01 Jan 2022

Optimal Adaptive Output Regulation of Uncertain Nonlinear Discrete-Time Systems using Lifelong Concurrent Learning

R. Moghadam

B. Farzanegan

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

P. Natarajan

Recommended Citation

R. Moghadam et al., "Optimal Adaptive Output Regulation of Uncertain Nonlinear Discrete-Time Systems using Lifelong Concurrent Learning," *Proceedings of the IEEE Conference on Decision and Control*, pp. 2005 - 2010, Institute of Electrical and Electronics Engineers, Jan 2022.
The definitive version is available at https://doi.org/10.1109/CDC51059.2022.9993219

2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Optimal Adaptive Output Regulation of Uncertain Nonlinear Discrete-time Systems using Lifelong Concurrent Learning

R. Moghadam, *Member, IEEE*, B. Farzanegan, S. Jagannathan, *Fellow IEEE*, and P. Natarajan

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9993219

***Abstract*— This paper addresses neural network (NN) based optimal adaptive regulation of uncertain nonlinear discrete-time systems in affine form using output feedback via lifelong concurrent learning. First, an adaptive NN observer is introduced to estimate both the state vector and control coefficient matrix, and its NN weights are adjusted using both output error and concurrent learning term to relax the persistency excitation (PE) condition. Next, by utilizing an actor-critic framework for estimating the value functional and control policy, the critic network weights are tuned via both temporal different error and concurrent learning schemes through a replay buffer. The actor NN weights are tuned using control policy errors. To attain lifelong learning for performing effectively during multiple tasks, an elastic weight consolidation term is added to the critic NN weight tuning law. The state estimation, regulation, and the weight estimation errors of the observer, actor and critic NNs are demonstrated to be bounded when performing tasks by using Lyapunov analysis. Simulation results are carried out to verify the effectiveness of the proposed approach on a Vander Pol Oscillator. Finally, extension to optimal tracking is given briefly.**

I. INTRODUCTION

One of the major research thrusts in the control community has been optimal control of dynamical systems. The optimal control policy can be derived by using the Hamiltonian-Jacobi-Bellman (HJB) equation for an affine nonlinear system. Nevertheless, a closed-form solution [1] does not exist for the HJB equation, even for nonlinear systems with known dynamics. Therefore, iterative methods using adaptive dynamic programming (ADP) have been introduced to solve both the HJB equation and the optimal control policy.

The approximate dynamic programming (ADP) is a powerful framework to find the optimal control policy in a forward-in-time manner for nonlinear systems with unknown dynamics [2], [3]. Several iterative optimal adaptive control (OAC) techniques using ADP were reported mainly for regulation [4], [5]. The convergence of iterative methods to finding a solution to the HJB equation for nonlinear systems is shown when the number of iterations tends to infinity [1] which appears to be a bottleneck in the real-time implementation. Despite the drawback of iterative methods, traditional ADP framework using neural networks (NN) has been employed to generate an approximate optimal regulation policy online for uncertain nonlinear systems.However, a persistent excitation condition (PE) is needed for the convergence of critic NN weights and control policy.

In concurrent learning method for continuous-time system [6], [7], current and historical data, which is stored in a buffer, simultaneously are utilized to fulfill the PE requirement [8]. In [9], an OAC technique with concurrent learning were introduced for discrete systems (DT) without the proof of convergence and stability. In all regulation efforts [4], [5], single layer NN with the appropriate selection of basis functions and state measurements has been utilized. In contrast, an approximate optimal adaptive regulator (OAR) for unknown nonlinear DT systems using traditional adaptive output feedback control approach is not investigated.

Thus, the OAR for uncertain nonlinear DT systems in affine form is presented. First, an adaptive single-layer NN observer to estimate the state vector is introduced whose weights are adjusted using both measured output error and concurrent learning to relax PE. Next, the optimal value functional is expressed in relation to a cost-to-go function of the estimate state vector. By using the recursive Bellman equation, defined in terms of the value functional, and through stationarity condition, the optimal control input is derived.

The proposed OAR scheme relaxes the need for internal dynamics and control coefficient matrix. The NN weights of the actor NN are updated through control input errors. The weights of the critic NN are tuned through temporal difference error (TDE) which is generated using estimated state vector of the observer. Next, in order to achieve lifelong learning under multiple tasks, the elastic weight consolidation (EWC) term is added to the critic NN weight tuning law. Subsequently, the overall closed-loop stability of the NN-based OAR scheme is demonstrated by using lifelong concurrent learning. In other words, three single-layer NNs–observer, critic and actor are employed for the control design.

Notation. The natural and real numbers are $\mathbb{R}$ and $\mathbb{N}$, respectively. I is an identity matrix with proper dimensions and $\|.\|$ denotes the Euclidean norm. The transpose of the matrix A, is given by A^T and $rank(A)$ denotes the rank of the matrix A. The $\lambda_{min}(A)$ and $tr(A)$ denote the minimum eigenvalue and the trace of the matrix A, respectively.

II. ADAPTIVE DISCRETE-TIME ESTIMATION USING CONCURRENT LEARNING

In this section, the adaptive parameter estimation problem of DT systems is provided using concurrent learning. Consider the

The project or effort undertaken is sponsored in part by the Department of the Navy, Office of Naval Research Grant N00014-21-1-2232, Department of Army Cooperative Research Agreement W911NF2120260, Intelligent Systems Center at Missouri University of Science and Technology, Rolla, and Fulbright Fellowship to the last author.

R. Moghadam is with the Dept. of Electrical and Electronic Engineering, California State University-Sacramento, Sacramento, CA, USA.

Behzad Farzanegan and S. Jagannathan are with the Dept. of Elec. and Comp. Engg, Missouri University of S&T, Rolla, MO, USA.

P. Natarajan is with the Dept. of Instr. Engg, MIT, Anna Univ, Chennai, India. Email: (moghadam@csus.edu, b.farzanegan@mst.edu, sarangap@mst.edu and npappa@annauniv.edu).

general form of a DT system as [10]

$$y(k+1) = \theta^T \phi(k)\,, \tag{1}$$

where $y(k) \in \mathbb{R}^n$ is the system output, $u(k) \in \mathbb{R}^n$ is the control input, and $\theta \in \mathbb{R}^{n\times m}$ and $\phi(k) = \phi(y(k)) \in \mathbb{R}^m$ denote the system parameter matrix and regressor, respectively. Let the estimated output be defined as

$$\hat{y}(k+1) = \hat{\theta}(k)^T \phi(k)\,, \tag{2}$$

where $\hat{\theta}(k)$ is the estimation parameter matrix and $e(k) = y(k) - \hat{y}(k)$ is the output estimation error. Using (1) and (2), the dynamics of the output estimation error becomes

$$e(k+1) = \tilde{\theta}(k)^T \phi(k)\,, \tag{3}$$

with $\tilde{\theta}(k) = \theta - \hat{\theta}(k)$ as the parameter estimation error matrix. The traditional update law for parameter estimation is given as [10]

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \alpha\phi(k)e^T(k+1)\,, \tag{4}$$

which makes the parameter estimation error matrix, $\tilde{\theta}$, and the output estimation error, $e(k)$, are UUB if the PE condition is satisfied. The PE condition provided in [10] is listed as follows.

Definition 1. An input sequence $x(k)$ is said to be persistently exciting, if there are $\lambda > 0$ and an integer $k_1 \geq 1$ such that

$$\lambda_{min}\left[\sum_{k=k_0}^{k_1+k-1} x(k)x^T(k)\right] > \lambda \quad ,\forall k_0 \geq 0 \tag{5}$$

Note that the parameter tuning law (4) employs only instantaneous information available for adaptation. However, if the tuning law uses recorded data concurrently with current data for adaptation, and if the recorded data were sufficiently rich, then intuitively it should be possible to guarantee parameter convergence without requiring persistently exciting $\phi(k)$. A concurrent learning algorithm is presented for adaptive parameter identification that builds on this intuitive concept. Let $j \in \{1, 2, ...p\}$ denote the index of a stored data point y_j , let $\phi(y(k_j))$ or $\phi(k_j)$ denote the regressor vector evaluated at point y_s, let $\epsilon_j = \tilde{\theta}(k)^T\phi(k_j)$, then, the concurrent learning scheme is given as

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \alpha\phi(k)e^T(k+1) + \alpha\sum_{j=1}^{p}\phi\left(k_j\right)\epsilon_j^T, \tag{6}$$

Using (6) and (3), the parameter estimation error dynamics can be found as

$$\tilde{\theta}(k+1) = (I - \alpha\phi(k)\phi(k)^T)\tilde{\theta}(k) - \alpha\sum_{j=1}^{p}\phi(k_j)\phi(k_j)^T\tilde{\theta}(k) \tag{7}$$

which can be written as

$$\tilde{\theta}(k+1) = (I - \alpha\phi(k)\phi(k)^T - \alpha\sum_{j=1}^{p}\phi(k_j)\phi(k_j)^T)\tilde{\theta}(k)\,, \tag{8}$$

Note that the extra term from concurrent learning for relaxing the PE is proposed for continuous-time systems and it is similar to the one from [10]. The main difference being the construction of this term using linear independence condition of the stored data which characterizes its richness [6]. This is now given for DT systems.

Condition 1. [6] The recorded data has as many linearly independent elements as the dimension of $\phi(k)$. That is, if $Z = [\phi\left(y_1\right), \ldots . \phi\left(y_p\right)]$, then $\mathrm{rank}(Z) = m$.

This condition requires that the stored data contain sufficiently different elements to form a basis for the linearly parameterized uncertainty. This condition differs from the condition on PE $\phi(k)$ in the following ways: 1) This condition applies only to recorded data which is a subset of all past data, whereas PE applies to how $\phi(k)$ should behave in the future; 2) This condition is conducive to online monitoring since the rank of a matrix can be determined online; 3) It is always possible to record data such that condition 1 is met when the system states are exciting over a finite time interval; 4) It is also possible to meet this condition by selecting and recording data during a normal course of operation over a long period without requiring PE. Using Lyapunov stability theorem, it can be shown that condition 1 is sufficient to guarantee global exponential parameter convergence for concurrent learning update law.

To this end, consider the Lyapunov candidate function as $L(\tilde{\theta}) = trace\{\tilde{\theta}(k)^T\tilde{\theta}(k)\}$ with the first difference as $\Delta L(\tilde{\theta}) = trace\{\tilde{\theta}(k+1)^T\tilde{\theta}(k+1) - \tilde{\theta}(k)^T\tilde{\theta}(k)\}$. Defining $\Gamma = I - \alpha\left(\phi(k)\phi(k)^T + \sum_{j=1}^{p}\phi(k_j)\phi(k_j)\right)^T$ and substituting (8) gives$\Delta L(\tilde{\theta}) = trace\{\tilde{\theta}(k)^T\Gamma^T\Gamma\tilde{\theta}(k) - \tilde{\theta}(k)^T\tilde{\theta}(k)\}$. After some manipulations, one has $\Delta L(\tilde{\theta}) \leq -trace\{\tilde{\theta}(k)^T\left(I - \|\Gamma\|^2\right)\tilde{\theta}(k)\}$ which can be written as $\Delta L(\tilde{\theta}) \leq -\lambda_{min}\left(I - \|\Gamma\|^2\right)L(\tilde{\theta})$. Since according to Condition 1, $\alpha\sum_{j=1}^{p}\phi(k_j)\phi(k_j) > 0$, then, $(I - \|\Gamma\|^2)$ is a positive definite matrix as long as $0 \leq \alpha \leq 2$ for all k. Now, from Theorem 5.7 in [11], $\tilde{\theta} = 0$ becomes exponentially stable which from (8) results in the exponential stability of the parameter estimation error. This concurrent learning term for DT systems will be included as part of the regulator.

III. PROBLEM FORMULATION

The OAR is formulated for an uncertain affine nonlinear DT system given by

$$\begin{cases} x(k+1) = f(x(k)) + g(x(k))u(k) \\ y = Cx(k) \end{cases}, \tag{9}$$

where $x(k) \in \mathbb{R}^n$ is the state vector, $u(k) \in \mathbb{R}^m$ is the control input vector and $y(k) \in \mathbb{R}^p$ denotes the state vector. The matrix $C \in \mathbb{R}^{p\times n}$ is the known output matrix, $f(x(k)) \in \mathbb{R}^n$ denotes the internal dynamics which is considered uncertain and $g(x(k)) \in \mathbb{R}^{n\times m}$ represents the control coefficient matrix, which is treated known, such that $\|g(x(k))\|_F \leqslant g_M$ with g_M is a positive constant.

The regulator aims at computing the optimal policy $u^*(k)$ that minimizes the infinite horizon value functional, i.e. $J(x(k))$, defined as a function of the system state and the control input, as

$$J(x(k)) = \sum_{i=k}^{\infty} L(x(k+i), u(k+i)), \tag{10}$$

with $L(x(k),u(k)) = x(k)^T Q x(k) + u(k)^T R u(k)$, given $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ as a positive semi-definite and positive definite matrices, respectively, being the cost-to-go function. The discounted value functional (10) can be written as a recursive Bellman equation given by

$$J(x(k)) = L(x(k),u(k)) + J(x(k+1))\,. \tag{11}$$

By applying the Bellman's optimal principle, optimal value functional, $J^*(x(k))$, is not a function of time though it must satisfy the discrete-time HJB equation such that $J^*(x(k)) = \min_{u(k)}(L(x(k))) + J^*(f(x(k)) + g(x(k))u(k))$. The optimal value functional, $J^*(x(k))$, is a solution to the fixed point DT HJB equation. The optimal control policy $u^*(x(k))$ that minimizes $J^*(x(k))$ can be obtained by using the stationarity condition as $\partial J^*(x(k))/\partial u(k) = 2Ru(k) + g(x(k))^T \partial J^*(x(k+1))/\partial(x(k+1)) = 0$ which results in

$$u^*(k) = -\frac{1}{2} R^{-1} g(x(k))^T \frac{\partial J^*(x(k+1))}{\partial(x(k+1))}\ . \tag{12}$$

The optimal control policy (12) is given in terms of the state vector, $x(k+1)$, and the optimal value functional, $J^*(x(k))$. However, the optimal control input cannot be computed, even if the full system state vector is available, as the gradient of the value functional with respect to $x(k+1)$ is unavailable due to the future value of the state vector [12].

Further, since finding a closed-form solution to the DT HJB equation, i.e. $J^*(x(k))$, is difficult, this article proposes a novel learning scheme for seeking an approximate solution of the value functional $\hat{J}_k(x(k))$ to the DT HJB equation by using estimated state vector obtained from a NN observer. This estimated value functional is in turn used to generate the estimated optimal policy. Thus, the need for the full state availability is relaxed through a novel observer introduced in the next section.

Fact. When the optimal control policy is asserted on (9), the closed-loop system will be bounded such that $\|f(x(k)) + g(x(k))u^*(x(k))\| \leqslant \bar{k}$ given a known constant $\bar{k}$ [12].

Remark 1. The above fact is not restrictive since the optimal control input must ensure closed-loop stability for a nonlinear system [12]. The above fact is used to show boundedness.

The value functional that is expressed by (10) can be approximated by using critic NN with two-layers as

$$J(x(k)) = w_c^T \sigma_c(x(k)) + \varepsilon_{jk}\,, \tag{13}$$

where w_c is the critic NN weights, ε_{jk} is the NN reconstruction error considered to be bounded and σ_c is the NN activation functions. Similarly, the optimal control policy given by (12) with actor NN based approximator is given by

$$u(x(k)) = w_a^T \sigma_a(x(k)) + \varepsilon_{uk}\,, \tag{14}$$

where w_a represents the actor NN weights with ε_{uk} as the NN functional reconstruction error and σ_a as the activation function. For simplicity $\sigma(x(k))$ will be shown as $\sigma(k)$. In the above formulation, the state vector availability is assumed whereas in the next section, the estimated state vector from a novel observer is considered after stating the assumption.

Assumption 2. The weights and the reconstruction errors of both critic and actor NN are upper bounded [10] such that $\|w_c\| \leqslant w_{cM}, \|w_a\| \leqslant w_{aM}, |\varepsilon_{jk}| \leqslant \varepsilon_{jM}, |\varepsilon_{uk}| \leqslant \varepsilon_{uM}$ where w_{cM},w_{aM},ε_{jM},ε_{uM} are positive constants. Moreover, the gradient of the NN reconstruction errors are bounded above as $\|\partial\varepsilon_{jK}/\partial x(k+1)\|_F \leqslant \varepsilon_{jM}^{'}$ [12].

IV. OPTIMAL ADAPTIVE OUTPUT FEEDBACK CONTROL

The OAC of the nonlinear DT system (9) by using concurrent learning is introduced, in this section. Two NNs are employed: one NN acting as the critic and another NN as an actor to approximate the value functional and optimal control policy, respectively. A novel weight tuning law incorporating the concurrent learning is proposed for the critic NNs to relax the need of PE condition. In addition, one has to ensure the closed-loop stability and boundedness of the value functional. To guarantee the convergence of the critic NN weights to the ideal values, the PE condition should be satisfied.

A. Observer Design

The dynamics of the nonlinear system (9) can be reformulated

$$\begin{cases} x(k+1) = Ax(k) + F(x(k)) + g(x(k))u(k) \\ y(k) = Cx(k) \end{cases} \tag{15}$$

where $A \in \mathbb{R}^{n\times n}$ is a Schur matrix such that (A,C) is observable, and $F(x(k)) = f(x(k)) - Ax(k)$. As shown in [13], NN can be utilized as an effective method in the estimation of nonlinear systems due to its online learning capability. Therefore, using the universal approximation property [10], the system dynamics (15) can be represented by using NN on a compact set Ω as $x(k+1) = Ax(k) + F(x(k)) + g(x(k))u(k) = Ax(k) + W_F^T\sigma_F(x(k)) + W_g^T\sigma_g(x(k))u(k) + \varepsilon_{Fk} + \varepsilon_{gk}u(k) = Ax(k) + \begin{bmatrix}\varepsilon_{Fk} & \varepsilon_{gk}\end{bmatrix}\begin{bmatrix}1\\u(k)\end{bmatrix} + \begin{bmatrix}W_F & W_g\end{bmatrix}\begin{bmatrix}\sigma_F(x(k)) & 0\\ 0 & \sigma_g(x(k))\end{bmatrix}\begin{bmatrix}1\\u(k)\end{bmatrix}$ which can be written as

$$x(k+1) = Ax(k) + W^T\sigma(x(k))\bar{u}(k) + \bar{\varepsilon}(k) \tag{16}$$

where $W = \begin{bmatrix}W_F & W_g\end{bmatrix} \in \mathbb{R}^{\ell\times n}, \sigma(x(k)) = \begin{bmatrix}\sigma_F(x(k)) & 0\\ 0 & \sigma_g(x(k))\end{bmatrix} \in \mathbb{R}^{\ell\times(1+m)}, \bar{u}(k) = \begin{bmatrix}1\\u(k)\end{bmatrix} \in \mathbb{R}^{(1+m)}$ and $\bar{\varepsilon}(k) = \begin{bmatrix}\varepsilon_{Fk} & \varepsilon_{gk}\end{bmatrix}\bar{u}(k) \in \mathbb{R}^n$, with ℓ as the number of neurons. Additionally, the target NN weights, activation function and reconstruction error are assumed to be bounded as $\|W\| \leq W_M, \|\sigma(x(k))\| \leq \sigma_M$ and $\|\bar{\varepsilon}(k)\| \leq \bar{\varepsilon}_M$, where W_M, σ_M and $\bar{\varepsilon}_M$ are positive constants. Since the states of the system are not measurable, the following observer is defined as [14]

$$\begin{cases} \hat{x}(k+1) = A\hat{x}(k) + \hat{W}(k)^T\sigma\left(\hat{x}(k)\right)\bar{u}(k) \\ +L\left(y(k) - C\hat{x}(k)\right) \\ \hat{y}(k) = C\hat{x}(k) \end{cases} \tag{17}$$

where $\hat{W}(k)$ is the estimated value of the target NN weights W at each time step $k, \hat{x}(k)$ is the estimated system state, $\hat{y}(k)$ is the estimated output and $L \in \mathbb{R}^{n\times p}$ is the observer gain matrix to be designed. Using (16) and (17), the state estimation error can be expressed as $\tilde{x}(k+1) = x(k+1) - \hat{x}(k+1) = A_c\tilde{x}(k) + \tilde{W}(k)^T\sigma\left(\hat{x}(k)\right)\bar{u}(k) + W^T\tilde{\sigma}\left(x(k),\hat{x}(k)\right)\bar{u}(k) + \bar{\varepsilon}(k)$ which can be written as

$$\tilde{x}(k+1) = A_c\tilde{x}(k) + \tilde{W}(k)^T\sigma\left(\hat{x}(k)\right)\bar{u}(k) + \bar{\varepsilon}_{Ok}, \quad (18)$$

where $A_c = A - LC$ is the closed-loop matrix, $\tilde{W}(k) = W - \hat{W}(k)$ is the NN weight estimation error, $\tilde{\sigma}\left(x(k),\hat{x}(k)\right) = \sigma\left(x(k)\right) - \sigma\left(\hat{x}(k)\right)$ and $\bar{\varepsilon}_{Ok} = W^T\tilde{\sigma}\left(x(k),\hat{x}(k)\right)\bar{u}(k) + \bar{\varepsilon}(k)$ are bounded terms due to the bounded values of ideal NN weights, activation functions and reconstruction errors. The control input is also bounded for identification whereas this is relaxed when it is combined with control. To relax the PE condition, a novel concurrent learning based tuning update law is proposed for the NN weights as

$$\begin{aligned}\hat{W}(k+1) = {} & (1-\alpha_I)\hat{W}(k) + \beta\sigma\left(\hat{x}(k)\right)\bar{u}(k)\tilde{y}(k+1)^T\Upsilon^T \\ & + \beta\sum\nolimits_{j=1}^{p}\sigma\left(\hat{x}(k_j)\right)\bar{u}(k_j)\tilde{y}(k_j+1)^T\Upsilon^T\end{aligned} \quad (19)$$

where $\alpha_I > 0, \beta > 0$ are the tuning parameters, $\tilde{y}(k+1) = y(k+1) - \hat{y}(k+1)$ is the output error, and $\Upsilon \in \mathbb{R}^{n\times p}$ is a design matrix. Note that the first term in (19) guarantees the stability of the observer weight estimation error, while the second term (19) uses the output estimation error to update the observer NN weights. The third term belongs to the concurrent learning which uses a set of system outputs. The design parameters such as observer gain have to be carefully designed to ensure faster convergence of the observer to the system. Next, using (19), the NN weight estimation error can be written as

$$\begin{aligned}&\tilde{W}(k+1) = W - \hat{W}(k+1) = \\ &(1-\alpha_I)\tilde{W}(k) + \alpha_I W - \beta\sigma(k)\bar{u}(k)\tilde{y}(k+1)^T\Upsilon^T \\ &- \beta\sum\nolimits_{j=1}^{p}\sigma(k_j)\bar{u}(k_j)\tilde{y}(k_j+1)^T\Upsilon^T\end{aligned} \quad (20)$$

which by using the state estimation error (18) yields

$$\begin{aligned}&\tilde{W}(k+1) = (1-\alpha_I)\tilde{W}(k) + \alpha_I W - \beta\Gamma(k)\tilde{W}(k)C^T\Upsilon^T \\ &- \beta\Xi(k)\tilde{x}(k)A_c^TC^T\Upsilon^T - \beta\Xi(k)\bar{\varepsilon}_{Ok}^TC^T\Upsilon^T\end{aligned} \quad (21)$$

with $\Gamma(k) = \sigma(k)\bar{u}(k)\bar{u}(k)^T\sigma(k)^T - \sum_{j=1}^{p}\sigma(k_j)\bar{u}(k_j)\bar{u}(k_j)^T\sigma(k_j)^T$ and $\Xi(k) = \sigma(k)\bar{u}(k) - \sum_{j=1}^{p}\sigma(k_j)\bar{u}(k_j)$.

Theorem 1: **(Boundedness of the observer error)**. Let the nonlinear system (9) be controllable and observable and the system output, $y(k) \in \Omega_y$, be measurable. Let the initial NN observer weights $\hat{W}(k)$ be selected from the compact set Ω_{OB} which contains the ideal weights W. Given an initial admissible control input, $u_0 \in \Omega_u$ and bounded input for all time, let the proposed observer be given by (17) and the update law for tuning the NN weights be given by (19). Then, the observer error $\tilde{x}(k)$ and the NN weight estimation errors $\tilde{W}(k)$ are uniformly ultimately bounded (UUB).

Remark 1: The boundedness of the control input assumption in the above theorem is necessary here since the control design is not done yet. This assumption is relaxed next.

B. Value Function Approximation

In this subsection, a critic NN utilizes to estimate the value functional and its weights are tuned using TDE. Next, the boundedness of the value functional is shown by constructing a Lyapunov function and using its first difference. To guarantee the convergence of the critic NN weights, the recent transition samples are stored in an experience replay buffer and also presented to the update law as part of concurrent learning [8]. Since it is assumed that the state of the system is not available, the observer state will be used in the calculation.

Let the value functional that is estimated in the form of a critic NN be given by

$$\hat{J}_k(\hat{x}(k)) = \hat{w}_c^T\sigma_c(k) \quad (22)$$

where $\hat{J}_k(\hat{x}(k))$ is the estimated value functional and $\hat{w}_c^T$ is the estimated critic NN weights. Note that $\sigma_c(k)$ is $\sigma_c(\hat{x}(k))$. Using the estimated value functional (22) in (11) results in the following TDE

$$\begin{aligned}\mathcal{E}_{TD}(k) = {} & L(\hat{x}(k-1), u(\hat{x}(k-1))) \\ & + \hat{w}_c^T\Delta\sigma_c(k-1)\end{aligned}, \quad (23)$$

where $\Delta\sigma_c(k-1) = \sigma_c(k) - \sigma_c(k-1)$. Using (13) in (11) gives $L(\hat{x}(k-1), u(\hat{x}(k-1))) = w_c^T\sigma_c(k-1) - w_c^T\sigma_c(k) - \Delta\varepsilon_{jk}$ where $\Delta\varepsilon_{jk} = \varepsilon_{jk} - \varepsilon_{jk-1}$ which by replacing in (23) yields

$$\mathcal{E}_{TD}(k) = \hat{w}_c^T\Delta\sigma_c(k-1) - w_c^T\Delta\sigma_c(k-1) - \Delta\varepsilon_{jk} \quad (24)$$

Let the critic weight estimation error be defined as $\tilde{w}_c = w_c - \hat{w}_c$, then, the TDE (23) becomes $\mathcal{E}_{TD}(k) = -\tilde{w}_c^T(k)\Delta\sigma_c(k-1) - \Delta\varepsilon_{jk}$. Using the gradient-descent scheme, the traditional critic update law can be expressed as [10]

$$\hat{w}_c(k+1) = \hat{w}_c(k) - \frac{\alpha_J\Delta\sigma_c(k)\mathcal{E}_{TD}^T(k)}{\Delta\sigma_c^T(k)\Delta\sigma_c(k)+1} \quad (25)$$

where α_J is a constant learning rate. Note that when using the update law (28), the estimated critic NN weights converge to their actual values, i.e. $\|\hat{w}_c - w\| \to 0$, if and only if $\Delta\sigma_c(k)$ is persistently exciting [10]. In the next section, the concept of concurrent learning [6] is utilized for the critic NN update law in order to relax the need of PE condition.

It is worth noting that the improved update law minimizes both the instantaneous TDE and the TDE associated with the stored transition error. The samples are stored in a history stack. To collect data online in the history stack, we evaluate values of $\sigma_c(k)$ and $L(\hat{x}(k), u(k)$ at the recorded time t_r as $\Delta\sigma_{cr} = \sigma_c(t_r) - \sigma_c(t_r - 1)$ and $L_r = L(\hat{x}(t_r), u(t_r))$. Thus, the TDE at the recorded time t_r is defined as

$$\mathcal{E}_{TDr}(t_r) = L_r + \hat{w}_c^T\Delta\sigma_{cr} \quad (26)$$

The tendency of the learned model to catastrophically forget existing information when learning from novel observation is the main reason to utilize lifelong learning in the proposed update laws. Since critic NN approximates the cost functional which in

turn is used by the actor NN to compute the control policy, the lifelong functionality is introduced only for the critic NN. Here, the NN weights of the critic network are updated to minimize the following performance measure

$$Ec = \frac{1}{2}\mathcal{E}_{TD}(k)^2 + \frac{\lambda_w}{2}||\hat{w}_c - \hat{w}_c^\star||^2_{F_w} \tag{27}$$

where λ_w set the relevance of the old tasks with respect to the new one, F_w is the fisher information matrix, and $\hat{w}_c^\star$ is the weights at the end of the previous tasks. The second term in (27) is added to the performance measure to reduce large changes in $\hat{w}_c$ when learning a new task. Therefore, the proposed novel critic NN weight update law is given by

$$\begin{aligned}\hat{w}_c(k+1) &= \hat{w}_c(k) - \frac{\alpha_J \Delta\sigma_c(k)\mathcal{E}_{TD}^T(k)}{\Delta\sigma_c^T(k)\Delta\sigma_c(k)+1} \\ &- \alpha_J \sum_{j=1}^{n} \frac{\Delta\sigma_{cj}\mathcal{E}_{TDj}^T}{\Delta\sigma_{cj}^T\Delta\sigma_{cj}+1} - \alpha_J\lambda_w F_w(\hat{w}_c(k) - \hat{w}_c^\star)\end{aligned} \tag{28}$$

where α_J is a constant learning rate and the index j stands for the j^{th} sample data ($j = 1, \ldots, n$), stored in the history stack. Let the history stack be defined as

$$Z = [\Delta\bar{\sigma}_1, \ldots, \Delta\bar{\sigma}_l] \tag{29}$$

where $\Delta\bar{\sigma}_j = \Delta\sigma_{cj}/(\Delta\sigma_{cj}^T\Delta\sigma_{cj}+1)$. Therefore, Z in the recorded data includes as many linearly independent elements as the number of neurons in (22), i,e. $rank(Z) = m$. The number of samples in the history stack is a fixed value $n > m$.

Theorem 2: Let $u_0(k)$ be any initial admissible control policy defined in a compact set. Let the critic NN weights with the experience replay tuning law be given as (28). If the history stack matrix Z is full rank, then, the estimated value functional (22) becomes bounded as the critic weight estimation error $\tilde{w}_c$ converges to the residual set $R_{sw} = \{\tilde{w}_c | \, ||\tilde{w}_c|| \leq c_w\}$ with constant $c_w > 0$.

C. Optimal Control Policy Approximation

A single layer NN is deployed for the actor network in order to obtain optimal policy by using the approximated value functional from the critic NN. The approximated control policy can be written as

$$\hat{u}(\hat{x}(k)) = \hat{w}_a^T\sigma_a(k) \tag{30}$$

where $\hat{w}_a$ is the estimated weights. Next, define the control input error which is the difference between the estimated control input (30) and the control input that minimizes the estimated cost function (22), which can be written as

$$\tilde{u}(k) = \hat{w}_a^T\sigma_a(k) + \frac{1}{2}R^{-1}\hat{g}\left(\hat{x}(k)\right)^T\frac{\partial\sigma_c(k+1)^T}{\partial x(k+1)}\hat{w}_c \tag{31}$$

where $\hat{g}(k) = W_g^T\sigma_g(\hat{x}(k))$. Substituting (13) and (14) in (12) yields

$$\begin{aligned}w_a^T\sigma_a\left(v_a^T\sigma(\hat{x}(k))\right) + \varepsilon_{uk} &= -\frac{1}{2}R^{-1}g(k)^T\frac{\partial\sigma_c(k+1)^T}{\partial x(k+1)}w_c \\ &- \frac{1}{2}R^{-1}g(k)^T\frac{\partial\varepsilon_{jk+1}}{\partial x(k+1)}\end{aligned} \tag{32}$$

Employing (32) in (31) renders $\tilde{u}(k) = \hat{w}_a^T\sigma_a(k) - w_a^T\sigma_a(k) - \varepsilon_{uk} + 1/2R^{-1}\hat{g}(k)^T\partial\sigma_c(k+1)^T/\partial x(k+1)\hat{w}_c - 1/2R^{-1}g(k)^T\partial\sigma_c(k+1)^T/\partial x(k+1)w_c - 1/2R^{-1}g(k)^T\partial\varepsilon_{jk+1}/\partial x(k+1)$. Let the actor weight and the input matrix g estimation error be defined as $\tilde{w}_a = w_a - \hat{w}_a$ and $\tilde{g} = g - \hat{g}$, respectively. Adding and subtracting $1/2R^{-1}\hat{g}(k)^T\partial\sigma_c(k+1)^T/\partial x(k+1)w_c(k)$, gives

$$\begin{aligned}\tilde{u}(k) &= -\tilde{w}_a^T(k)\sigma_a(k) - \tilde{\varepsilon}_{uk} \\ &- \frac{1}{2}R^{-1}\hat{g}(k)^T\frac{\partial\sigma_c(k+1)^T}{\partial x(k+1)}\tilde{w}_c(k), \\ &- \frac{1}{2}R^{-1}\tilde{g}(k)^T\frac{\partial\sigma_c(k+1)^T}{\partial x(k+1)}w_c(k)\end{aligned} \tag{33}$$

where $\tilde{\varepsilon}_{uk} = (1/2)R^{-1}g\left(x(k)\right)^T\left(\partial\varepsilon_{jk+1}/\partial x(k+1)\right) + \varepsilon_{uk}$. Realizing that the control input error $\tilde{u}(x(k))$ is measurable, the weight tuning laws for the actor NN is selected as follows

$$\hat{w}_a(k+1) = \hat{w}_a(k) - \frac{\alpha_u\sigma_a(k)\tilde{u}^T(k)}{\sigma_a(k)^T\sigma_a(k)+1} \tag{34}$$

where $0 < \alpha_u < 1$. Using (34), the actor NN weight estimation error dynamics becomes $\tilde{w}_a(k+1) = \tilde{w}_a(k) + \alpha_u\sigma_a(k)\tilde{u}^T(k)/\sigma_a(k)^T\sigma_a(k)+1$.

Theorem 3: (Estimated optimal control boundedness) Let $u_0(k)$ be an initial admissible control policy for (9) with value functional defined as (10). The critic NN weight update rules are given by (28), and the actor NN weights are tuned by (34). Then, under the PE condition on the actor NN, the positive constants α_u and α_J exists such that the augmented state $x(k)$, critic NN weight estimation error,$\tilde{w}_c$,$\tilde{v}_c$ and weight estimation errors of the actor, $\tilde{v}_a$, and $\tilde{w}_a$, are all UUB with the bounds given by $\|\tilde{w}_c\| \leq b'_{w_c}$,$\|\tilde{v}_c\| \leq b'_{v_c}$, $\|\tilde{w}_a\| \leq b'_{w_a}$, $\|\tilde{v}_a\| \leq b'_{v_a}$ for positive constants b'_{w_c}, b'_{v_c}, b'_{w_a} and b'_{v_a}. This assures, the closeness of the actual control input to its optimal value.

Remark 2: Notice that in the above theorem, the need for the boundedness of the control input in Theorem 1 is relaxed and demonstrated during the proof provided an initial admissible control input is selected.

V. SIMULATION RESULTS

Consider the Van der Pol oscillator [14] dynamics as $\dot{x}_1 = x_2, \dot{x}_2 = (1 - x_1^2)x_2 - x_1 + u, y = x_1$ which can be discretized by a sampling interval of $T = 10ms$ as $x_1(k+1) = x_1(k) + Tx_2(k), x_2(k+1) = x_2(k) + T((1 - x_1^2(k))x_2(k) - x_1(k) + u(k)), y(k) = x_1(k)$. The desired set point, here it is trajectory, for different tasks is described by $x_d(t) = [0, 0]^T$, $0 < t \leq 40, x_d(t) = [0.1, 0]^T$, $30 < t \leq 60, x_d(t) = [0, 0]^T$, $60 < t \leq 100$. The penalty matrices of the quadratic function value in (10) are selected as $Q = 10I_2$ and $R = 0.1$. The initial values for the state vector is given by $x_0 = \begin{bmatrix} -0.02 & 0.005 \end{bmatrix}^T$, and the initial admissible control input is given by $u_0 = -5e_2 + e_1^2e_2$ where $e_1 = x_1 - x_{d_1}$ and $e_2 = x_2 - x_{d_2}$. The observer NN is composed of 20 neurons in the hidden layer with sigmoid activation functions and the weights of NNs are randomly initialized. The learning rates β and the damping factors α_I are chosen as $\beta = 50$ and $\alpha_I = 0.5$. Moreover, the actor and critic NNs are composed of

20 and 15 neurons in the hidden layer with sigmoid and polynomial activation functions, respectively. The design parameters are selected as $\alpha_u = 0.11$ and $\alpha_J = 0.06$. The NN weights are initialized at random in the interval $[0, 1]$ and $[-0.1, 0.1]$ for critic and actor NN, respectively.

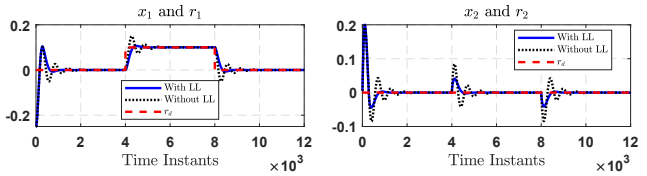

Fig. 1: The controller performance for the Van der Pol oscillator.

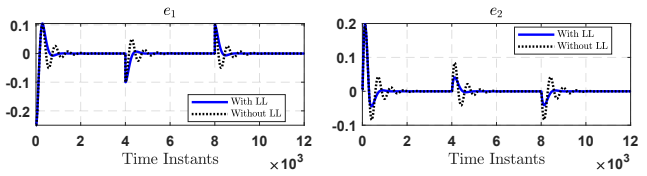

Fig. 2: Regulation errors for the Van der Pol oscillator.

Figs. 1 through 3 show the performance of the proposed approach with the lifelong learning (LL) term in (28), and results are compared with [3] without using the lifelong learning technique. A random noise is utilized for 100-time instants with the estimated control policy to ensure the PE condition is satisfied for the third case. Note that PE condition is not required in the proposed controller approach for the critic NN whereas external noise is injected for the actor.

Remark 3: Note that the proposed method can be easily extended to optimal tracking by using the augmented system in [2]. To this end, different trajectories are defined at time instants as $r(k) = e^{(-k/4)}[sin(k), cos(k) - 1/4sin(k)]^T$, $0 < k \leq 3000$, $r(k) = e^{(-k/4)}[sin(2k), 2cos(2k) - 1/4sin(2k)]^T$, $3000 < k \leq 6000$ and $r(k) = e^{(-k/4)}[sin(k), cos(k) - 1/4sin(k)]^T$, $6000 < k \leq 1000$ are considered for this scenario. In Fig. 4, the state and reference trajectories are depicted for three different cases where in the first case, the lifelong learning term in (28) is considered for completely unknown dynamic system while in the second case, the lifelong learning term is considered for the partially unknown dynamic system, and for the last case, the results are considered without the lifelong learning term for the partially unknown dynamic system. As can be seen in Fig. 4, the LL method enables faster convergence.

VI. CONCLUSION

An online OAR scheme for nonlinear DT system with uncertainties is introduced. The NN observer is able to estimate the state vector which is subsequently employed in the OAR design. The critic NN weight tuning using TDE and actor NN weight tuning using control input errors appears to generate an acceptable performance though the stability analysis is involved. In the proposed learning scheme for tuning the weights for the critic NN to relax PE, the experience replay buffer based term from concurrent learning is incorporated.

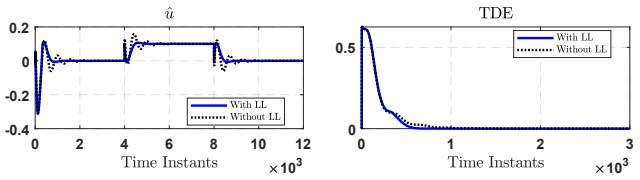

Fig. 3: The control input $\hat{u}$ and TDE $\mathcal{E}_{TD}$.

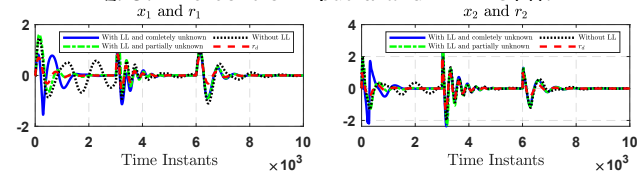

Fig. 4: The controller performance for the Van der Pol oscillator.

REFERENCES

[1] F. Al-Tamimi and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 4, pp. 943–949, 2008.

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

[3] R. Moghadam, P. Natarajan, and S. Jagannathan, "Online optimal adaptive control of partially uncertain nonlinear discrete-time systems using multilayer neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[4] 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 transactions on cybernetics*, vol. 50, no. 7, pp. 3182–3194, 2019.

[5] B. Farzanegan, A. A. Suratgar, M. B. Menhaj, and M. Zamani, "Distributed optimal control for continuous-time nonaffine nonlinear interconnected systems," *International Journal of Control*, pp. 1–15, 2021.

[6] G. Chowdhary and E. Johnson, "Concurrent learning for convergence in adaptive control without persistency of excitation," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 3674–3679, 2010.

[7] 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.

[8] 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.

[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] J. Sarangapani, *Neural Network Control of Nonlinear Discrete-time Systems*. CRC press, 2006.

[11] N. Bof, R. Carli, and L. Schenato, "Lyapunov theory for discrete time systems," *arXiv*, 2018.

[12] 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 Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1118–1129, 2012.

[13] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[14] Q. Zhao, H. Xu, and S. Jagannathan, "Near optimal output feedback control of nonlinear discrete-time systems based on reinforcement neural network learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 4, pp. 372–384, 2014.