01 Jan 2023

Optimal Tracking Of Nonlinear Discrete-time Systems Using Zero-Sum Game Formulation And Hybrid Learning

Behzad Farzanegan

S. (Sarangapani) Jagannathan
Missouri University of Science and Technology, sarangap@mst.edu

Recommended Citation

B. Farzanegan and S. Jagannathan, "Optimal Tracking Of Nonlinear Discrete-time Systems Using Zero-Sum Game Formulation And Hybrid Learning," *Proceedings of the American Control Conference*, pp. 2715 - 2720, Institute of Electrical and Electronics Engineers, Jan 2023.
The definitive version is available at https://doi.org/10.23919/ACC55779.2023.10156305

IEEE Control Systems Letters paper presented at
2023 American Control Conference (ACC)
San Diego, CA, USA. May 31 - June 2, 2023

Optimal Tracking of Nonlinear Discrete-time Systems using Zero-Sum Game Formulation and Hybrid Learning

Behzad Farzanegan[1] and S. Jagannathan[1]

Abstract— **This paper presents a novel hybrid learning-based optimal tracking method to address zero-sum game problems for partially uncertain nonlinear discrete-time systems. An augmented system and its associated discounted cost function are defined to address optimal tracking. Three multi-layer neural networks (NNs) are utilized to approximate the optimal control and the worst-case disturbance inputs, and the value function. The critic weights are tuned using the hybrid technique, whose weights are updated once at the sampling instants and in an iterative manner over finite times within the sampling instants. The proposed hybrid technique helps accelerate the convergence of the approximated value functional to its actual value, which makes the optimal policy attain quicker. A two-layer NN-based actor generates the optimal control input, and its weights are adjusted based on control input errors. Moreover, the concurrent learning method is utilized to ease the requirement of persistent excitation. Further, the Lyapunov method investigates the stability of the closed-loop system. Finally, the proposed method is evaluated on a two-link robot arm and demonstrates promising results.**

Index Terms— **Discrete-time concurrent learning, experience replay, zero-sum game formulation, optimal tracking control.**

I. Introduction

Zero-sum games, which have drawn extensive attention from researchers, appear widely in nonlinear discrete-time (DT) systems. Generally, the primary aim is to design an optimal feedback controller to minimize the user-defined performance index comprising two penalties corresponding to the system state and control signals while maximizing the penalty related to disturbances. In practice, there are external disturbances that weaken the closed-loop system performance. The optimal adaptive control (OAC) as a zero-sum game (ZSG) has been investigated to achieve the best control performance in the presence of external disturbances [1].

The optimal control policies for nonlinear systems, which are subject to disturbances, are obtained by solving the nonlinear PDE referred to as Hamilton–Jacobi–Isaacs (HJI) equation. Adaptive dynamic programming (ADP) is a robust forward-in-time framework to solve the HJI equation approximately.

The optimal control schemes are also discussed in the game theoretical framework [2]. In [3], reinforcement learning (RL) has constructed an iterative method to find the optimal control policies for a quadratic zero-sum game of unknown nonaffine nonlinear systems. In [4], a neural network (NN)-based online simultaneous update policy algorithm (SUPA) has been proposed to solve the HJI equation, and an integral RL has been utilized to ease the need for internal dynamics. Unlike SUPA, the authors [1] utilized policy iterations on two players by the Gauss-Newton method to find optimal solution.

To deal with the DT zero-sum game, the authors in [5] have proposed an event triggered control strategy using gradient descent ADP. In [6], a heuristic ADP algorithm has been proposed to handle the ZSG problem for nonlinear DT systems. Three NNs were employed to address the HJB equation associated with H_∞ optimal regulation control problem. In [7], two ADP approaches, comprised of an iterative offline learning procedure and a modified gradient-descent-based online method, have been presented for DT multi-player games.

Like traditional adaptive control, a persistence of excitation (PE) condition is needed to guarantee convergence and boundedness of the value function NN weights [8]. In [9], [10], [11], concurrent learning has been introduced and utilized to eliminate the requirement for the PE condition by adopting an easy-to-check and online condition. Instead of applying external noise to fulfill the PE condition, both current and past data saved in a replay buffer are used at the same time [10], [11]. In [10], [11], a concurrent learning-based single-layer NN optimal control has been employed for nonlinear continuous-time systems. The authors in [9] have proposed an optimal adaptive control method that incorporates concurrent learning for discrete systems. However, they did not provide evidence of its stability and convergence.

In contrast, this paper presents a novel optimal adaptive tracking (OAT) scheme to solve the ZSG problems for partially uncertain DT nonlinear systems using a two-layer NN-based ADP framework and a hybrid-learning strategy. Two-layer NN is an efficient function approximator and does not require explicit selection of basis functions. In contrast, the analysis and closed-loop stability are involved, and this aspect is addressed in this effort. First, the original system and its cost function are augmented with the desired trajectory to address optimal tracking via ZSG formulation.

Next, three two-layer NNs are employed to evaluate the value functional, the optimal control input, and the worst-case disturbance. The temporal difference error (TDE) is defined based on the actual and approximated value functional by using two-layer NN in comparison with single-layer NN in the literature. The weights of the critic are adjusted using the hybrid technique through instantaneous TDE. This involves tuning the weights at the sampling instants once, and then adjusting them a finite number of times within the sampling instants. The suggested hybrid tuning method aids in speeding

[1]B. Farzanegan and S. Jagannathan are with the Dept. of Elec. and Comp. Engg, Missouri University of Science and Technology, Rolla, MO, USA. b.farzanegan@mst.edu and sarangap@mst.edu.

This work has been supported 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.

up the convergence of the estimated value function toward its optimal value. Moreover, the concurrent learning term is included to ease the requirement of the PE condition. Lastly, we demonstrate that the errors in tracking and weight estimation of the multi-layer NNs (MNN) used for the critic and actor are proven to be uniformly ultimately bounded (UUB) through the use of Lyapunov's direct method.

In short, this article's contributions are as follows:

- To address the HJI equation in the context of tracking, a combination of traditional adaptive control and iterative technique is employed, known as hybrid learning. This approach differs from relying solely on policy iteration techniques, as seen in [1].
- Using the concurrent learning terms relaxes persistent excitation in the critic MNN weight update laws in contrast to [10], [11]
- Unlike [9], the actor-critic framework that employs both hybrid and concurrent learning for multi-layer NNs is shown to have overall closed-loop stability.

II. Problem Formulation

Consider a nonlinear DT system given in the affine form as

$$\xi_{k+1} = f(\xi_k) + g(\xi_k)u_k + d(\xi_k)w_k, \tag{1}$$

where $\xi_k \in \mathbb{R}^n$ represents the system state, $u_k \in \mathbb{R}^m$ is the control input, and $w_k \in \mathbb{R}^q$ denotes the external disturbance, which satisfies $w_k \in L_2$. The smooth functions $f(.) \in \mathbb{R}^n$ represents unknown internal dynamics, $g(.) \in \mathbb{R}^{n\times m}$ denotes a bounded known input coefficient matrix satisfying $||g(\xi_k)||_F < g_M$ on a compact set, and $d(.) \in \mathbb{R}^{n\times q}$ is a bounded disturbance function satisfying $||d(\xi_k)||_F < d_M$.

The reference trajectory can be generated by

$$r_{k+1} = h(r_k), \tag{2}$$

where $r_k \in \mathbb{R}^n$ presents the reference trajectory that is bounded and $h(r_k)$ is a C^∞ function such that $h(0) = 0$. Let the tracking error define as

$$e_k = \xi_k - r_k\,. \tag{3}$$

Next, combine (1), (2), and (3) to get

$$\begin{aligned} e_{k+1} &= f(e_k + r_k) + g(e_k + r_k)u_k \\ &+ d(e_k + r_k)w_k - h(r_k). \end{aligned} \tag{4}$$

Now, by augmenting (4) and (2), the augmented system dynamics are derived as

$$\begin{aligned} \xi^a_{k+1} &= \begin{bmatrix} f(e_k + r_k) - h(r_k) \\ h(r_k) \end{bmatrix} \\ &+ \begin{bmatrix} g(e_k + r_k) \\ 0 \end{bmatrix} u_k + \begin{bmatrix} d(e_k + r_k) \\ 0 \end{bmatrix} w_k, \end{aligned} \tag{5}$$

where $\xi^a_k = [e_k^\top, r_k^\top]^\top \in \mathbb{R}^{2n}$. For simplicity, the augmented system in (5) can be represented as $\xi^a_{k+1} = F(\xi^a_k) + G(\xi^a_k)u_k + D(\xi^a_k)w_k$ with $F(\xi^a_k) = \begin{bmatrix} f(e_k + r_k) - h(r_k) \\ h(r_k) \end{bmatrix}$, $G(\xi^a_k) = \begin{bmatrix} g(e_k + r_k) \\ 0 \end{bmatrix}$ and $D(\xi^a_k) = \begin{bmatrix} d(e_k + r_k) \\ 0 \end{bmatrix}$. It is assumed the state and desired trajectory vectors are measurable, and a control input exists for (5) that is admissible. Therefore, the primary aim of this paper is to find an optimal control policy u_k minimizing the infinite horizon discounted cost function defined as

$$J(\xi^a_k) = \sum_{j=k}^{\infty} \gamma_d^{j-k} r(\xi^a_j, u_j, w_j), \tag{6}$$

where $r(\xi^a_k, u_k, w_k) = \xi^{a\top}_k Q\xi^a_k + u_k^\top Ru_k - \gamma^2 w_k^\top Pw_k$ is the cost-to-go function. The design matrices $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$, and $P \in \mathbb{R}^{q\times q}$ are positive definite, $0 < \gamma_d < 1$ is a discount factor, and γ is the disturbance attenuation factor. By applying the ADP framework, the recursive Bellman equation is achieved as

$$J(\xi^a_k) = r(\xi^a_k, u_k, w_k) + \gamma_d J(\xi^a_{k+1}), \tag{7}$$

By invoking the system dynamics (5), we have

$$\begin{aligned} J(\xi^a_k) &= r(\xi^a_k, u_k, w_k) \\ &+ \gamma_d J\big(F(\xi^a_k) + G(\xi^a_k)u_k \\ &+ D(\xi^a_k)w_k\big). \end{aligned} \tag{8}$$

Not only does the control input stabilize the nonlinear system (1), but the cost function (6) must also be finite. Therefore, we define admissible control next.

Definition 1: A feedback control strategy $u(\xi_k)$ is referred to as admissible for system (1) with respect to (6) on a compact set $\Omega \subset \mathbb{R}^n$ if a) $u(0) = 0$; b) u_k stabilizes system (1); c) $u(\xi_k)$ makes the performance (6) finite, i.e., $J(x(0), u(\xi_k)) < \infty$.

In this paper, the control problem is a zero-sum or two-player min-max game where two strategies, i.e., $w^\star(\xi^a_k)$ and $u^\star(\xi^a_k)$ are the worst case disturbance and the optimal control input of the cost function (6), respectively. By utilizing Bellman's principle of optimality, the optimal value function can be derived as

$$\begin{aligned} J^\star(\xi^a_k) &= \min_u \max_w \Big(\sum_{j=k}^{\infty} \gamma_d^{j-k} r(\xi^a_j, u_j, w_j) \Big) \\ &= \max_w \min_u \Big(\sum_{j=k}^{\infty} \gamma_d^{j-k} r(\xi^a_j, u_j, w_j) \Big) \\ &= \xi^{a\top}_k Q\xi^a_k + u^\star(\xi^a_k)^\top Ru^\star(\xi^a_k) \\ &- \gamma^2 w^{\star\top} Pw^\star + \gamma_d J^\star\big(F(\xi^a_k) \\ &+ G(\xi^a_k)u_k + D(\xi^a_k)w_k\big). \end{aligned} \tag{9}$$

The Hamiltonian function is obtained as

$$\begin{aligned} H(\xi_a, J, u, w) &= \gamma_d J\big(F(\xi^a_k) + G(\xi^a_k)u_k \\ &+ D(\xi^a_k)w_k\big) - J(\xi^a_k) + \xi^{a\top}_k Q\xi^a_k \\ &+ u_k^\top Ru_k - \gamma^2 w_k^\top Pw_k. \end{aligned} \tag{10}$$

The HJI equation is derived as

$$\min_{u_k} \max_{w_k} H(\xi_a, J, u, w) = 0. \tag{11}$$

Assume that there exists a unique saddle point for (11) with $J(\xi_a(\infty)) = 0$. Hence, by applying the stationarity conditions

$\partial H(\xi_a, J, u, w)/\partial u(\xi_k^a) = 0$ and $\partial H(\xi_a, J, u, w)/\partial w(\xi_k^a) = 0$, we can obtain the Nash equilibrium solution given the input matrix as

$$u^\star(\xi_k^a) = -\frac{\gamma_d}{2} R^{-1} G^\top(\xi_k^a) \frac{\partial J^\star(\xi_{k+1}^a)}{\partial \xi_{k+1}^a} \tag{12}$$

$$w^\star(\xi_k^a) = \frac{\gamma_d}{2\gamma^2} P^{-1} D^\top(\xi_k^a) \frac{\partial J^\star(\xi_{k+1}^a)}{\partial \xi_{k+1}^a} \tag{13}$$

Based on the results, the following fact can be presented.

Fact 1: Applying the optimal control and the worst case disturbance inputs to the augmented system in (5) results in a bounded closed-loop system such that $\|F(\xi_k^a) + G(\xi_k^a)u^\star(\xi_k^a) + D(\xi_k^a)w^\star(\xi_k^a)\| \leqslant k^\star$ given a constant $k^\star$.

Since the optimal control policy ensures the stability of closed-loop for the nonlinear system in the presence of disturbance, and the internal dynamics, input coefficient matrix, and disturbance function are the Lipschitz functions, the aforementioned fact is reasonable [12]. Fact 1 will be utilized to demonstrate the boundedness of the closed-loop system.

Remark 1: Note that to achieve the optimal control policy $u^\star$ and the worst case disturbance $w^\star$, we need the future state variables ξ_{k+1}^a, which is unavailable. In the following section, we address this issue by using NNs.

III. HYBRID OPTIMAL TRACKING CONTROL

In this section, the hybrid optimal control is derived for the ZSG problem of the nonlinear DT system in (1). First, a two-layer critic NN is constructed to evaluate the value functional and another two-layer actor NN to approximate the optimal control strategy. A new weight-tuning law combining hybrid learning and concurrent learning is presented for the critic NN. A third NN is utilized to estimate the worst-case disturbance. Moreover, the closed-loop system stability and the value function boundedness will be guaranteed using the hybrid learning scheme embedded with a finite iteration.

The value function in (6) can be approximated as

$$J(\xi_k^a) = w_c^\top \sigma_c(v_c^\top \sigma(\xi_k^a)) + \varepsilon_{jk} \tag{14}$$

where $\sigma_c \in \mathbb{R}^{N_c}$ is the vector of the hidden layer activation function, v_c is the hidden layer, w_c is the output layer, and ε_{jk} denotes the critic NN error. Moreover, the optimal control policy in (12) and worst case w_k in (13) are approximated by

$$u(\xi_k^a) = w_a^\top \sigma_a(v_a^\top \sigma(\xi_k^a)) + \varepsilon_{uk}, \tag{15}$$

$$w(\xi_k^a) = w_w^\top \sigma_w(v_w^\top \sigma(\xi_k^a)) + \varepsilon_{wk}, \tag{16}$$

where $\sigma_a \in \mathbb{R}^{N_a}$ and $\sigma_w \in \mathbb{R}^{N_w}$ are the vector of the hidden layer activation functions, v_a and v_w are the hidden layers, w_a and w_w are the weights of the critic output layer, and ε_{uk} and ε_{wk} are the approximation errors. The following mild assumption is needed.

Assumption 1: The neural network weights and the approximation errors and their gradients are bounded over a compact set [12], i.e., $\|w_c\| \leqslant w_{cM}, \|v_c\| \leqslant v_{cM}, \|w_a\| \leqslant w_{aM}, \|v_a\| \leqslant v_{aM}, \|w_w\| \leqslant w_{wM}, \|v_w\| \leqslant v_{wM}, \|\varepsilon_{jk}\| \leqslant \varepsilon_{jM}, \|\varepsilon_{uk}\| \leqslant \varepsilon_{uM}, \|\varepsilon_{wk}\| \leqslant \varepsilon_{wM}, \|\nabla\varepsilon_{jk}\|_F \leqslant \varepsilon_{jM}^{'}, \|\nabla\varepsilon_{uk}\|_F \leqslant \varepsilon_{uM}^{'}$, and $\|\nabla\varepsilon_{wk}\|_F \leqslant \varepsilon_{wM}^{'}$.

A. Concurrent Hybrid Learning

The estimated value functional, $\hat{J}_k(\xi_k^a)$ is given by

$$\hat{J}_k(\xi_k^a) = \hat{w}_c^\top \sigma_c(\hat{v}_c^\top \sigma(\xi_k^a)) \tag{17}$$

where $\hat{v}_c^\top$ and $\hat{w}_c^\top$ present the estimated hidden and output layer critic weights. Substituting the estimated value functional (17) into (7) results in TDE given by

$$\begin{aligned} \mathcal{E}_k = {} & r(\xi_{k-1}^a, u(\xi_{k-1}^a), w(\xi_{k-1}^a)) \\ & + \hat{w}_c^\top \Delta\sigma_c(\xi_{k-1}^a), \end{aligned} \tag{18}$$

where $\mathcal{E}_k \in \mathbb{R}$ is the TDE, and $\Delta\sigma_c(\xi_{k-1}^a) = \gamma_d \sigma_c(\hat{v}_c^\top \sigma(\xi_k^a)) - \sigma_c(\hat{v}_c^\top \sigma(\xi_{k-1}^a))$. Not only does the TDE in (18) relies on the tracking error, but also it depends on the desired trajectory unlike in the regulation problem.

Employing (14) in (7) renders $r(\xi_{k-1}^a, u(\xi_{k-1}^a), w(\xi_{k-1}^a)) = w_c^\top \sigma_c(v_c^\top \xi_{k-1}^a) - \gamma_d w_c^\top \sigma_c(v_c^\top \xi_k^a) - \Delta\varepsilon_{jk}$ where $\Delta\varepsilon_{jk} = \gamma_d \varepsilon_{jk} - \varepsilon_{jk-1}$. Substituting $r(\xi_{k-1}^a, u(\xi_{k-1}^a), w(\xi_{k-1}^a))$ into (18) yields

$$\begin{aligned} \mathcal{E}_k = {} & w_c^\top \sigma_c(v_c^\top \sigma(\xi_{k-1}^a)) \\ & - \gamma_d w_c^\top \sigma_c(v_c^\top \sigma(\xi_k^a)) - \Delta\varepsilon_{jk} \\ & + \gamma_d \hat{w}_c^\top \sigma_c(\hat{v}_c^\top \sigma(\xi_k^a)) \\ & - \hat{w}_c^\top \sigma_c(\hat{v}_c^\top \sigma(\xi_{k-1}^a)). \end{aligned} \tag{19}$$

Add and subtract $\gamma_d w_c^\top \sigma_c(\hat{v}_c^\top \sigma(\xi_k^a))$ and $w_c^\top \sigma_c(\hat{v}_c^\top \sigma(\xi_{k-1}^a))$ to (19) to get

$$\begin{aligned} \mathcal{E}_k = {} & -\tilde{w}_c^\top \Delta\sigma_c(\xi_{k-1}^a) \\ & + w_c^\top \left[\gamma_d \tilde{\sigma}_{ck} + \tilde{\sigma}_{c(k-1)}\right] - \Delta\varepsilon_{jk}, \end{aligned} \tag{20}$$

where $\tilde{w}_c = w_c - \hat{w}_c$ is the weight estimation error of the critic output layer and $\tilde{\sigma}_{ck} = \sigma_c(\hat{v}_{ck}^\top \xi_k^a) - \sigma_c(v_c^\top \xi_k^a)$. Eq. (20) can be expressed as

$$\mathcal{E}_k = -\tilde{w}_c^\top \Delta\sigma_c(\xi_{k-1}^a) + \varepsilon_B, \tag{21}$$

where $\varepsilon_B = w_c^\top \Pi_k - \Delta\varepsilon_{jk}$ with $\Pi_k = \tilde{\sigma}_c + \tilde{\sigma}_{c(k-1)}$. Due to the fact that $\Delta\hat{\sigma}_c(\xi_{k-1}^a) \leqslant \sigma_M$, $\|\Delta\varepsilon_{jk}\| \leqslant \varepsilon_{JM}$, $w_c \leqslant w_{cM}$, and $\|\Pi_k\| \leqslant \Pi_M$, the approximation error is bounded on the compact set, i.e., $\|\varepsilon_B\| < \varepsilon_{Bmax}$.

Since a PE condition guarantees the critic NN weight boundedness, the current transition samples are saved in a stack. We employ them to the critic neural network update law as a concurrent learning term [10]. The proposed update law minimizes the summation of the TDE at k and the TDE corresponding with the recorded time k_j in the experience replay buffer.

To store samples in the experience buffer in real-time, the values of $\hat{\sigma}_c$ and $r(\xi_k^a, u_k, w_k)$ are evaluated at the recorded time k_j as $\hat{\sigma}_c(k_j)$ and $r(\xi_{k_j}^a, u_{k_j}, w_{k_j})$. Thus, we define

$$\Delta\sigma_{cj} = \gamma_d \sigma_{ck_j} - \sigma_{c(k_j-1)} \tag{22}$$

$$r_j = r(\xi_{k_j}^a, u_{k_j}, w_{k_j}). \tag{23}$$

Then, the TDE error associated with the recorded time k_j is defined as

$$\mathcal{E}_{k_j} = r_j + \hat{w}_c^\top \Delta\sigma_{cj} \tag{24}$$

Based on the gradient-descent method, the tuning law for the critic two-layer NN weights is proposed as

$$\begin{aligned}
\hat{w}_{c(k+1)} &= \hat{w}_c \\
&- \frac{\alpha_J \Delta\sigma_c(\hat{v}_c^\top \sigma(\xi_k^a))\mathcal{E}_k}{\Delta\sigma_C^\top(\hat{v}_c^\top \sigma(\xi_k^a))\Delta\sigma_c(\hat{v}_c^\top \sigma(\xi_k^a)) + 1} \\
&- \alpha_J \sum_{j=1}^{l} \frac{\Delta\sigma_{cj}}{\Delta\sigma_{cj}^\top \Delta\sigma_{cj} + 1} \mathcal{E}_(k_j), \\
\hat{v}_{c(k+1)} &= \hat{v}_c - \sigma(\xi_k^a)(\hat{v}_c^\top \xi_k^a \\
&+ B_1 k_v \mathcal{E}_k)^\top - \sum_{j=1}^{l} \sigma(\xi_{k_j}^a)(B_1 k_v \Delta\sigma_{cj} \mathcal{E}_{k_j})^\top
\end{aligned} \tag{25}$$

where B_1 and k_v are constant matrices with proper dimensions, and η_J is a fixed rate of learning. The subscript j represents the $j-th$ instance in the history stack of stored sample data. l is the number of the stored samples. Verifying the PE condition becomes checking the following condition in concurrent learning.

Condition 1: The experience buffer is defined as

$$Z = [\Delta\bar{\sigma}_1, \ldots, \Delta\bar{\sigma}_l] \tag{26}$$

where $\Delta\bar{\sigma}_j = \Delta\sigma_{cj}/(\Delta\sigma_{cj}^\top \Delta\sigma_{cj} + 1)$. Thus, the number of linearly independent recorded data in Z is equal to the number of critic NN neurons, i,e., $rank(Z) = N_c$. The number of samples in the experience replay buffer is a fixed value $l > N_c$.

Next, an innovative method for adjusting the weights of sample intervals, i.e., $[k, k+1)$ is provided as

$$\begin{aligned}
\hat{w}_{c(k+1)}^{i+1} &= \hat{w}_c^i \\
&- \frac{\alpha_J \Delta\sigma_c(\hat{v}_c^{i\top} \sigma(\xi_k^a))\mathcal{E}_k}{\Delta\sigma_C^\top(\hat{v}_c^{i\top} \sigma(\xi_k^a))\Delta\sigma_c(\hat{v}_c^{i\top} \sigma(\xi_k^a)) + 1} \\
&- \alpha_J \sum_{j=1}^{l} \frac{\Delta\sigma_{cj}}{\Delta\sigma_{cj}^\top \Delta\sigma_{cj} + 1} \mathcal{E}_{k_j}, \\
\hat{v}_{c(k+1)}^{i+1} &= \hat{v}_c^i \\
&- \sigma(\xi_k^a)\big(\hat{v}_c^{i\top} \sigma(\xi_k^a) + B_1 k_v \mathcal{E}_k\big)^\top \\
&- \sum_{j=1}^{l} \sigma(\xi_{k_j}^a)(B_1 k_v \mathcal{E}_{k_j})^\top \;\; i = 1, \ldots, \mathcal{L},
\end{aligned} \tag{27}$$

where i is the finite iteration number, and $\mathcal{L}$ is the total number of iterations during sampling intervals.

Remark 2: The hybrid learning technique enhances the convergence rate of the learning scheme. The estimated control policy reaches optimal value quickly when the value function converges faster.

Now, by invoking (12), (13) and (17), the estimated control input and the estimated worst case disturbance are obtained as

$$\hat{u}_k = -\frac{\gamma_d}{2} R^{-1} G^\top \frac{\partial\sigma_c\big(\hat{v}_c^\top \sigma(\xi_{k+1}^a)\big)^\top}{\partial\xi_{k+1}^a} \hat{w}_c \;, \tag{28}$$

$$\hat{w}_k = \frac{\gamma_d}{2\gamma^2} P^{-1} D^\top \frac{\partial\sigma_c\big(\hat{v}_c^\top \sigma(\xi_{k+1}^a)\big)^\top}{\partial\xi_{k+1}^a} \hat{w}_c. \tag{29}$$

In the next subsection, the actor NN weight tuning law is investigated, and then, the boundedness of the overall closed-loop system is guaranteed through Lyapunov analysis.

B. Approximate Optimal Control Policy and Disturbance

To estimate the optimal control input and the worst disturbance obtained in (12) and (13), two feedforward NNs of two-layer each are utilized for the actor networks as

$$\hat{u}(\xi_k^a) = \hat{w}_a^\top \sigma_a(\hat{v}_a^\top \sigma(\xi_k^a)) \tag{30}$$

$$\hat{w}(\xi_k^a) = \hat{w}_w^\top \sigma_w(\hat{v}_w^\top \sigma(\xi_k^a)) \tag{31}$$

where $\hat{w}_a$, $\hat{w}_w$, $\hat{v}_a$, and $\hat{v}_w$ are the weights and σ_a, σ_w and σ are the activation functions of the actor NNs. By invoking (30) and (28), the control input error is expressed as

$$\begin{aligned}
\tilde{u}_k &= \hat{w}_a^\top \sigma_a\left(\hat{v}_a^\top \sigma(\xi_k^a)\right) \\
&+ \frac{\gamma_d}{2} R^{-1} G\left(\xi_k^a\right)^\top \frac{\partial\sigma_c\big(\hat{v}_c^\top \sigma(\xi_{k+1}^a)\big)^\top}{\partial\xi_{k+1}^a} \hat{w}_c.
\end{aligned} \tag{32}$$

Similarly, the worst case error can be written as

$$\begin{aligned}
\tilde{w} &= \hat{w}_w^\top \sigma_w(\hat{v}_w^\top \sigma(\xi_k^a)) \\
&- \frac{\gamma_d}{2\gamma^2} P^{-1} D^\top(\xi_k^a) \frac{\partial\sigma_c\big(\hat{v}_c^\top \sigma(\xi_{k+1}^a)\big)^\top}{\partial\xi_{k+1}^a} \hat{w}_c.
\end{aligned} \tag{33}$$

Substituting (14) and (15) into (12) gives

$$\begin{aligned}
&w_a^\top \sigma_a\left(v_a^\top \sigma(\xi_k^a)\right) + \varepsilon_{uk} = \\
&- \frac{\gamma_d}{2} R^{-1} G\left(\xi_k^a\right)^\top \frac{\partial\sigma_c\big(v_c^\top \sigma(\xi_{k+1}^a)\big)^\top}{\partial\xi_{k+1}^a} w_c \\
&- \frac{\gamma_d}{2} R^{-1} G\left(\xi_k^a\right)^\top \frac{\partial\varepsilon_{jk+1}}{\partial\xi_{k+1}^a}.
\end{aligned} \tag{34}$$

Employing (34) in (32) renders

$$\begin{aligned}
\tilde{u}_k &= \hat{w}_a^\top \sigma_a\left(\hat{v}_a^\top \sigma(\xi_k^a)\right) \\
&+ \frac{\gamma_d}{2} R^{-1} G^\top\left(\xi_k^a\right) \frac{\partial\sigma_c\big(\hat{v}_c^\top \sigma(\xi_{k+1}^a)\big)^\top}{\partial\xi_{k+1}^a} \hat{w}_c \\
&- w_a^\top \sigma_a\left(v_a^\top \sigma(\xi_k^a)\right) + \varepsilon_{uk} \\
&- \frac{\gamma_d}{2} R^{-1} G^\top\left(\xi_k^a\right) \frac{\partial\sigma_c\big(v_c^\top \sigma(\xi_{k+1}^a)\big)^\top}{\partial\xi_{k+1}^a} w_c \\
&- \frac{\gamma_d}{2} R^{-1} G^\top\left(\xi_k^a\right) \frac{\partial\varepsilon_{jk+1}}{\partial\xi_{k+1}^a}.
\end{aligned} \tag{35}$$

Adding and subtracting $w_a^\top \sigma_a\left(\hat{v}_a^\top \sigma(\xi_k^a)\right)$ and $\frac{\gamma_d}{2} R^{-1} G^\top(\xi_k^a) \frac{\partial \sigma_c\left(\hat{v}_c^\top \sigma(\xi_{k+1}^a)\right)^\top}{\partial \xi_{k+1}^a} w_c$ in (35) and after performing some manipulations, we have

$$\begin{aligned}\tilde{u}_k &= -\tilde{w}_a^\top \sigma_a - w_a^\top \tilde{\sigma}_a \\ &- \frac{\gamma_d}{2} R^{-1} G^\top(\xi_k^a) \frac{\partial \sigma_c\left(\hat{v}_c^\top \xi_{k+1}^a\right)^\top}{\partial \xi_k^a} \tilde{w}_c \\ &- \frac{\gamma_d}{2} R^{-1} G^\top(\xi_k^a) \frac{\partial \tilde{\sigma}_c(k+1)}{\partial \xi_{k+1}^a} w_c - \tilde{\varepsilon}_{uk},\end{aligned} \tag{36}$$

where $\tilde{\sigma}_a = \sigma_a\left(v_a^\top \sigma(\xi_k^a)\right) - \sigma_a\left(\hat{v}_a^\top \sigma(\xi_k^a)\right)$, $\sigma_a = \sigma_a\left(\hat{v}_a^\top \sigma(\xi_k^a)\right)$ and $\frac{\gamma_d}{2} R^{-1} G^\top(\xi_k^a) \frac{\partial \varepsilon_{jk+1}}{\partial \xi_{k+1}^a}$. The actor weight estimation error is $\tilde{w}_a = w_a - \hat{w}_a$. Since $\tilde{u}(\xi_k^a)$ is measurable, the actor weight updating rules are achieved as

$$\begin{aligned}\hat{w}_{a(k+1)} &= \hat{w}_a \\ &- \frac{\alpha_u \sigma_a\left(\hat{v}_a^\top \sigma(\xi_k^a)\right) \tilde{u}^\top}{\left(\sigma_a{}^\top\left(\hat{v}_a^\top \sigma(\xi_k^a)\right) \sigma_a\left(\hat{v}_a^\top \sigma(\xi_k^a)\right) + 1\right)} \\ \hat{v}_{a(k+1)} &= \hat{v}_a \\ &+ \sigma(\xi_k^a)\left(\hat{v}_a^\top \sigma(\xi_k^a) + B_2 k_v \tilde{u}_k\right)^\top,\end{aligned} \tag{37}$$

where $0 < \eta_u < 1$, B_2, and k_v are a positive learning rate parameter and matrices of suitable dimensions, respectively. By using (37), the actor weight estimation error dynamics can be rewritten as

$$\begin{aligned}\tilde{w}_{a(k+1)} &= \tilde{w}_a \\ &- \frac{\alpha_u \sigma_a\left(\hat{v}_a^\top \sigma(\xi_k^a)\right) \tilde{u}^\top}{\left(\sigma_a{}^\top\left(\hat{v}_a^\top \sigma(\xi_k^a)\right) \sigma_a\left(\hat{v}_a^\top \sigma(\xi_k^a)\right) + 1\right)}, \\ \tilde{v}_{a(k+1)} &= \tilde{v}_a \\ &+ \sigma(\xi_k^a)\left(\hat{v}_a^\top \sigma(\xi_k^a) + B_2 k_v \tilde{u}_k\right)^\top.\end{aligned} \tag{38}$$

Similar to the actor update laws, the weight tuning laws associated with the worst case disturbance are selected as

$$\begin{aligned}\hat{w}_{w(k+1)} &= \hat{w}_w \\ &- \frac{\alpha_w \sigma_w\left(\hat{v}_w^\top \sigma(\xi_k^a)\right) \tilde{w}^\top}{\left(\sigma_w{}^\top\left(\hat{v}_w^\top \sigma(\xi_k^a)\right) \sigma_w\left(\hat{v}_w^\top \sigma(\xi_k^a)\right) + 1\right)} \\ \hat{v}_{w(k+1)} &= \hat{v}_w \\ &+ \sigma(\xi_k^a)\left(\hat{v}_{wk}^\top \sigma(\xi_k^a) + B_3 k_{v3} \tilde{w}_k\right)^\top,\end{aligned} \tag{39}$$

where $0 < \eta_w < 1$, B_3, and k_{v3} are a positive learning rate parameter and matrices of suitable dimensions, respectively. By using (39), the actor NN weight estimation error dynamics associated with the worst case disturbance can be rewritten as

$$\begin{aligned}\tilde{w}_w(k+1) &= \tilde{w}_w \\ &- \frac{\alpha_w \sigma_w\left(\hat{v}_w^\top \sigma(\xi_k^a)\right) \tilde{w}^\top}{\left(\sigma_w{}^\top\left(\hat{v}_w^\top \sigma(\xi_k^a)\right) \sigma_w\left(\hat{v}_w^\top \sigma(\xi_k^a)\right) + 1\right)}, \\ \tilde{v}_w(k+1) &= \tilde{v}_w \\ &+ \sigma(\xi_k^a)\left(\hat{v}_w^\top \sigma(\xi_k^a) + B_3 k_{v3} \tilde{u}_k\right)^\top.\end{aligned} \tag{40}$$

Note unlike the critic NN, the actor NN weights are only tuned once at sampling instants. Next, the following theorem shows the boundedness of the overall closed-loop system.

Theorem 1: Consider the augmented system in (5) under Assumption 1 with Condition 1. Let the critic NN weights update law be adjusted by (25), and the actor neural network weights tuned by (37) and (39). Then, there exist $\eta_u > 0$, $\eta_w > 0$ and $\eta_J > 0$ such that the augmented state ξ_k^a, the tracking error e_k, the hidden and output weight estimation error of critic NNs $\tilde{w}_c$ and $\tilde{v}_c$ and the weight estimation errors of the actor $\tilde{v}_a$, $\tilde{w}_a$, $\tilde{v}_w$, and $\tilde{w}_w$, are all UUB.

Proof: Due to space constraints, the proof is omitted. ■

IV. SIMULATION RESULTS

In this section, a two-link robot is employed to show the effectiveness of the proposed approach. We consider a two-link robot manipulator defined by

$$\begin{aligned}X_1((k+1)T) &= X_1(kT) + TX_2(kT) \\ X_2((k+1)T) &= X_2(kT) + T(F(X_1, X_2) + M(X_1)^{-1}U),\end{aligned} \tag{41}$$

where $X_1 = [x_1, x_2]^T$ denotes the joint position, $X_2 = [x_3, x_4]^T$ presents the joint velocities and $U = [u_1, u_2]^T$ are the torque inputs for the joints. The time step is $T = 0.01s$. The nonlinear function is expressed as $F(X_1, X_2) = -[M(X_1)]^{-1} N(X_1, X_2)$ with

$$M(X_1(kT)) = \begin{bmatrix} 3 + 2cos(x_2(kT)) & 1 + cos(x_2(kT)) \\ 1 + cos(x_2(kT)) & 1 \end{bmatrix} \tag{42}$$

$$N(X_1(kT), X_2(kT)) = \begin{bmatrix} -(2x_3x_4 + x_4^2)sin(x_2) + 19.6cos(x_1) + 9.8cos(x_1 + x_2) \\ x_1^2 sin(x_2) + 9.8cos(x_1 + x_2) \end{bmatrix} \tag{43}$$

The disturbance gain is taken as $d(\xi_k) = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}^\top$, and the disturbance $w = \frac{1}{20} e^{-kT}$ is applied to the robot at $k = 200$. We define the reference trajectory as

$$r_k = exp(-0.25k) \begin{bmatrix} sin(k) \\ cos(k) \\ cos(k) - \frac{1}{4} sin(k) \\ -sin(k) - \frac{1}{4} cos(k) \end{bmatrix}. \tag{44}$$

The augmented quadratic function value is selected as $r(\xi_k^a, u_k, w_k) = \xi_k^{a\top} Q \xi_k^a + u_k^\top R u_k - \gamma^2 w_k^\top P w_k$ with $\bar{Q} = \begin{bmatrix} Q & 0_{2\times 2}; 0_{2\times 2} & 0_{2\times 2} \end{bmatrix}$, with the selected value of $Q = I_4$, $R = 0.01 I_2$, $P = I_2$, and $\gamma = 100$. The initial values for the state set as $x_0 = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^T$, and the initial admissible control input is set to $u_0 = -\begin{bmatrix} 100 & 0 & 20 & 0 \\ 0 & 100 & 0 & 20 \end{bmatrix} e_0$.

To verify the effectiveness of the concurrent hybrid learning technique, we select a 2-layer NN with 36, 11, and 1 neurons in the input, hidden and output layers for the critic neural network, respectively. The hidden and output layers are selected tangent hyperbolic and polynomial activation functions, respectively. The hybrid factor is set as $\mathcal{L} = 10$. The design parameters are taken as $\gamma_d = 0.5$, $\eta_u = 0.02$, and $\eta_J = 0.01$. The value of B_i is chosen as constant vector of 0.01 with $B_{ci} \in \mathbb{R}^{36}$ for critic NN with 36 hidden layer neurons and $B_{ai} \in \mathbb{R}^{20}$ for actor NN with 20 hidden layer neurons. The critic and

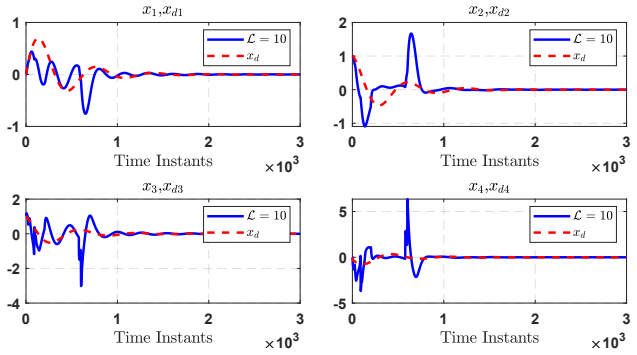

Fig. 1: The system state and reference trajectory, when the concurrent hybrid learning update law (25) is utilized.

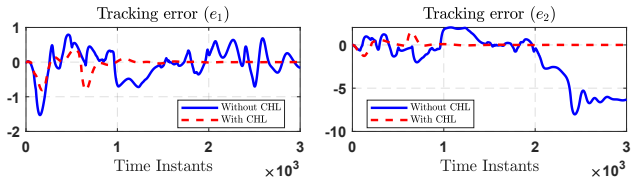

Fig. 2: Performance of the concurrent hybrid learning approach.

actor NN weight initialization is chosen randomly selected in the interval $[0, 1]$ and $[-0.1, 0.1]$, respectively.

In Fig. 1, the state and reference trajectories are depicted. The tracking errors also are demonstrated in Fig. 2, which shows the convergence of the tracking error. Indeed, the proposed method helps generate optimal control input and enables faster convergence of tracking error near zero and neural network weights without the PE signal. In Fig.3, the estimated control actions are depicted. It is worth noting that the hybrid control policy implemented in the critic NN does not necessitate the PE condition, while external noise is applied to the actor. It is also noticed that the TDE and control policy errors converge close to zero when the tracking error approaches near zero. In Fig. 4, the simulation results are shown for two distinct scenarios where in the first scenario, the concurrent hybrid learning term in (25) and (27) is not taken into account. In contrast, the second scenario considers the results with the concurrent hybrid term. In Fig. 4, the norm of the critic neural network weights is illustrated.

V. CONCLUSION

This paper presented a concurrent learning-based optimal tracking control to solve the ZSG for partially uncertain nonlinear DT systems. Using three two-layer NNs, the optimal control policy, the worst-case disturbance, and the value function were directly obtained. The novel hybrid technique to update the critic weights at the sampling instants as well as within the sampling instants in a finite iterative fashion appears to enhance the controller performance. As can be seen, the proposed hybrid tuning approach promoted accelerating convergence. Moreover, the concurrent learning method was devised to relax the need for the PE condition. Furthermore, based on the Lyapunov stability theorem, the tracking and weight estimation errors of all NNs are UUB. Finally, the simulation outcomes have confirmed the validity of the

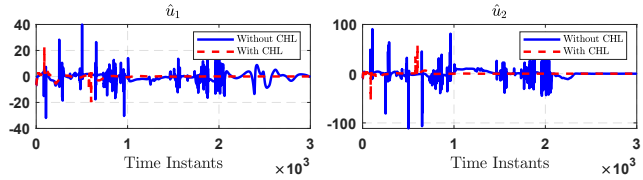

Fig. 3: Estimated control input without (without CHL) and with concurrent learning term (with CHL).

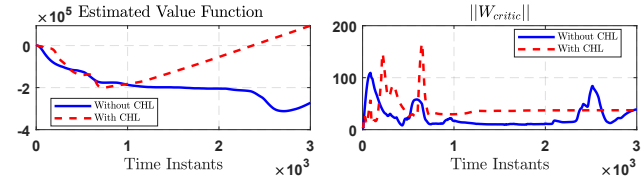

Fig. 4: Total cost and the norm of the critic NN weight comparison without and with concurrent learning term.

proposed concurrent hybrid learning-based optimal tracking control for the ZSG problems of nonlinear DT systems.

REFERENCES

[1] R. Song, J. Li, and F. L. Lewis, "Robust optimal control for disturbed nonlinear zero-sum differential games based on single neural network and least squares," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4009–4019, 2019.

[2] S. Mehraeen, T. Dierks, S. Jagannathan, and M. L. Crow, "Zero-sum two-player game theoretic formulation of affine nonlinear discrete-time systems using neural networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1641–1655, 2012.

[3] X. Zhang, H. Zhang, Y. Luo, and M. Dong, "Iteration algorithm for solving the optimal strategies of a class of nonaffine nonlinear quadratic zero-sum games," in *2010 Chinese Control and Decision Conference*, pp. 1359–1364, IEEE, 2010.

[4] H.-N. Wu and B. Luo, "Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear H_∞ control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 12, pp. 1884–1895, 2012.

[5] Y. Zhang, B. Zhao, D. Liu, and S. Zhang, "Event-triggered control of discrete-time zero-sum games via deterministic policy gradient adaptive dynamic programming," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 8, pp. 4823–4835, 2021.

[6] X. Zhong, H. He, D. Wang, and Z. Ni, "Model-free adaptive control for unknown nonlinear zero-sum differential game," *IEEE transactions on cybernetics*, vol. 48, no. 5, pp. 1633–1646, 2017.

[7] H. Jiang and H. Zhang, "Iterative adp learning algorithms for discrete-time multi-player games," *Artificial Intelligence Review*, vol. 50, no. 1, pp. 75–91, 2018.

[8] I. Ganie and S. Jagannathan, "Adaptive control of robotic manipulators using deep neural networks," *IFAC-PapersOnLine*, vol. 55, no. 15, pp. 148–153, 2022.

[9] S. Adam, L. Busoniu, and R. Babuska, "Experience replay for real-time reinforcement learning control," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 2, pp. 201–212, 2011.

[10] 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, 2014.

[11] 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 Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

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