ai}(X_a(k))\right) \tag{36}$$

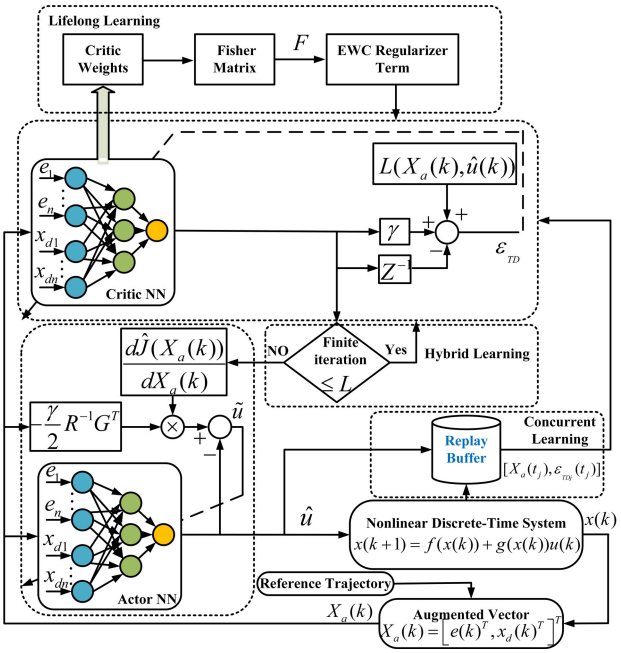


Fig. 1. Overall LHL-based OAT.

where $\hat{v}_{ai}$ is the weights of the $i$th layer of the actor NN. Similarly, select the actor NN update law as

$$\begin{cases} \hat{w}_a(k+1) \\ \quad = \hat{w}_a(k) \\ \quad - \dfrac{\alpha_u \phi_a\left(\sum_{i=1}^{N-1} \hat{v}_{ai}^T(k)\phi_{ai}(k)\right)\tilde{u}^T}{\phi_a^T\left(\sum_{i=1}^{N-1} \hat{v}_{ai}^T(k)\phi_{ai}(k)\right)\phi_a\left(\sum_{i=1}^{N-1} \hat{v}_{ai}^T(k)\phi_{ai}(k)\right)+1} \\ \hat{v}_{ai}(k+1) = \hat{v}_{ai}(k) + \phi_{ai}(k)\left(\hat{v}_{ai}^T(k)\phi_{ai}(k) + B_{ai}k_v\tilde{u}\right)^T \end{cases} \tag{37}$$

where $\phi_{ai}$ are the actor NN activation functions, $B_{ai}$ are design matrices with appropriate dimensions, and $v_{ai}$ are the $i$th layer NN weights with $i = 1, \ldots, N-1$. Unlike any gradient-based weight tuning where the errors at the output layer are propagated backward through the NN, from the above MNN critic and actor NN weight tuning laws (35) and (37), respectively, it is clear that TDE and control input errors are directly utilized. As a consequence, we can still use sigmoid activation functions, and basis function selection is not needed. Also, the TDE that is computed using tracking error is employed to tune the critic and actor NN weights for the OAT control technique.

*Theorem 5 (Estimated Optimal Control Using N Layer NN):* Suppose that $u_0(k)$ is an initial admissible control input for (6), and the value functional is described by (7). Let the critic and actor NN weights be given by (35) and (37). Then, the augmented state vector $X_a(k)$, and critic and action MNN weight estimation errors are all UUB under the PE condition of the actor NN. In addition, the estimated control input is bounded close to its optimal value.

*Proof:* The proof follows steps similar to Theorem 3. ■

*Remark 5:* Note that the control input and TDE errors depend on all the NN-layered outputs and augmented state vector. As a consequence, the performance of the overall system improves as demonstrated in the simulation.

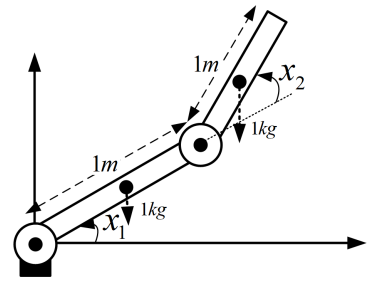


Fig. 2. Two-link robot manipulator.

*Remark 6:* Although the proposed LHL OAT approach requires the state vector, however, it can be extended to the output feedback by using an NN observer and by replacing the state vector with its estimated value from the observer as shown in [29].

In the following theorem, it is shown that the vanishing gradient issue, which is common with gradient-based methods, does not occur in the proposed approach.

*Theorem 6:* For the OAT scheme, let the critic NN weights be tuned by using (35), whereas the actor NN weights are updated using (37). Then, the vanishing gradient problem is not observed with the OAT scheme as we increase the number of hidden layers.

*Proof:* The proof follows as in the case of optimal adaptive regulation in [12] except, in this article, we use augmented state vector for the purpose of tracking. In the case of backpropagation, in the backward recursion for tuning the weights, the activation functions multiply together. Since the derivative of an NN activation function is less than one, multiplying the derivatives less than one over the number of layers becomes quite small.

In contrast, in this article, TDE, which depends on both tracking error and desired trajectory, is directly used for tuning the critic NN weights. As a result, the critic NN weights will not vanish or become small despite the addition of concurrent and LL terms. The derivative of the value function estimate with respect to the augmented vector determines the error in the control policy utilized to tune the actor NN weights. Since the critic NN weights are tuned by TDE, which do not become small or zero, they prevent the derivative of the activation functions of the action NN weights to be small. Thus, the vanishing gradient does not occur with our method. ■

## IV. Simulation Results

In this section, a simulation example of a robot manipulator, as shown in Fig. 2, is presented to show the effectiveness of the proposed approaches.

Consider a two-link robot manipulator [38] defined by

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= F(X_1, X_2) + M(X_1)^{-1}U \end{aligned} \tag{38}$$

where $X_1 = [x_1, x_2]^T$ represents the joint position, $X_2 = [x_3, x_4]^T$ denotes the joint velocities, and $U = [u_1, u_2]^T$

are the torque inputs for the joints. The nonlinear function is expressed as $F(X_1, X_2) = -[M(X_1)]^{-1}N(X_1, X_2)$ with

$$M(X_1) = \begin{bmatrix} 3 + 2\cos(x_2) & 1 + \cos(x_2) \\ 1 + \cos(x_2) & 1 \end{bmatrix} \tag{39}$$

and

$$N(X_1, X_2) = \begin{bmatrix} -(2x_3x_4 + x_4^2)\sin(x_2) + 19.6\cos(x_1) + 9.8\cos(x_1 + x_2) \\ x_1^2\sin(x_2) + 9.8\cos(x_1 + x_2) \end{bmatrix}. \tag{40}$$

Note that the system in (38) is a CT nonlinear system that is discretized. The time increment is set at 10 ms. The reference trajectory is defined as

$$\begin{bmatrix} x_{d_1}(k) \\ x_{d_2}(k) \\ x_{d_3}(k) \\ x_{d_4}(k) \end{bmatrix} = e^{(-k/4)} \begin{bmatrix} \sin(k) \\ \cos(k) \\ \cos(k) - \frac{1}{4}\sin(k) \\ -\sin(k) - \frac{1}{4}\cos(k) \end{bmatrix}. \tag{41}$$

Next, define the quadratic value function as (4) $L(X_a(k), u(X_a(k))) = X_a(k)^T \bar{Q} X_a(k) + u(X_a(k))^T R u(X_a(k))$ with $\bar{Q} \in \Re^{4\times 4}$ as $\bar{Q} = [\, Q \;\; 0_{4\times 4}; 0_{4\times 4} \;\; 0_{4\times 4} \,]$, with the selected value of $Q = I_4$ and $R = 0.01$. The initial state variables set as $x_0 = [\,0 \;\; 1 \;\; 1 \;\; 0\,]^T$, and the initial control policy is set to $u_0 = -\left[\begin{smallmatrix} 100 & 0 & 20 & 0 \\ 0 & 100 & 0 & 20 \end{smallmatrix}\right] e_0$. A four-layer NN with 11, 11, and 11 neurons in the output, hidden, and input layers is used for the critic NN, respectively. Similarly, the actor NN is a four-layer NN with 20, 20, and 20 neurons in the input, hidden, and output layers, respectively. The design parameters are chosen as $\gamma = 0.5$, $\alpha_u = 0.02$, and $\alpha_J = 0.08$. The value of $B_i$ is chosen as a constant vector of 0.01 with $B_{ci} \in \Re^{11}$ for critic NN with 11 hidden layer neurons and $B_{ai} \in \Re^{20}$ for actor NN with 20 hidden layer neurons.

The parameter selection in every NN-based controller depends on the system dynamics and the conditions derived in the theorems. Even though it has been considered that the system dynamics are uncertain, limited a priori information of the system helps in the reasonable selection of controller parameters and initial NN weights, thus satisfying the bounds required in the theorems. This process has been employed for selecting design parameters in this work. The hidden and output layer activation functions are selected as the sigmoid activation functions. The critic and actor weights are initialized at random within the interval [0, 1] and [−0.1, 0.1], respectively.

### A. Proposed MNN Hybrid Learning

To show the effect of variation in intersampling instants $\mathcal{L}$ on the OAT control, $\mathcal{L}$ is varied as 1, 3, and 10 for the nonlinear case. The case without the hybrid learning scheme, i.e., actor–critic-based OTC (AOTC) [2], is also included. While PE is relaxed by concurrent learning in critic NN, random noise with mean zero and variance 0.8 is used for 200-time instants with the estimated control policy of the actor NN to ensure the PE condition. It is seen from Figs. 3 and 4 that the proposed hybrid OAT helps in generating optimal control input over the existing AOTC approach [2] and enables both faster convergence of tracking error to near zero and NN weights after the removal of the PE signal.

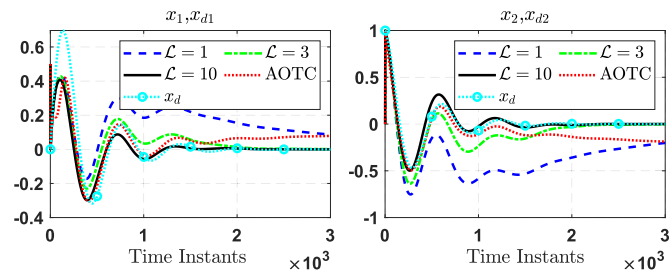


Fig. 3. Controller performance for the two-link robot with $\mathcal{L} = 1$, $\mathcal{L} = 3$, $\mathcal{L} = 10$ and the AOTC method [2].

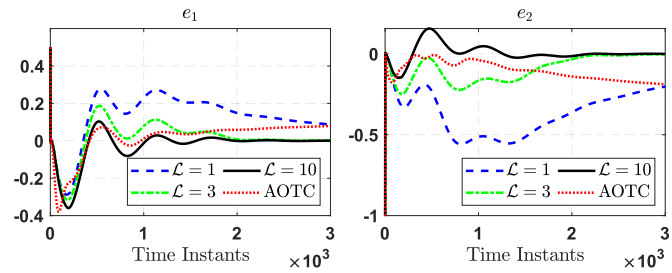


Fig. 4. Tracking errors for the two-link robot with $\mathcal{L} = 1$, $\mathcal{L} = 3$, $\mathcal{L} = 10$ and the AOTC method [2].

Simulations are carried out by varying the critic NN weight update within sampling instants by a factor represented as an intersampling instant rate $\mathcal{L}$ varied as one, three, and ten in the present work. Past feedback values are used to evaluate the value function and the control policy. The approximation error of the optimal value function is dependent on how frequently the critic NN weight matrix is updated within the sampling instants. However, although the existing method, without hybrid learning, shows an acceptable result for stability, its convergence rate is slow.

Hence, from Fig. 5, it is observed that with an increase in value function updates in the iterative fashion from one to ten, the approximation error and the convergence time for the state vector decrease, indicating that the value function and the state vector converge faster over recent literature AOTC in [2]. The performance of the proposed OAT approach using concurrent hybrid learning is better than AOTC by comparing TDE and cumulative cost.

### B. Lifelong Hybrid Learning

In this section, the novel update law (32) has been used in the simulation, first to relax the need for PE and then to improve the performance of the controller by utilizing the LL demonstrated via knowledge of the learned weights and acceleration of the convergence with lower cost. For this purpose, various desired trajectories are selected at distinct time points as $x_d(k) = e^{(-0.25k)}[\sin(k), \cos(k), \cos(k) - 0.25\sin(k), -\sin(k) - 0.25\cos(k)]^T, k \in (0, 3000]$; $x_d(k) = e^{(-0.25k)}[\sin(2k), \cos(2k), 2\cos(2k) -$

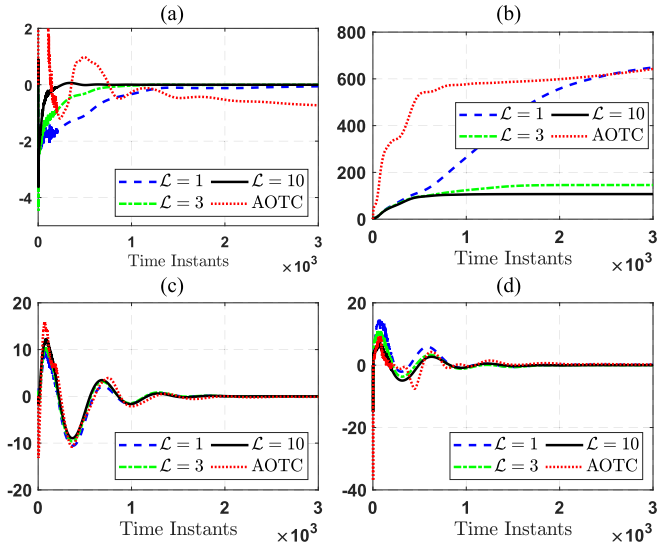


Fig. 5. Performance of the hybrid learning: (a) TDE, $\mathcal{E}_{\mathrm{TD}}$, (b) cumulative value function, (c) $\tilde{u}_1$, and (d) $\tilde{u}_2$.

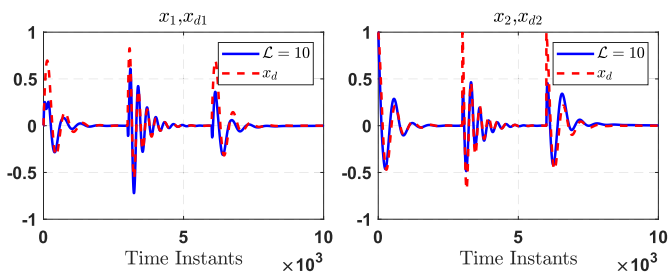


Fig. 6. System state and reference trajectories with the LHL.

$0.25\sin(2k), -2\sin(2k) - 0.25\cos(2k)]^T, k \in (3000, 6000]$; and $x_d(k) = e^{(-0.25k)}[\sin(k), \cos(k), \cos(k) - 0.25\sin(k), -\sin(k) - 0.25\cos(k)]^T, k \in (6000, 10\,000]$. As can be seen in Fig. 6, the desired trajectory, $x_d(k)$, consists of three tasks where the first and third tasks are identical.

To show the effectiveness of the experience replay method and LL, we respectively choose a two-layer NN with 11, 36, and one neurons in the hidden, input, and output layers for the critic NN. The hidden layer with tangent hyperbolic activation functions and the output layer with polynomial activation functions are chosen. We select $\mathcal{L} = 10$ for the hybrid factor. The design parameters are chosen as $\gamma = 0.5$, $\alpha_u = 0.02$, $\alpha_J = 0.01$, and $\lambda_i = 0.05$. The value of $B_i$ is chosen as constant vector of 0.01, with $B_{ci} \in \Re^{36}$ for critic NN with 36 hidden layer neurons and $B_{ai} \in \Re^{20}$ for actor NN with 20 hidden layer neurons.

Probing noise is added to the actor output to generate the PE signal, whereas concurrent learning ensures PE for the critic NN. The state and reference trajectories are shown in Fig. 6. As can be seen, the proposed LHL-based OAT control strategy achieves a high level of tracking performance. It is clear that all states under the LHL control input follow the desired reference signals. Also, the results confirm that the proposed method can ensure stability. In Figs. 7–9, the simulation results are shown for two distinct cases. In the first case, the LL term in (32)

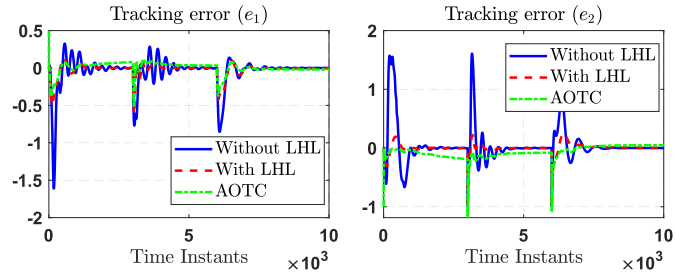


Fig. 7. Performance of the proposed LHL approach, without LHL [21], and AOTC [2].

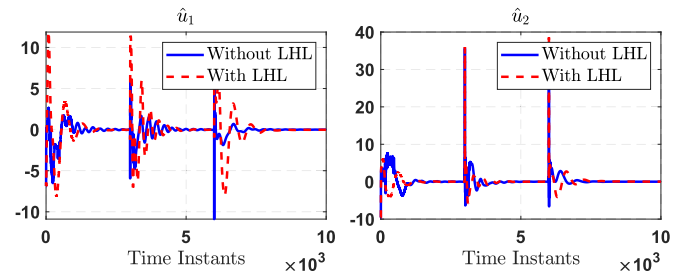


Fig. 8. Estimated control input with and without LHL term [21].

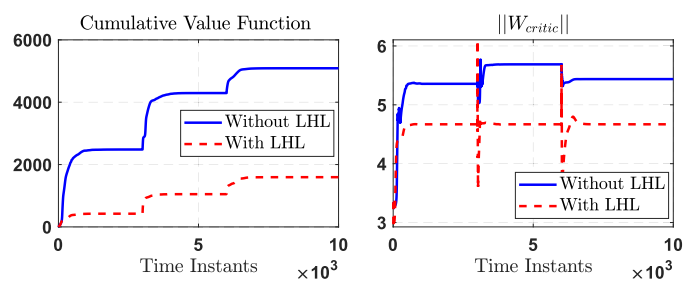


Fig. 9. Total cost and critic NN weight norm comparison with LHL and without LHL term [21].

and the hybrid learning term in (23) are not considered [21]. On the other hand, the second case incorporates the LHL term, and the corresponding results are presented.

The convergence of tracking errors is observed in Fig. 7. Without the LHL approach, there is a noticeable increase in error and significant fluctuations during the task transitions from task 1 to task 2 and back to task 1. These fluctuations occur due to the presence of the catastrophic forgetting issue. However, by integrating LHL techniques, the impact of catastrophic forgetting during task transitions is mitigated. In particular, when comparing the results to those without the LHL method, the proposed LHL strategy demonstrates a remarkable reduction in the root-mean-square (rms) error, specifically by 44% for the third task. These findings indicate that the LHL approach effectively addresses the issue of catastrophic forgetting, leading to improved tracking performance and reduced error, particularly during task transitions.

In Fig. 8, the estimated control actions are depicted. Realizing that the presented hybrid controller technique for the critic NN does not necessitate the PE condition, however, external noise is introduced as part of the approach for the actor NN. Fig. 9 clearly shows that the proposed LHL control strategy not only achieves superior tracking performance but

also significantly reduces costs and control effort. When compared to the scenario without the LHL method [21], the results indicate a remarkable reduction in total cost, amounting to 68% across all three tasks. Particularly for the third task, the proposed strategy showcases a notable cost reduction of 31%. These findings highlight the effectiveness and efficiency of the LHL control strategy in minimizing costs and control efforts while achieving the desired tracking objectives. In Fig. 9, the norm of the critic NN weights is also illustrated. It is clear that by using the LL method, the NN controller overcomes forgetting the previously accumulated knowledge from the former tasks when executing the newer ones.

## V. Conclusion

In this article, a novel LHL-based OAT control strategy was presented for nonlinear DT systems with uncertain internal dynamics. Applying TDE to adjust the weights of the critic and employing the control input errors to tune the weights of the actor led to a desirable performance outcome. The hybrid learning approach proposed for tuning the critic NN weights helped improve the value function convergence toward its optimal value. Thus, the control input error decreased, and the actual control input approached its optimal solution faster. Moreover, the experience replay method was utilized to assure the convergence of the critic NN weights. This was done by using a simple condition that can be verified online, unlike traditional PE conditions.

In addition, additional layers in the critic NN reduced NN functional approximation errors and resulted in better tracking performance. The vanishing gradient problem was not observed as the TDE and control policy errors were directly utilized for weight tuning. Despite the strong control performance, the multilayer NN-based controller was unable to effectively perform well in the presence of accumulated knowledge resulting from multiple tasks and its associated changes in dynamics. The EWC term, which was included in the critic NN weight tuning, mitigated the problem and led to LL. The ultimate boundedness of the overall closed-loop system was proven via Lyapunov stability analysis.

## References


[1] L. Yuan, T. Li, S. Tong, Y. Xiao, and Q. Shan, "Broad learning system approximation-based adaptive optimal control for unknown discrete-time nonlinear systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 8, pp. 5028–5038, Aug. 2022.

[2] B. Kiumarsi and F. L. Lewis, "Actor–critic-based optimal tracking for partially unknown nonlinear discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 140–151, Jan. 2015.

[3] S. Li, L. Ding, H. Gao, Y.-J. Liu, L. Huang, and Z. Deng, "ADP-based online tracking control of partially uncertain time-delayed nonlinear system and application to wheeled mobile robots," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3182–3194, Jul. 2020.

[4] H. Li, Y. Wu, and M. Chen, "Adaptive fault-tolerant tracking control for discrete-time multiagent systems via reinforcement learning algorithm," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1163–1174, Mar. 2021.

[5] Q. Wei, L. Zhu, T. Li, and D. Liu, "A new approach to finite-horizon optimal control for discrete-time affine nonlinear systems via a pseudolinear method," *IEEE Trans. Autom. Control*, vol. 67, no. 5, pp. 2610–2617, May 2022.

[6] Q. Wei, F. L. Lewis, D. Liu, R. Song, and H. Lin, "Discrete-time local value iteration adaptive dynamic programming: Convergence analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 6, pp. 875–891, Jun. 2018.

[7] E. N. Sanchez and F. Ornelas-Tellez, *Discrete-Time Inverse Optimal Control for Nonlinear Systems*. Boca Raton, FL, USA: CRC Press, 2017.

[8] G. Xiao, H. Zhang, and Y. Luo, "Online optimal control of unknown discrete-time nonlinear systems by using time-based adaptive dynamic programming," *Neurocomputing*, vol. 165, pp. 163–170, Oct. 2015.

[9] F. L. Lewis and D. Liu, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, vol. 17. Hoboken, NJ, USA: Wiley, 2013.

[10] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2042–2062, Jun. 2018.

[11] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[12] R. Moghadam, P. Natarajan, and S. Jagannathan, "Online optimal adaptive control of partially uncertain nonlinear discrete-time systems using multilayer neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4840–4850, Sep. 2022.

[13] H. Dong, X. Zhao, and B. Luo, "Optimal tracking control for uncertain nonlinear systems with prescribed performance via critic-only ADP," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 561–573, Jan. 2022.

[14] D. Wang, D. Liu, and Q. Wei, "Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach," *Neurocomputing*, vol. 78, no. 1, pp. 14–22, Feb. 2012.

[15] T. Dierks and S. Jagannathan, "Online optimal control of affine nonlinear discrete-time systems with unknown internal dynamics by using time-based policy update," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1118–1129, Jul. 2012.

[16] Y. Huang and D. Liu, "Neural-network-based optimal tracking control scheme for a class of unknown discrete-time nonlinear systems using iterative ADP algorithm," *Neurocomputing*, vol. 125, pp. 46–56, Feb. 2014.

[17] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[18] T. Dierks and S. Jagannathan, "Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics," in *Proc. 48h IEEE Conf. Decis. Control (CDC) Held Jointly With 28th Chin. Control Conf.*, Dec. 2009, pp. 6750–6755.

[19] H. Zhang, Y. Luo, and D. Liu, "Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, Sep. 2009.

[20] R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon, "Approximate optimal trajectory tracking for continuous-time nonlinear systems," *Automatica*, vol. 51, pp. 40–48, Jan. 2015.

[21] R. Moghadam, P. Natarajan, and S. Jagannathan, "Multilayer neural network-based optimal adaptive tracking control of partially uncertain nonlinear discrete-time systems," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 2204–2209.

[22] D. Wang, M. Ha, and L. Cheng, "Neuro-optimal trajectory tracking with value iteration of discrete-time nonlinear dynamics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4237–4248, Aug. 2021.

[23] G. Chowdhary and E. Johnson, "Concurrent learning for convergence in adaptive control without persistency of excitation," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, Dec. 2010, pp. 3674–3679.

[24] R. Kamalapurkar, B. Reish, G. Chowdhary, and W. E. Dixon, "Concurrent learning for parameter estimation using dynamic state-derivative estimators," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3594–3601, Jul. 2017.

[25] C. Li, F. Liu, Y. Wang, and M. Buss, "Concurrent learning-based adaptive control of an uncertain robot manipulator with guaranteed safety and performance," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 5, pp. 3299–3313, May 2022.

[26] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, Jan. 2014.

[27] M. Lin, B. Zhao, and D. Liu, "Policy gradient adaptive critic designs for model-free optimal tracking control with experience replay," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 6, pp. 3692–3703, Jun. 2022.

[28] B. Luo, Y. Yang, and D. Liu, "Adaptive $Q$-learning for data-based optimal output regulation with experience replay," *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3337–3348, Dec. 2018.

[29] R. Moghadam, B. Farzanegan, S. Jagannathan, and P. Natarajan, "Optimal adaptive output regulation of uncertain nonlinear discrete-time systems using lifelong concurrent learning," in *Proc. IEEE 61st Conf. Decis. Control (CDC)*, Dec. 2022, pp. 2005–2010.

[30] A. Saha, P. Rai, H. Daumé III, and S. Venkatasubramanian, "Online learning of multiple tasks and their relationships," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, in Proceedings of Machine Learning Research, vol. 15, G. Gordon, D. Dunson, and M. Dudík, Eds., Fort Lauderdale, FL, USA, Apr. 2011, pp. 643–651.

[31] H. B. Ammar, E. Eaton, P. Ruvolo, and M. E. Taylor, "Online multi-task learning for policy gradient methods," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, vol. 32, 2014, pp. 1206–1214.

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

[33] P. Ruvolo and E. Eaton, "ELLA: An efficient lifelong learning algorithm," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 507–515.

[34] Z. Chen and B. Liu, "Lifelong machine learning, second edition," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 12, no. 3, pp. 1–207, Aug. 2018.

[35] K. James et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[36] J. Xu, J. Wang, J. Rao, Y. Zhong, and H. Wang, "Adaptive dynamic programming for optimal control of discrete-time nonlinear system with state constraints based on control barrier function," *Int. J. Robust Nonlinear Control*, vol. 32, no. 6, pp. 3408–3424, Apr. 2022.

[37] L. Meng, "Method for computation of the Fisher information matrix in the expectation-maximization algorithm," 2016, *arXiv:1608.01734*.

[38] F. L. Lewis, S. Jagannathan, and A. Yesildirak, *Neural Network Control of Robot Manipulators and Non-Linear Systems*. Boca Raton, FL, USA: CRC Press, 1998.


**Behzad Farzanegan** (Graduate Student Member, IEEE) received the B.Sc. degree in biomedical engineering and the B.Sc. and M.Sc. degrees in electrical engineering from the Amirkabir University of Technology, Tehran, Iran, in 2011, 2012, and 2013, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Missouri University of Science and Technology, Rolla, MO, USA, with a research focus on optimal control, adaptive control, reinforcement learning, and autonomous/robotic systems.

**Rohollah Moghadam** (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the Missouri University of Science and Technology, Rolla, MO, USA, in 2020.

He was a Visiting Scholar with The University of Texas at Arlington Research Institute, Arlington, TX, USA, in 2016. He was an Assistant Professor of electrical engineering at Arkansas Tech University, Russellville, AR, USA, from 2020 to 2021, and PennWest California, California, PA, USA, in 2022. He is currently an Assistant Professor with the Department of Electrical and Electronic Engineering, California State University at Sacramento, Sacramento, CA, USA. His research interests include cyber–physical systems, reinforcement learning, robot decision and control, cooperative control systems, time-delay systems, and neural networks.

**Sarangapani Jagannathan** (Fellow, IEEE) is currently a Rutledge-Emerson Distinguished Professor of electrical and computer engineering at the Missouri University of Science and Technology, Rolla, MO, USA. He served as the Director of the NSF Industry/University Cooperative Research Center on Intelligent Maintenance Systems, Missouri University of Science and Technology, for 13 years. He has coauthored 191 peer-reviewed journal articles and 296 refereed IEEE conference articles, authored several book chapters, and authored/co-edited six books. He has received 21 U.S. patents and one patent defense publication. He has graduated 32 Ph.D. and 31 M.S. thesis students. He has co-edited the Institution of Engineering and Technology (IET) book series on control from 2010 until 2013. His research interests include learning, adaptation and control, secure human–cyber–physical systems, prognostics/bigdata analytics, and autonomous systems/robotics.

Dr. Jagannathan is a fellow of the National Academy of Inventors, the Institute of Measurement and Control, U.K., and the IET, U.K. He is serving on many editorial boards, including the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He has been on organizing committees of several IEEE conferences. He received many awards, including the 2000 NSF Career Award, the 2001 Caterpillar Research Excellence Award, the 2007 Boeing Pride Achievement Award, and the 2018 IEEE CSS Transition to Practice Award.

**Pappa Natarajan** received the B.E. degree from Annamalai University, Chidambaram, India, in 1990, the M.Tech. degree from the Cochin University of Science and Technology, Kochi, India, in 1992, and the Ph.D. degree from Anna University, Chennai, India, in 2003.

She was a Fulbright Scholar with the Missouri University of Science and Technology, Rolla, MO, USA, from August 2019 to March 2020. She is currently a Professor with the Department of Instrumentation Engineering, Madras Institute of Technology (MIT), Anna University. She has published in reputed international journals and conferences. Her research interests include artificial neural networks (ANNs), image-based measurement, and advanced controller design.