01 Jan 2023

Continual Optimal Adaptive Tracking Of Uncertain Nonlinear Continuous-time Systems Using Multilayer Neural Networks

Irfan Ganie

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

Recommended Citation

I. Ganie and S. Jagannathan, "Continual Optimal Adaptive Tracking Of Uncertain Nonlinear Continuous-time Systems Using Multilayer Neural Networks," *Proceedings of the American Control Conference*, pp. 3395 - 3400, Institute of Electrical and Electronics Engineers, Jan 2023.
The definitive version is available at https://doi.org/10.23919/ACC55779.2023.10156466

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

Continual Optimal Adaptive Tracking of Uncertain Nonlinear Continuous-time Systems using Multilayer Neural Networks

Irfan Ganie[1] and S. Jagannathan

Abstract— **This study provides a lifelong integral reinforcement learning (LIRL)-based optimal tracking scheme for uncertain nonlinear continuous-time (CT) systems using multilayer neural network (MNN). In this LIRL framework, the optimal control policies are generated by using both the critic neural network (NN) weights and single-layer NN identifier. The critic MNN weight tuning is accomplished using an improved singular value decomposition (SVD) of its activation function gradient. The NN identifier, on the other hand, provides the control coefficient matrix for computing the control policies. An online weight velocity attenuation (WVA)-based consolidation scheme is proposed wherein the significance of weights is derived by using Hamilton-Jacobi-Bellman (HJB) error. This WVA term is incorporated in the critic MNN update law to overcome catastrophic forgetting. Lyapunov stability is employed to demonstrate the uniform ultimate boundedness of the overall closed-loop system. Finally, a numerical example of a two-link robotic manipulator supports the theoretical claims.**

Index Terms— **Lifelong learning, Reinforcement learning, Multilayer neural networks, Catastrophic forgetting, Continual learning.**

I. Introduction

Optimal control of dynamic system has been a research topic in the controls community in the recent past. Optimal control aims to identify control policies in order to minimize an objective function that is subject to system dynamics. Optimal policies can be derived either by applying the Pontryagin minimum principle from classical methods or by solving the Hamilton-Jacobi-Bellman (HJB) equation [1], [2] in dynamic programming; however, obtaining a closed-form analytical solution to the partial differential HJB equation is a major challenge. Traditionally, solutions to HJB equation [1], [2] are often offline and require a complete knowledge of system dynamics in order to obtain the control policies.

A number of online optimal adaptive approximation-based control approaches over infinite time horizon, also known as adaptive critic designs (ACDs) by using adaptive dynamic programming (ADP) framework, are introduced in [3] to the traditional offline and backward-in-time techniques. The HJB equation is solved using successive approximations by the ACD approaches [3] utilizing either value or policy iteration techniques and the solution is employed to generate optimal control policies. In contrast to value iteration-based techniques, policy iteration methods often require an initial admissible control input [3] which can be difficult to find when the system dynamics are uncertain.

In ADP, NNs have been used to provide an approximative solution to the HJB equation [1],[4] and an optimal control of CT nonlinear systems (CTNS) [5] over infinite time horizon was presented provided the control coefficient matrix can be invertible. In [6], an augmented system consisting of tracking error and desired reference trajectory overcame this restriction by transforming the optimal tracking to a regulation problem.

The above-discussed techniques [3], [4] depend on comprehensive knowledge of system dynamics which could be a bottleneck in practical applications. Integral reinforcement learning (IRL) schemes using value or policy iteration have recently been introduced in the literature [6], [7] as an alternate formulation that does not require drift dynamics. However, the control coefficient matrix is still required in [6], [7] and single-layer NN with basis functions are employed. Multilayer NN (MNN) relax the need for basis function especially when the system dynamics are uncertain whereas discovering weight tuning laws for MNN is a major challenge.

On the other hand, although NN control techniques for nonlinear systems use predominantly online learning [6], [7] using single-layer NN instead of offline training, forgetting the previously learned knowledge is a prevalent problem with NN-based techniques when these nonlinear systems operate in multitask environment. Available LL techniques such as elastic weight consolidation (EWC) [8], [9] perform well in offline scenario to mitigate the issues of catastrophic forgetting, however, certain weights can become extremely large in EWC based scheme [8],[10] thus leading to exploding gradient issues [11].

Therefore, in this paper, by using MNN-based IRL framework, a novel optimal adaptive tracking scheme is proposed for uncertain nonlinear CT systems in affine form without using policy/value iterations. This tracking scheme includes singular value decomposition (SVD) based weight tuning method for the critic MNN to overcome vanishing gradient of the activation functions. The SVD based method decomposes the gradient into singular values and singular vectors, and by analyzing and modifying the singular values, the impact of vanishing gradients on system performance can be overcome. An identifier is introduced so as to obviate the need for control coefficient dynamics. A novel online weight velocity attenuation (WVA) based LL scheme is included as part of the weight tuning for tracking to mitigate catastrophic forgetting in multitask systems. Unlike previous offline based

[1]Irfan Ganie and S. Jagannathan are with the Dept. of Elec. and Comp. Engg, Missouri University of Science and Technology, Rolla, MO, USA . iag76b@mst.edu and sarangap@mst.edu .

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

approaches [8], [11] that relied on having access to target values, in the proposed online LL method, the significance of weights is obtained by using the estimated HJB error. The net result is the development of a novel online lifelong integral reinforcement learning (LIRL)-based MNN optimal adaptive tracking control scheme.

II. PROBLEM STATEMENT

Consider a continuous-time nonlinear system given by

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

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $f(x) \in \mathbb{R}^n$, $g(x) \in \mathbb{R}^{n\times m}$ represents system states, control input, system drift dynamics, and the system input dynamics respectively. It is assumed that the drift and input dynamics are unknown but locally Lipschitz continuous in x over a set $\psi \subset \mathbb{R}^n$.

Assumption 1: Assume $r_d(t)$ be the reference trajectory governed by $\dot{r}_d(t) = f_d(r_d(t)) \in \mathbb{R}^n$ and $f(0) = 0$, $r_d(t)$ is bounded and $f_d(r_d(t))$ is Lipschitz continuous in $r_d(t)$.

Define the tracking error as

$$e_{tr}(t) = x(t) - r_d(t). \tag{2}$$

The dynamics of the tracking error can be written as

$$\dot{e}_{tr}(t) = f(e_{tr}(t)+r_d(t))+g(e_{tr}(t)+r_d(t))u(t)-f_d(r_d(t)). \tag{3}$$

The dynamics of the augmented system state vector $z = [e_{tr}^\top, r_d^\top]^\top \in \mathbb{R}^{2n}$ can be expressed as

$$\dot{z}(t) = \mathcal{F}_{aug}(z(t)) + \mathcal{G}_{aug}(z(t))u(t), \tag{4}$$

where $\mathcal{F}_{aug}(z(t)) = \begin{bmatrix} f(e_{tr}+r_d(t)) - f_d(r_d(t)) \\ f_d(r_d(t)) \end{bmatrix}$ and $\mathcal{G}_{aug}(z(t)) = \begin{bmatrix} g(e_{tr}+r_d(t)) \\ 0 \end{bmatrix}$.

Assumption 2 ([4]): The nonlinear system is controllable and observable.The control coefficient matrix satisfies $\|g(x)\| \leq g_M$, where g_M is an unknown constant.

Now the objective is to identify the optimal control input that minimizes the following cost function given by

$$J(z(t)) = \int_t^\infty e^{-\mu(s-t)} \left[z^\top(s)Q_1^z z(s) + u^\top(s)Ru(s)\right] ds, \tag{5}$$

where $Q_1^z = \begin{bmatrix} Q_1^z & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix}$ and $Q_1^z \geq 0, R = R^\top > 0, \mu$ is a discount factor. By taking the derivative of (5) along the system trajectories (4) and rearranging the terms, one can write

$$\begin{aligned} &\nabla J(\mathcal{F}_{aug}(z) + \mathcal{G}_{aug}(z)u) - \mu J(z) + z^\top Q_1^z z + \mathcal{U}(u) \\ &= \mathcal{H}_{aug}(z, u, \nabla J) = 0, \end{aligned} \tag{6}$$

where $\mathcal{H}_{aug}$ being the Hamiltonian based on augmented system, $\mathcal{U}(u)$ denotes $u^\top(s)Ru(s)$ and ΔJ represents the partial derivative of the value function J with respect to z. Let the optimal cost function $J^*(z)$ satisfies $\mathcal{H}_{aug} = 0$ and is written as

$$J^*(z) = \min_u \int_t^\infty \mathrm{e}^{-\mu(s-t)} \left[z^\top Q_1^z z + \mathcal{U}(u)\right] \mathrm{d}s. \tag{7}$$

Thus $J^* \triangleq J^*(z)$. Therefore, $\mathcal{H}_{aug}(z, u, \nabla J^*) = 0$ can be re-written as

$$\nabla J^*(\mathcal{F}_{aug}(z)+\mathcal{G}_{aug}(z)u)-\mu J^*+z^\top Q_1^z z+\mathcal{U}(u) = 0, \tag{8}$$

where ΔJ^* is the partial derivative of the value function J^* with respect to z. By taking the derivative of (8) with respect to u, i.e. $\partial\mathcal{H}_{aug}/\partial u = 0$, the optimal control policy, u^*, is obtained as

$$u^* = -\frac{1}{2}R^{-1}\mathcal{G}_{aug}^\top(z)\nabla J^*(z). \tag{9}$$

As a result, the optimal value function can be found as

$$\begin{aligned} J^*(z(t)) = \int_t^\infty e^{-\mu(s-t)} &\left[z^\top(s)Q_1^z z(s)\right. \\ &\left.+ u^{*\top}(s)Ru^*(s)\right]ds. \end{aligned} \tag{10}$$

By substituting (9) into (8) , the equivalent HJB equation for tracking can be obtained as

$$\begin{aligned} &Q^z(z) - \mu J^*(z) + \nabla J^{*\top}(z)\mathcal{F}_{aug}(z) \\ &- \frac{1}{4}\nabla J^{*\top}(z)\mathcal{G}_{aug}(z)R^{-1}\mathcal{G}_{aug}^\top(z)\nabla J^*(z) = 0, \end{aligned} \tag{11}$$

where $Q^z = zQ_1^z z$. Next, an online reinforcement learning approach using MNNs is given to solve the HJB equation approximately and generate the optimal control policies.

III. CONTINUAL MULTI-LAYER CRITIC NN CONTROL

The cost function (10) and ∇J^* can be expressed using a MNN as

$$J^*(z) = D^\top\sigma(V^\top\sigma(z)) + \epsilon(z), \tag{12}$$

$$\nabla J^*(z) = \nabla\sigma_2^\top V\nabla\sigma_1^\top D + \nabla\epsilon(z), \tag{13}$$

where $\sigma_1 = \sigma(V^\top\sigma(z)), \sigma_2 = \sigma(z), \nabla\sigma_1 = \frac{\partial\sigma_1}{\partial z}, \nabla\sigma_2 = \frac{\partial\sigma_2}{\partial z}$, $\epsilon(z)$ is the approximation error, σ_1, σ_2 are the activation functions for output and hidden layer respectively, D and V are the output and hidden layer target weights respectively.

Now to use the IRL formulation, it is necessary to integrate the infinitesimal representation of (5) throughout the time range $[t-T, t]$, where T is a fixed time interval to get

$$\begin{aligned} J(z_{t-T}) = &\int_{t-T}^{T} e^{-\mu(s-t+T)}[Q^z(z(s)) + \mathcal{U}(u(s))]ds \\ &+ e^{-\mu T}J(z(t)). \end{aligned} \tag{14}$$

Now using the approximation (13) in (9), the control input in terms of target weights can be shown as

$$u = -\frac{1}{2}R^{-1}\mathcal{G}_{aug}^\top(z)D^\top\nabla\sigma_1 V^\top\nabla\sigma_2. \tag{15}$$

Substituting (15) in (14), the HJB approximation error can be written as

$$\begin{aligned} &\int_{t-T}^{T} e^{-\mu(s-t+T)}\left[Q^z(z) + \mathcal{U}(u)\right] ds \\ &+ e^{-\mu T}D^\top\sigma(z(t)) - D^\top\sigma(z(t-T)) \equiv \varepsilon_B, \end{aligned} \tag{16}$$

where, $\varepsilon_B = \varepsilon(z(t-T)) - e^{-\mu T}\varepsilon(z(t))$.

On substituting (12) and (13) into (11), and doing a few mathematical operations, the approximated HJB equation for tracking is obtained as

$$Q^z(z) - \mu D^\top \sigma(V^\top \sigma(z)) + D^\top \nabla\sigma_1 V^\top \nabla\sigma_2 \mathcal{F}_{aug}(z) - \frac{1}{4} D^\top \nabla\sigma_1 V^\top \nabla\sigma_2 \wedge \nabla\sigma_2^\top V \nabla\sigma_1^\top D + \epsilon_{HJB} = 0, \tag{17}$$

where $\wedge = \mathcal{G}_{aug}(z) R^{-1} \mathcal{G}_{aug}(z)^\top > 0$. Since the target weights, D and V, are unknown, approximate (12), and (13) as

$$\hat{J}(z) = \hat{D}^\top \sigma(\hat{V}^\top \sigma(z)), \tag{18}$$

$$\nabla \hat{J}(z) = \hat{D}^\top \nabla\hat{\sigma}_1 \hat{V}^\top \nabla\sigma_2, \tag{19}$$

where $\hat{D}, \hat{V}$ are the estimated weights for the output and hidden layer respectively and $\hat{\sigma}_1 = \sigma(\hat{V}^\top \sigma(z))$. The estimated control input can be thus represented as

$$\hat{u} = -\frac{1}{2} R^{-1} \mathcal{G}_{aug}^\top(z) \hat{D}^\top \nabla\hat{\sigma}_1 \hat{V}^\top \nabla\sigma_2. \tag{20}$$

Next the following assumption is needed.

Assumption 3: The target weights D,V and $\sigma(z), \hat{\sigma}_1(z), \nabla\sigma(z), \nabla\hat{\sigma}_1, \epsilon(z)$ and $\nabla\epsilon(z)$ are bounded such that $\|D\| \leq D_n$, $\|V\| \leq V_n$, where D_n, V_n are unknown constants, $\|\sigma(z)\| \leq b_\sigma, \|\hat{\sigma}_1(z)\| \leq b_{\sigma 1}, \|\nabla\sigma(z)\| \leq b_{\nabla\sigma}, \|\epsilon(z)\| \leq b_\epsilon$, and $\|\nabla\epsilon(z)\| \leq b_{\nabla\epsilon}$ [6].

Remark 1: The augmented approach requires the knowledge of only the control input matrix $\mathcal{G}_{aug}$ to calculate $\hat{u}$. Because the control coefficient matrix is unknown according to equations (1) and (4), an identifier will be used.

To move on, using (18), (19) in (16), the instantaneous HJB error can be represented as

$$\int_{t-T}^{T} e^{-\mu[s-t+T)} \left[Q^z(z) + \mathcal{U}(\hat{u})\right] ds + e^{-\mu T} \hat{D}^\top \hat{\sigma}_1(z_t) - \hat{D}^\top \hat{\sigma}_1(z_{t-T}) \equiv \hat{\varepsilon}_B, \tag{21}$$

with T must be chosen as small as possible in order to maintain the equivalence between (5) and (21) [6]. Replace $\hat{u}$ in $\mathcal{U}(u)$ to get $\mathcal{U}(\hat{u})$ and rewrite (21) as

$$\hat{\varepsilon}_B \equiv \int_{t-T}^{T} e^{-\mu[s-t+T)} \left[Q^z(z) + \mathcal{U}(\hat{u})\right] ds + \hat{D} \Delta\hat{\sigma}_1, \tag{22}$$

where $\hat{\sigma}_1 = \sigma(\hat{V}^\top \sigma(z))$, $\Delta\hat{\sigma}_1 = \hat{\sigma}_1(z_1) - \hat{\sigma}_1(z_{t-T})$. Next, define $\tilde{D} = D - \hat{D}$ as the weight estimation error. The instantaneous HJB error [6] can be obtained as

$$\hat{\varepsilon}_B = D^\top \sigma_1(z_{t-T}) - e^{-\mu} D^\top \sigma_1(z_t) - \Delta\epsilon(z) + e^{-\mu} \hat{D}^\top \hat{\sigma}_1(z_t) - \hat{D}^\top \hat{\sigma}_1(z_{t-T}), \tag{23}$$

Simplifying HJB error in (23) in terms of weight estimation error by using Taylor series expansion [9] to get

$$\hat{\varepsilon}_B = -\tilde{D}^\top \hat{\sigma}_1(z) - \hat{D}^\top \nabla\hat{\sigma}_1 \tilde{V}^\top \sigma_2(z) - \tilde{D}^\top \hat{\sigma}_1(z_{t-T}) - \hat{D}^\top \nabla\hat{\sigma}_1(z_{t-T}) \tilde{V}^\top \sigma_2(z_{t-T}) + w - \Delta\epsilon(z), \tag{24}$$

where w denotes the higher order terms of the Taylor series, $w \leq c_1 + c_2 \|\tilde{V}\|$, where $c_1, p_2 > 0$, $\sigma_2(z) = \sigma(z)$, the bound for $\Delta\epsilon(z) = e^{-\mu T}\epsilon(z(t)) - \epsilon(z(t-T))$ is given by $\|\Delta\epsilon(z)\| \leq \epsilon_{\max}$. Next, an identifier to approximate the unknown $\mathcal{G}_{aug}$ is introduced that is needed for optimal control input.

A. NN identifier

An identifier is utilized to develop the estimated control coefficient matrix, denoted as $\hat{\mathcal{G}_{aug}}$, in this section. The actual control coefficient matrix $\mathcal{G}_{aug}$ is replaced by the estimated control matrix $\hat{\mathcal{G}_{aug}}$ to obtain the estimated control policy. The reconstruction error of the proposed NN identifier is considered to be bounded as a function of state vector. Define

$$\dot{z}(t) = \begin{bmatrix} \mathcal{F}_{aug}(z(t)) & 0 \\ 0 & \mathcal{G}_{aug}(z(t)) \end{bmatrix} \times \begin{bmatrix} 1_n \\ u(t) \end{bmatrix}. \tag{25}$$

The augmented control input is now defined as $\bar{u} = \begin{bmatrix} 1 \\ u(t) \end{bmatrix}$, the function approximation property of the NN can be used to represent the nonlinear system on a compact set as $\mathcal{F}_{aug}(z) = V_{\mathcal{F}}^\top \sigma_{\mathcal{F}}(z) + \varepsilon_{\mathcal{F}}(z)$, $\mathcal{G}_{aug}(z) = V_{\mathcal{G}_1}^\top \sigma_{\mathcal{G}_1}(z) + \varepsilon_{\mathcal{G}_1}(z)$, $V_{\mathcal{F}} \in \mathbb{R}^{l \times n}, V_{\mathcal{G}_1} \in \mathbb{R}^{l \times n}$ represents the target NN weight matrices and $\sigma_{\mathcal{F}}(z) \in \mathbb{R}^l, \sigma_{\mathcal{G}_1}(z) \in \mathbb{R}^{l \times m}$ represent the activation functions, and $\varepsilon_{\mathcal{F}} \in \mathbb{R}^n, \varepsilon_{\mathcal{G}_1} \in \mathbb{R}^{n \times m}$, are the NN reconstruction errors, respectively. Now, we can write

$$\dot{z}(t) = \begin{bmatrix} V_{\mathcal{F}} \\ V_{\mathcal{G}_1} \end{bmatrix}^\top \begin{bmatrix} \sigma_{\mathcal{F}}(z) & 0 \\ 0 & \sigma_{\mathcal{G}_1}(z) \end{bmatrix} \bar{u} \tag{26}$$

One can write (26) as follows

$$\dot{z}(t) = W^\top \boldsymbol{\sigma}(\boldsymbol{\xi}) \bar{u} + \varepsilon_I(z), \tag{27}$$

where $W = \begin{bmatrix} V_{\mathcal{F}}^\top & V_{\mathcal{G}_1}^\top \end{bmatrix}^\top \in \mathbb{R}^{2l \times n}$ are the augmented NN identifier weights and $\sigma(\xi) = \operatorname{diag}\left\{\sigma_{\mathcal{F}}(z), \sigma_{\mathcal{G}_1}(z)\right\}$ represents the augmented activation function for NN identifier. The definition of the NN identifier reconstruction error $\varepsilon_I(z)$ is $\varepsilon_I(z) = (\varepsilon_{\mathcal{F}}(z) + \varepsilon_{\mathcal{G}}(z)u)$. Next the following assumption is stated.

Assumption 4 ([12]): The reconstruction error of the single-layer NN identifier is bounded above such that $\|\varepsilon_I(z)\|^2 \leq b_0 \|z\|^2$ and $\|W\| \leq W_n$.

Remark 2: Because $\varepsilon_I(z)$ depends on input $u(t)$ and the system state $z(t)$, therefore, it is assumed that it is bounded by the norm of the state vector unlike [13], where $\varepsilon_I(z)$ is bounded by a constant value. One can employ a MNN for the identifier similar to the controller whereas the stability analysis will be more involved.

Since the augmented activation function $\sigma(\xi)$ of the NN identifier is known, for an accurate approximation of the nonlinear system dynamics, which will be utilized to calculate the control input coefficient matrix, a suitable weight update law for the NN weight matrix W must be derived. Define the dynamics of the NN identifier as

$$\dot{\hat{z}}(t) = \hat{W}^\top \boldsymbol{\sigma}(\boldsymbol{\xi}) \bar{u} + K(z - \hat{z}), \tag{28}$$

where $\hat{z}(t) \in \mathbb{R}^n$ represents the estimator state, $K \in \mathbb{R}^{n \times n}$ is a constant gain matrix with the state estimation error given

by

$$e_i = z - \hat{z}. \tag{29}$$

One can write the dynamics of the state estimation error as

$$\dot{e}_i = -Ke_i + \tilde{W}\sigma(\boldsymbol{\xi})\bar{u} + \varepsilon_I(z), \tag{30}$$

where $\tilde{W} = W - \hat{W}$ denotes the identifier NN weight estimation error. The NN identifier weight update law be given by

$$\dot{\hat{W}} = -\alpha_w \hat{W} + \sigma(\boldsymbol{\xi})\bar{u}e_i^\top, \tag{31}$$

where e_i is the state estimation error, $\bar{u}$ is the augmented control input vector, and $\alpha_w > 0$ is a tuning parameter.

The next subsection demonstrates how to obtain the weight update law for the critic MNN using SVD.

B. SVD-based critic NN weight tuning

A novel SVD-based technique with an exploration feature has been developed. The approach entails modifying the singular values of the gradient by adding a small amount of random noise to prevent gradient instability. The proposed method is designed to ensure the stability of the gradient.

Define the SVD of the NN activation function gradient, which is a function of time, defined as $\nabla\sigma(z) = \mathscr{P}\mathscr{S}\mathscr{D}^\top = A$. The modified SVD, denoted by $\bar{A}_i$, is obtained as

$$\bar{A}_i = \mathscr{P}\mathscr{S}\mathscr{D}^\top + \mathscr{P}\mathfrak{e}_0 I\mathscr{D}^\top, \tag{32}$$

The above equation (32) introduces the concept of adding exploration noise to the singular values of the gradient. In this equation, $\mathfrak{e}_0$ represents a small amount of random noise added to the singular values, while the right and left time-varying singular vectors remain unchanged. The input to the activation function is denoted by z, and I is an identity matrix with the same dimension as $\mathscr{S}$. By adding exploration noise with singular values of the gradient, we can avoid vanishing gradient and saddle points, and improve learning in MNNs. Additionally, note that the SVD of NN gradient method can be used to extend this development to n-layer NN.

By utilizing the improved SVD-based direct error driven learning scheme to minimize the instantaneous HJB error (24), the following theorem is stated.

Theorem 1: Consider the system (1), augmented system dynamics (4), the cost function (5), let u_0 be an initial stabilizing control policy. Let SVD based critic NN weight update laws and the the estimated optimal control input, respectively, be given by

$$\begin{aligned}\dot{\hat{D}} = \beta_1 \frac{\bar{A}_1}{\left(1 + \bar{A}_1^\top \bar{A}_1\right)}\hat{\varepsilon}_B^\top - \bar{A}_2(\hat{V}^\top X_t)^\top \hat{V}^\top \\ + \bar{A}_3(\hat{V}^\top X_{t-T})^\top \hat{V}^\top - c_0\hat{D},\end{aligned} \tag{33}$$

$$\begin{aligned}\dot{\hat{V}} = \beta_2 \frac{\bar{A}_5}{(1 + \|\bar{A}_5\|^2)}\hat{\varepsilon}_B^\top - X_t(\bar{A}_2\hat{D}\hat{D}^\top)^\top \\ + X_{t-T}(\bar{A}_3^\top \hat{D}\hat{D}^\top)^\top - c_1\hat{V},\end{aligned} \tag{34}$$

$$\hat{u} = -\frac{1}{2}R^{-1}\hat{\mathcal{G}}_{aug}^\top(z)\hat{D}^\top \bar{A}_2 \hat{V}^\top \bar{A}_4, \tag{35}$$

where $\mathscr{P}_2\mathscr{S}_2\mathscr{D}_2 = \nabla\sigma_{11}$, $\mathscr{P}_1\mathscr{S}_1\mathscr{D}_1 = \Delta\sigma_{11}$, $\bar{A}_1 = \mathscr{P}_1\mathscr{S}_1\mathscr{D}_1 + \mathscr{P}_1\gamma_1\mathscr{D}_1$. $\bar{A}_2 = \mathscr{P}_2\mathscr{S}_2\mathscr{D}_2 + \mathscr{P}_2\gamma_2\mathscr{D}_2$ $\bar{A}_3 = \mathscr{P}_3\mathscr{S}_3\mathscr{D}_3 + \mathscr{P}_3\gamma_3\mathscr{D}_3$ and $\bar{A}_4 = \mathscr{P}_4\mathscr{S}_4\mathscr{D}_4 + \mathscr{P}_4\gamma_4\mathscr{D}_4$, where $\mathscr{P}_1$,$\mathscr{D}_1$,$\mathscr{P}_2$,$\mathscr{D}_2$,$\mathscr{P}_3$,$\mathscr{D}_3$, $\mathscr{P}_4$,$\mathscr{D}_4$ are right and left singular vectors and $\mathscr{S}_1, \mathscr{S}_2, \mathscr{S}_3, \mathscr{S}_4$ are the matrices containing the singular values of $\Delta\hat{\sigma}_{11}, \nabla\hat{\sigma}_{11}, \nabla\hat{\sigma}_{12}, \nabla\hat{\sigma}_{21}$ respectively and $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ are the random noise added to singular values, c_0, c_1 are the design parameters.

Let Assumptions 1 through 4 be satisfied with u_0 being an initial stabilizing policy, and the input be persistently exciting through the addition of exploration noise, the overall closed-loop system will remain uniformly ultimately bounded (UUB). Moreover, the estimated control policy will be bounded closely to the optimal one with the bounds given by

$$\begin{cases} \|z\| > \sqrt{\dfrac{K_{01}}{\bar{K}_1}} \quad or \|e_i\| > \sqrt{\dfrac{\bar{K}_{01}}{\bar{K}_6}} \\ or \|\tilde{D}\| > \sqrt{\dfrac{K_{01}}{\bar{K}_2} + \dfrac{\bar{K}_4}{2\bar{K}_2}} \quad or \|\tilde{V}\| > \sqrt{\dfrac{K_{01}}{\bar{K}_3} + \dfrac{\bar{K}_5}{2\bar{K}_3}} \\ or \|\tilde{W}\| > \sqrt{\dfrac{\bar{K}_{01}}{\bar{K}_7}} + \dfrac{K_8}{2\bar{K}_7}. \end{cases} \tag{36}$$

where $K_{01} = \frac{\bar{K}_4^2}{2\bar{K}_2^2} + \frac{\bar{K}_5^2}{2\bar{K}_3^2} + \frac{K_8^2}{2\bar{K}_7^2}$ and $\bar{K}_i, i = 1 \ldots 8$ are constant coefficients.

Remark 3: The proposed MNN utilizes normalized gradient descent and SVD to obtain weight update laws. The proposed SVD approach helps to mitigate the impact of unstable and vanishing gradients on the NN performance and ensures more stable learning.

Next, the LL for optimal tracking is introduced.

C. Lifelong learning

The offline EWC method can quite successfully mitigate catastrophic forgetting in practice [9]. However, in order to mitigate the catastrophic forgetting and explosion of gradients simultaneously in EWC, a LL technique called weight velocity attenuation (WVA) [11] was developed. However it is limited to offline scenarios, and therefore in this work a novel online LL technique is proposed by using HJB error as targets are unavailable for online control. This technique mitigates catastrophic forgetting by incorporating a regularizer term into the loss function given by

$$\begin{aligned} L(\hat{D}, \hat{V}) \approx L_B + \frac{\lambda_1}{2}\bar{\psi}_{Dj}\left(\hat{D} - \hat{D}_{A,i}^*\right)^2 \\ + \frac{\lambda_2}{2}\bar{\psi}_{vi}\left(\hat{V} - \hat{V}_{A,i}^*\right)^2, \end{aligned} \tag{37}$$

where $L_B = \frac{1}{2}\hat{\varepsilon}_B^2$ is the loss function for the current task B, $\bar{\psi}_{Dj} = diag\{\frac{\psi_{Di}}{\psi_{Di}+1}\}$, $\bar{\psi}_{vi} = diag\{\frac{\psi_{vi}}{\psi_{vi}+1}\}$ i=1…n, ψ_{Di} and ψ_{vi}, are the significance of i-th weight of NN after learning to prior tasks and $\bar{\psi}_{Dj}, \bar{\psi}_{vj}$ where j denotes the task, are estimated same as diagonal FIM [8] which is obtained by using HJB error in contrast to [8],[11]. The $\hat{D}_{Ai}^*$, $\hat{V}_{A,i}^*$, are the optimal weights of NN when performing on task A, and α_w, α_v are the NN learning rate. The significance of the

weight shows how much the change of the weights $\hat{D}$, $\hat{V}$, will be penalized when performing the next task. The FIM for each task at each layer is found as follows. Calculate the log-likelihood function as

$$\ell(\hat{D}, z) = \log p(\hat{\varepsilon}_B|\hat{D}, z) \tag{38}$$

where $\hat{\varepsilon}_B$ is the estimated HJB error and $p(\hat{\varepsilon}_B|\hat{D}, z)$ is the probability density function of the HJB error given the input and the weights. Obtain the Jacobian matrix as

$$\mathcal{J}(\hat{D}, z) = \frac{\partial \ell(\hat{D}, z)}{\partial \hat{D}} \tag{39}$$

where $\frac{\partial \ell(\hat{D},z)}{\partial \hat{D}}$ denotes the partial derivative of the log-likelihood function with respect to the weights. Generate the estimation of FIM as

$$F = \mathbb{E}\{\mathcal{J}(\hat{D}, z) \cdot \mathcal{J}(\hat{D}, z)^\top\} \tag{40}$$

where $\mathbb{E}$ denotes the expectation of the product of the Jacobian matrix and its transpose. The diagonal FIM is given by

$$\bar{\psi}_{Dj} = \text{diag}(F) \tag{41}$$

where $\bar{\psi}_{Dj}$ represents the diagonal FIM for weight $\hat{D}$ given the task j. In a similar way the FIM can be calculated for the weight matrix $\hat{V}$ in the input layer.

Next, we can derive the additional LL term in the weight update law by using the normalized gradient descent as

$$\begin{aligned} &-\frac{\partial}{\partial \hat{D}}(L(\hat{D}, \hat{V})) = \\ &-\beta_1 \left(\frac{\partial E}{\partial \hat{D}}\right) - \lambda_1 \bar{\psi}_{Dj}(\hat{D} - \hat{D}^*_{A,i}), \end{aligned} \tag{42}$$

$$\begin{aligned} &-\frac{\partial}{\partial \hat{V}}(L(\hat{D}, \hat{V})) = \\ &-\beta_2 \left(\frac{\partial E}{\partial \hat{V}}\right) - \lambda_2 \bar{\psi}_{vj}(\hat{V} - \hat{V}^*_{A,i}), \end{aligned} \tag{43}$$

where the first term of (42), (43) can be obtained from Theorem 1. For LL, the concepts from (42) are combined with the previously defined update laws from Theorem 1. The following theorem is stated.

Theorem 2: Consider the hypothesis stated in Theorem 1, with the LIRL critic MNN tuning laws given by

$$\begin{aligned} \dot{\hat{D}} = &\beta_1 \frac{\bar{A}_1}{(1 + \bar{A}_1^\top \bar{A}_1)} \hat{\varepsilon}_B^\top - \bar{A}_2 (\hat{V}^\top X_t)^\top \hat{V}^\top - c_0 \hat{D} \\ &+ \bar{A}_3 (\hat{V}^\top X_{t-T})^\top \hat{V}^\top - \alpha_d \lambda_d \bar{\psi}_{Dj}(\hat{D} - \hat{D}^*_{A,i}), \end{aligned} \tag{44}$$

$$\begin{aligned} \dot{\hat{V}} = &\beta_2 \frac{\bar{A}_5}{(1 + \|\bar{A}_5\|^2)} \hat{\varepsilon}_B^\top - X_t (\bar{A}_2 \hat{D} \hat{D}^\top)^\top - c_1 \hat{V} \\ &+ X_{t-T} (\bar{A}_3^\top \hat{D} \hat{D}^\top)^\top - \alpha_v \lambda_v \bar{\psi}_{vj}(\hat{V} - \hat{V}^*_{A,i}), \end{aligned} \tag{45}$$

where λ_d, λ_v are the design parameters, $\beta_1, \beta_2, \alpha_d, \alpha_v$ are the NN learning rates, ψ_{Di}, ψ_{vi} are the significance of weights after each task, for weights $\hat{D}$ and $\hat{V}$ respectively, $\hat{D}, \hat{V}$ are the weights to be optimized, and $\hat{D}^*_{A,i}, \hat{V}^*_{A,i}$ are the optimized weights from the previously learned task, if Assumption 1 to, 4 holds for each task, then e, z, $x, \tilde{D}, \tilde{V}$ are UUB with the bound $\bar{K} = K_{01} + K_{reg}$ where K_{01}, which is the bound in the absence of LL, and K_{reg}, which accounts for the effect of LL.

Remark 4: By referring to equation (42), it can be observed that as the importance of the weights increases, ψ_{Di} may tend towards infinity. This occurrence results in $\frac{\psi_{Di}}{\psi_{Di}+1}$ approaching 1, effectively preventing the gradient from exploding.

Remark 5: The first part of the NN weight update laws in Theorem 2 is same as Theorem 1 whereas the second part includes regularization terms resulting from LL.

Next, the simulation results for proposed MNN based IRL with and without LL are presented.

IV. Simulation Results

We consider a two-link robotic manipulator as an example to illustrate the effectiveness of the proposed method. The dynamics of the robot manipulator are described by the following equations:

$$\dot{x} = f(x) + g(x)u, \tag{46}$$

where $x = [q_1, q_2, \dot{q}_1, \dot{q}_2]^\top$ is the state vector, and

$$\begin{aligned} f &= \left[x_3, x_4, \left(\mathcal{M}^{-1}(-\mathcal{V}_m - \mathcal{F}_d)\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} - F_s\right)^\top\right]^\top, \\ g &= \begin{bmatrix} [[0,0]^\top, [0,0]^\top, (\mathcal{M}^{-1})^\top]^\top \end{bmatrix}, \\ \mathcal{M} &= \begin{bmatrix} m_1 + 2m_3 p_2 & m_2 + m_3 p_2 \\ m_2 + m_3 p_2 & m_2 \end{bmatrix}, \\ \mathcal{V}_m &= \begin{bmatrix} -m_3 b_2 \dot{q}_2 & -m_3 b_2 (\dot{q}_1 + \dot{q}_2) \\ m_3 b_2 \dot{q}_1 & 0 \end{bmatrix}, \\ \mathcal{F}_d &= \text{diag}[4.21, 2.23], \end{aligned} \tag{47}$$

$m_1 = 3, m_2 = 0.26, m_3 = 0.345, p_2 = \cos(q_2), b_2 = \sin(q_2)$.

The reference trajectories are defined as $x_d = [\cos(t), \cos(t), -\sin(t), -\sin(t)]^\top$. For the performance index defined in equation (5), we chose penalty matrices of $Q = \text{diag}[1, 1, 1, 1]$, $R = \text{diag}[1, 1]$, and $\mu = 0.13$. The NN used in this experiment had a hidden layer with 10 neurons, and the weights were initialized randomly from a uniform distribution over the range $[0, 1]$. We employed the sigmoid function as the activation function. The initial state vector was set to $[0.3; 1.6; 0; 0]$. The learning gains were selected as follows: $\mathfrak{e}_0 = 0.24$, $\lambda = 0.46$, $c_1 = 0.16$, $\alpha_v = 0.202$, $\alpha_d = 0.36$, $\alpha_z = 0.14$, and $\gamma_{1,2,3,4} = 0.12$.

The weight update laws based on LIRL from Theorem 2 are then utilized to create the control policy for a multitask environment. In this paper, we have altered the mass of the robotic manipulator after each completed task, simulating a scenario where the manipulator picks up objects or performs tasks that alter its mass value. We have only considered a two-task scenario. The mass matrix for task 1 is selected as $m_1 = 3.0, m_2 = 0.26, m_3 = 0.345$, while for task 2 it is $m_1 = 8.5, m_2 = 3.6$, and $m_3 = 3.8$,. The state and reference trajectories, and tracking errors are shown in Figs. 1 through 2. We observer that the tracking in the case of

LIRL-based MNN control scheme is better when compared with the single-layer RVFL NN method [14] without LL. Fig. 3 depicts the 3-D plot of optimal value function and cumulative cost in the multitask scenario. The cumulative cost is lower with proposed LIRL scheme over RVFL NN without LL.

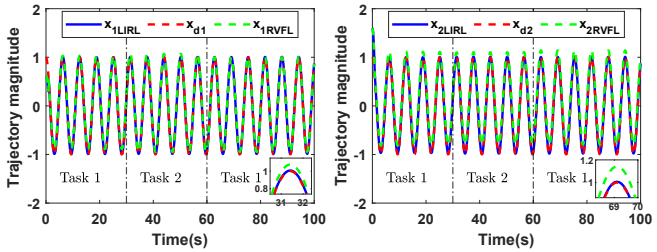

Fig. 1: Actual and reference trajectories using single-layer RVFL NN and proposed LIRL-based MNN methods.

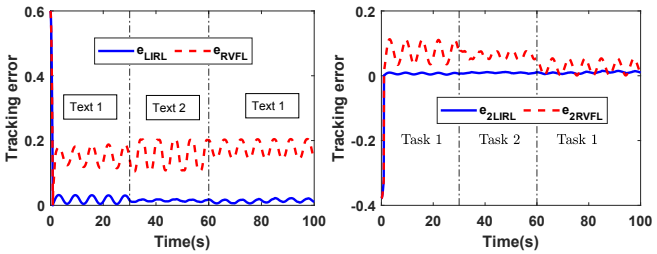

Fig. 2: Tracking errors using RVFL NN and LIRL-based MNN methods.

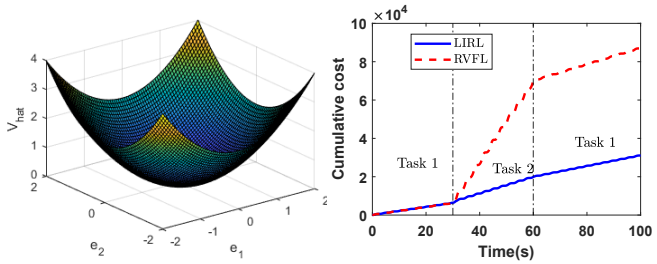

Fig. 3: Optimal value function and the cumulative cost using RVFL NN and LIRL-based MNN methods.

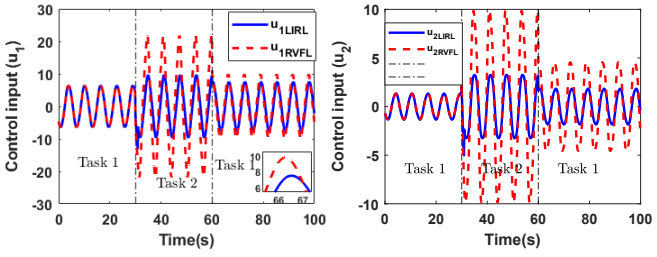

Fig. 4: Control input using RVFL NN and LIRL-based MNN.

V. Conclusion

This paper proposed a continual learning based optimal tracking technique using SVD based MNNs. The SVD based MNNs approach for optimal adaptive tracking scheme enhances the performance significantly while overcoming the vanishing action function gradient in MNNs. Despite enhanced performance, with multitask system, incorporating WVA to the critic weight update laws promoted LL by ensuring knowledge transfer between tasks while simultaneously reducing possible explosion gradient. Finally, simulation results by using a two-link robot manipulator system concur theoretical claims in a multitask environment.

References

[1] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[2] T. W. McLain and R. W. Beard, "Successive galerkin approximations to the nonlinear optimal control of an underwater robotic vehicle," *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 1, pp. 762–767 vol.1, 1998.

[3] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.

[4] T. Dierks and S. Jagannathan, "Optimal control of affine nonlinear continuous-time systems," in *Proceedings of the 2010 American Control Conference*, pp. 1568–1573, 2010.

[5] 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 Transactions on Neural Networks*, vol. 22, no. 12, pp. 2226–2236, 2011.

[6] H. Modares and F. L. Lewis, "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica*, vol. 50, no. 7, pp. 1780–1792, 2014.

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

[8] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, pp. 3521 – 3526, 2017.

[9] I. Ganie and S. Jagannathan, "Adaptive control of robotic manipulators using deep neural networks," *IFAC-PapersOnLine*, vol. 55, no. 15, pp. 148–153, 2022. 6th IFAC Conference on Intelligent Control and Automation Sciences ICONS 2022.

[10] 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, 2022.

[11] A. Kutalev and A. Lapina, "Stabilizing elastic weight consolidation method in practical ml tasks and using weight importances for neural network pruning," *ArXiv*, vol. abs/2109.10021, 2021.

[12] R. Moghadam and S. Jagannathan, "Optimal adaptive control of uncertain nonlinear continuous-time systems with input and state delays," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2021.

[13] A. Mishra and S. Ghosh, "Simultaneous identification and optimal tracking control of unknown continuous-time systems with actuator constraints," *International Journal of Control*, vol. 95, pp. 2005 – 2023, 2022.

[14] J. Na, Y. Lv, K. Zhang, and J. Zhao, "Adaptive identifier-critic-based optimal tracking control for nonlinear systems with experimental validation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 1, pp. 459–472, 2022.