01 Jan 2023

Lifelong Learning Control of Nonlinear Systems with Constraints using Multilayer Neural Networks with Application to Mobile Robot Tracking

Irfan Ganie

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

Recommended Citation

I. Ganie and S. Jagannathan, "Lifelong Learning Control of Nonlinear Systems with Constraints using Multilayer Neural Networks with Application to Mobile Robot Tracking," *2023 IEEE Conference on Control Technology and Applications, CCTA 2023*, pp. 727 - 732, Institute of Electrical and Electronics Engineers, Jan 2023.
The definitive version is available at https://doi.org/10.1109/CCTA54093.2023.10252103

2023 IEEE Conference on Control Technology and Applications (CCTA)
August 16-18, 2023. Bridgetown, Barbados

Lifelong Learning Control of Nonlinear Systems with Constraints Using Multilayer Neural Networks with Application to Mobile Robot Tracking

Irfan Ganie[1] and S. Jagannathan

2023 IEEE Conference on Control Technology and Applications (CCTA) | 979-8-3503-3544-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/CCTA54093.2023.10252103

Abstract— **This paper presents a novel lifelong multilayer neural network (MNN) tracking approach for an uncertain nonlinear continuous-time strict feedback system that is subject to time-varying state constraints. The proposed method uses a time-varying barrier function to accommodate the constraints leading to the development of an efficient control scheme. The unknown dynamics are approximated using a MNN, with weights tuned using a singular value decomposition (SVD)-based technique. An online lifelong learning (LL) based elastic weight consolidation (EWC) scheme is also incorporated to alleviate the issue of catastrophic forgetting. The stability of the overall closed-loop system is analyzed using Lyapunov analysis. The effectiveness of the proposed method is demonstrated by using a quadratic cost function through a numerical example of mobile robot control which demonstrates a 38% total cost reduction when compared to the recent literature and 6% cost reduction is observed when the proposed method with LL is compared to the proposed method without LL.**

Index Terms— **Multilayer neural networks, Singular value decomposition, Time-varying barrier functions, Lifelong learning.**

I. Introduction

The backstepping method, as outlined in the literature [1], has been extensively studied for the control of strict-feedback nonlinear systems with known and uncertain dynamics. In the backstepping design, the derivatives of the virtual control input are computed at each step, utilizing neural networks (NNs) in order to approximate the uncertain system dynamics. In the literature [2], a comprehensive NN control design by using the backstepping method is presented. In [3], an application of this method to robotic control is demonstrated. These methods, as outlined in [2] and [3], employ a Lyapunov-based control design and stability analysis utilizing single-layer NN for strict feedback nonlinear systems.

Approximation of nonlinear systems using single-layer NN requires either an appropriate selection of basis functions or the input to the hidden-layer weights selected at random with a huge number of hidden-layer neurons [4]. Despite these weaknesses, most control techniques employ single-layer NNs [2] since it is challenging to develop a multilayer NN (MNN) weight tuning method [5] though an MNN can relax the need for the selection of basis functions. Additionally, MNNs require fewer neurons per layer than shallow NNs to achieve the same level of accuracy in function approximation [6] even though closed-loop stability analysis becomes more difficult.

The development of online MNN-based adaptive control for general nonlinear systems has been presented recently in the literature [7]. In this approach, the inner-layer weights are trained in an iterative manner using offline techniques during discrete training periods, whereas the output-layer weights are updated online. While the stochastic gradient descent (SGD) approach can be used to tune MNN weights, proving closed-loop stability is extremely difficult, even when targets are available. For closed-loop feedback control applications, targets are unavailable when the nonlinear dynamics are uncertain, which makes supervised training less appealing. Due to the lack of online MNN with n-hidden layer weight tuning and stability analysis methods, control of nonlinear systems using MNN has not been explored in the literature.

On the other hand, recent literature [8] emphasizes the importance of incorporating constraints into control design since safety, which implies adhering to state, output, and input constraints, is crucial for autonomous driving, industrial robots, and aerospace vehicles. The use of barrier Lyapunov functions (BLFs) [9] in control design is an acceptable way to handle constraints. The application of BLFs in controlling nonlinear systems with constant state constraints is included in [8] with or without the use of tangent BLFs [10]. The authors in [9] developed a logarithmic barrier function-based control scheme for nonlinear systems with an output constraint. The NN tracking control of a robot manipulator using tangent BLF with time-varying joint space constraints is developed in [10] as in reality, many practical systems have time-varying constraints [11].

Besides safety, in practical applications, learning-based systems often operate in complex, dynamic, and task-dependent environments. To adapt to these changing conditions, a NN must be retrained by interleaving data for each task, as described in [12]. However, when control schemes use online learning instead of offline training, it is common for the NN to forget previous tasks, resulting in catastrophic forgetting [13] or significant model drift. To address this issue, control schemes should employ lifelong learning (LL) techniques [13].

Despite the proven effectiveness of LL in mitigating the problem of catastrophic forgetting in NNs, its implementation has been limited to offline scenarios [13] and has yet to be applied to real-time control. The integration of LL strategies into control design is an unexplored area of research with the potential to improve system performance. The incorporation of LL enables a control scheme to adapt

[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.

dynamically to new operating conditions and environments, thereby increasing its robustness.

Therefore, in this paper, we introduce a new adaptive MNN with n hidden layer control scheme for nonlinear systems in the strict feedback form that ensures safety. We use singular value decomposition (SVD) to derive the weight tuning for MNN with n hidden layers, which eliminates the need for basis functions, reduces the vanishing gradient problem, and guarantees the state vector remains within the time-varying barriers. Our control scheme employs a tangent-type BLF (TBLF) to handle both constrained and unconstrained situations, leading to a more flexible and efficient control design. Further, the weight tuning is modified by including elastic weight consolidation (EWC) terms to prevent catastrophic forgetting. Lyapunov analysis (not included) shows the boundedness of the closed-loop system.

II. System Description and Background

This section provides an introduction to the problem statement, along with a foundational overview of functional approximation and TBLF.

A. Class of Nonlinear Systems and Problem Formulation

Let's contemplate a nonlinear system in a strict feedback pattern as follows

$$\begin{aligned}\dot{x}_1(t) &= f_1(x_1) + g_1(x_1)x_2 + d_1(x_1,t),\\ \dot{x}_2(t) &= f_2(\bar{x}_2) + g_2(\bar{x}_2)u + d_2(x_2,t),\end{aligned} \tag{1}$$

where $\bar{x}_2(t) = [x_1(t), x_2(t)]^\top \in \mathbb{R}^2$ is the system state, $u \in \mathbb{R}$ is the control input. The functions $g_1(\bar{x}_1), g_2(\bar{x}_2)$ are well-defined smooth functions; $f_1(\bar{x}_1)$, and $f_2(\bar{x}_2)$ denote uncertain smooth nonlinear functions, while $d_1(\bar{x}_1,t)$, and $d_2(\bar{x}_2,t)$ represent unknown bounded disturbances.

The undermentioned mild assumptions and Lemma derived from existing literature are important for the control design.

The state vector is known and confined to constraints: $\Omega_x := \{x_i(t) \in \mathbb{R}, |x_i(t)| < k_{c_i}(t), i = 1,2, \forall t \geq 0\}$, $k_{c_i}(t) \in \mathbb{R}^+$ are pre-specified time-dependent continuous functions that are differentiable up to the nth order.

Moreover, we invoke the following assumptions.

Assumption 1 ([5]): The functions $g_i(.)(i = 1,2)$ are known and $\exists, g_0 > 0$ such that $0 < g_0 \leq |g_i(.)|$. Furthermore, it is assumed that $g_i(.)$ are all positive.

Assumption 2 ([11]): The system reference trajectory $r_d(t)$ and the time-varying state constraints $k_{c_i}(t)$ are all known, bounded, continuous, and differentiable up to the nth order. There are positive constants d_{ij} and $Y_j(i = 1,2, j = 0,1,2)$, such that $\left|r_d^{(j)}(t)\right| \leq Y_j$ and $\left|k_{ci}^{(j)}(t)\right| \leq d_{ij}, \forall t \geq 0$

Assumption 3 ([9]): An unknown constant upper boundary d_{iM} exists for the input disturbance $d_i \leq d_{iM}$.

Lemma 1 ([9]): Given bounded initial conditions, if there exists a C^1 continuous and positive definite Lyapunov function $V(x)$ that satisfies $\gamma_1(|x|) \leq V(x) \leq \gamma_2(|x|)$ and $\dot{V}(x) = \frac{\partial V}{\partial x}\dot{x} \leq -\kappa V(x) + c$, where $\gamma_1, \gamma_2 : \mathbb{R}^n \to \mathbb{R}$ are class ϑ functions and κ, c are positive constants, then the solution $x(t)$ is uniformly ultimately bounded (UUB).

B. Approximation of Functions

MNNs, due to their inherent universal function approximation properties, can represent nonlinear functions with adequate accuracy. Given any continuous function $h(X)$ within a bounded set Ω_X, an approximation by MNNs can be represented as $h^*(X) = W^{*\top}\sigma\left(V^{*\top}X\right)$, ensuring that $\sup_{X\in\Omega_X}|h^*(X) - h(X)| \leq \epsilon$. Here, $X \in \mathbb{R}^n$ is the input vector. The function $h(X)$ can be approximated as

$$h(X) = W^{*\top}\sigma\left(V^{*\top}X\right) + \epsilon(X), \quad \forall X \in \Omega_X \subset \mathbb{R}^m,$$

where W^*, V^* are the NN weights, σ represents the activation function, and ϵ denotes the approximation error. As assumed in [14], the target weights and the reconstruction error are bound above by unknown constants, represented as W_M, V_M, and ϵ_{jM}, respectively.

Next, we will introduce background on the TBLF and the SVD approach for weight tuning.

C. Time-Varying Barrier Lyapunov Function

In this study, we explore the case of full state constraints and broaden the application of TBLF to encompass all stages of the backstepping design. This approach enables us to impose constraints on each error signal $\xi_1 = x_1 - r_d$, and $\xi_2 = x_2 - \alpha_1$ where r_d, α_1 signifies the reference trajectory and virtual control input respectively. To deal with the state constraints, we introduce the TBLF as

$$V_i^* = \frac{\vartheta_i^2(t)}{\pi}\tan\left(\frac{\pi\xi_i^2(t)}{2\vartheta_i^2(t)}\right), \quad |\xi_i(0)| < \vartheta_i(0), i = 1,2 \tag{2}$$

where $\xi_i \in \Omega_e := \{\xi_i \in \mathbb{R}, |\xi_i(t)| < \vartheta_i(t),\ i = 1,2, \forall t \geq 0\}$. The terms $\vartheta_1(t)$, and $\vartheta_i(t)$ are the constraints on the error variable ξ_i, and ξ_i respectively, with $\vartheta_1(t) = k_{c1} - Y_0 > 0$, and $\vartheta_i(t) = k_{ci} - \bar{\alpha}_{i-1,0} > 0, \bar{\alpha}_{i-1,0}$ is positive constant $i = 1,2$.

Remark 1: Based on the expression of TBLF as shown in (2), it can be stated that

$$\left\{ \lim_{\xi_i \to 0} V_i^* = 0, \lim_{\xi_i \to k_i} V_i^* = \infty \right. \tag{3}$$

Remark 2: When there are no constraints on the system state vector, that is $k_{c_i} \to \infty$, we have $\vartheta_i \to \infty$, $i = 1,2$. Using L'Hospital's rule, we have

$$\lim_{\vartheta_i \to \infty}\frac{\vartheta_i^2}{\pi}\tan\left(\frac{\pi\xi_i^2}{2\vartheta_i^2}\right) = \frac{1}{2}\xi_i^2.$$

Consequently, if the state vector remains unconstrained, the TBLF bears similarity to quadratic forms, unlike what's suggested in [9], [11]. Given this, for the integration of constraint analysis into a universal method, the TBLF is favored over the time-dependent logarithmic-type BLF [11] employed in previous literature.

Next, the SVD of NN gradients will be presented.

D. Singular Value Decomposition of NN Gradients

In order to overcome the issue of vanishing gradients, unlike [15], our method utilizes SVD of gradient and includes an exploration feature. Here, a small random noise is added to alter the singular values of the gradient to promote exploration, prevent gradient instability and enhance the learning process.

Define the SVD of the NN activation function gradient defined as $\dot{\sigma}(\hat{\bar{x}}) = A_y = \mathfrak{W}\Sigma\mathfrak{V}^\top$, where $\hat{\bar{x}} = \hat{V}^\top x$,with $\hat{V}$ as the estimated hidden-layer NN weight matrix, $\mathfrak{W}, \mathfrak{V}$ are the right and left singular vector respectively and Σ represents the matrix of singular values. The modified SVD is denoted by $\dot{A}_y$ and it is obtained as

$$\dot{A}_y = \mathfrak{W}\Sigma\mathfrak{V}^\top + \mathfrak{W}\mathfrak{e}_0 I\mathfrak{V}^\top, \tag{4}$$

where $\mathfrak{e}_0$ denotes a slight random perturbation introduced to the singular values while the time-dependent right and left singular vectors remain same, and I stands for an identity matrix of equivalent dimension to Σ. The SVD approach overcomes saddle points and vanishing gradients. Next, the safe lifelong MNN (SLMNN) controller design is introduced.

III. Lifelong Multilayer NN Control

The goal of the SLMNN control scheme is to achieve satisfactory tracking and maintain the boundedness of all closed-loop system signals in the presence of disturbances while also addressing the issue of catastrophic forgetting.

A. Multiayer NN Backstepping Control Design

A two-layer neural network (NN) controller is devised utilizing the two step backstepping approach. For convenience, time 't' is omitted from here on.

Let the error system be defined as

$$\begin{aligned} \xi_1 &= x_1 - r_d, \\ \xi_2 &= x_2 - \alpha_1, \end{aligned} \tag{5}$$

where r_d is the desired output, and α_1 is the virtual control input. Differentiating (5) with respect to t to get

$$\dot{\xi}_1 = \dot{x}_1 - \dot{r}_d, \tag{6}$$

Substituting (1) in (6) to obtain

$$\dot{\xi}_1 = f_1(x_1) + g_1(x_1)x_2 + d_1(.) - \dot{r}_d. \tag{7}$$

Define $U(X_1) = f_1(x_1) - \dot{r}_d$. A two-layer NN is used to approximate $U(X_1)$ on a compact set Ω_1 as

$$U_1(X_1) = W_1^{*\top}\sigma\left(V_1^{*\top}(X_1)\right) + \epsilon_1(X_1), \tag{8}$$

where, $\epsilon_1(.)$ represents the approximation error of the NN, with W_1^* and V_1^* as the target weights of the NN. The activation function is symbolized as σ and $X_1 = [x_1, r_d, \dot{r}_d]$ within Ω_1 serves as the NN input. The virtual controller α_1 is

$$\begin{aligned} \alpha_1 = \frac{1}{g_1}(&\bar{\vartheta}_1\xi_1 - \frac{\varphi_1}{2} - \hat{W}_1^\top\sigma\left(\hat{V}_1^\top(X_1)\right) \\ &- \frac{\left(k_1\vartheta_1^2\sin\left(\frac{\pi\xi_1^2}{2\vartheta_1^2}\right)\cos\left(\frac{\pi\xi_1^2}{2\vartheta_1^2}\right)\right)}{\pi\xi_1}). \end{aligned} \tag{9}$$

Upon differentiation of (5) with respect to time, we can express $\dot{\xi}_2$ as $\dot{x}_2 - \dot{\alpha}_1$. Hence, by following the same procedure in (6), we acquire

$$\dot{\xi}_2 = f_2(\bar{x}_2) + g_2(\bar{x}_2)u + d_2(.) - \dot{\alpha}_1. \tag{10}$$

Let's define $U_2(X_2) = f_2(\bar{x}_2) - \dot{\alpha}_1$. A two-layer NN is utilized to approximate $U_2(X_2)$ on a compact set Ω_2:

$$U_2(X_2) = W_2^{*\top}\sigma\left(V_2^{*\top}(X_2)\right) + \epsilon_2(X_2), \tag{11}$$

where $\epsilon_2(.)$ is the approximation error of the NN, and W_2^* and V_2^* stand for the target weights of the NN, and $X_2 = [x_2, \alpha_1, \frac{\partial\alpha_1}{\partial x_1}, \omega_1] \in \Omega_2$ is the NN input, $\omega_1 = [\frac{\partial\alpha_1}{\partial r_d}\dot{r}_d + \frac{\alpha_1}{\partial\hat{W}_1}\dot{\hat{W}}_1 + \frac{\alpha_1}{\partial\hat{V}_1}\dot{\hat{V}}_1]$. Hence, the actual control input, denoted by u, is chosen in accordance with the demonstration provided in Theorem 1 as

$$\begin{aligned} u = \frac{1}{g_2(.)}\Bigg(&(\bar{\vartheta}_2 - \frac{\varphi_1 g_1(.)}{\varphi_2})\xi_2 - \frac{\varphi_2}{2} - \hat{W}_2^\top\sigma\left(\hat{V}_2^\top(X_2)\right) \\ &- \frac{\left(k_2\vartheta_2^2\sin\left(\frac{\pi\xi_2^2}{2\vartheta_2^2}\right)\cos\left(\frac{\pi\xi_2^2}{2\vartheta_2^2}\right)\right)}{\pi\xi_2}\Bigg), \end{aligned} \tag{12}$$

where $\hat{W}_2$, and $\hat{V}_2$ are the estimated NN weights, $\varphi_j = \sec^2\left(\frac{\pi\xi_j^2}{2\vartheta_j^2}\right)\xi_j, j = 1, 2$, $\hat{Z}_2 = diag[\hat{W}_2, \hat{V}_2]$, k_2, ϑ_2, K_{c2} and $\bar{\vartheta}_2$ are the design constants. Next, the following theorem is stated.

Theorem 1: Consider a strict feedback system as detailed in equations (1), and complying with Assumptions 1 to 3. Let the following proposed weight tuning laws employing TBLF and SVD based approach be

$$\dot{\hat{W}}_j = \Gamma_j\left(-\varphi_j\left(\hat{\sigma}_j - \dot{A}_{y,j}\hat{V}_j^\top(X_j)\right) - \gamma_{1,j}\hat{W}_j\right), \tag{13}$$

$$\dot{\hat{V}}_j = G_j\left(-\varphi_j\hat{W}_j^\top\dot{A}_{y,j}X_j - \gamma_{2,j}\hat{V}_j\right), \tag{14}$$

In the equations above, Γ_j and G_j stand for positive definite design matrices, $\gamma_{1,j}$ and $\gamma_{2,j}$ are designer-determined quantities, X_j symbolizes the NN input, $\varphi_j = \sec^2\left(\frac{\pi\xi_j^2}{2\vartheta_j^2}\right)\xi_j$, $\dot{A}_{y,j}$ denotes the SVD of gradient inclusive of exploration, and j signifies the number of backstepping steps.

The tracking errors, $\xi_j(t)$, and the weight approximation errors, $\tilde{Z}_j = diag\{\tilde{W}_j, \tilde{V}_j\}; j = 1, 2$, are assured to be uniformly bounded (UUB), while the state vector x_j remains within a bounded set. The bounding limits are defined as

$$\begin{aligned} \Omega_\xi = \{\xi_j : &\|\xi_j\| \\ &\le \vartheta_j\sqrt{\frac{2}{\pi}tan^{-1}\left(\frac{\pi}{\vartheta_j^2}\left(\left(V_2(0) - \frac{C}{\kappa_j}\right)e^{-\kappa_j t} + \frac{C}{\kappa_j}\right)\right)}\Bigg\}, \end{aligned} \tag{15}$$

$$\begin{aligned} \Omega_{\tilde{z}} = \Big\{\tilde{Z}_j : &\|\tilde{Z}_j\| \\ &\le \sqrt{\frac{2}{\Psi_{min}(\chi_j^{-1})}\left(\left(V_2(0) - \frac{C}{\kappa_j}\right)e^{-\kappa_j t} + \frac{C}{\kappa_j}\right)}\Bigg\}, \end{aligned} \tag{16}$$

In the aforementioned equations, C and κ_j are the positive constants, provided $k_j \ge 2\bar{\vartheta}_j$, $j = 1, 2$, $\chi_j = diag\{\Gamma_j, G_j\}$, and the initial conditions ensure that $\xi_j(0) \in \Omega_\xi := \left\{\xi_j \in \mathbb{R}^j : \|\xi_j\| < \vartheta_j\right.$.

Finally, each state x_j is retained within the confined set Ω_x, where

$$\Omega_x := \{x_j(t) \in \mathbb{R}, |x_j(t)| < k_{c_j}, j = 1, 2, \forall t \geq 0\}.$$

In addition, all closed-loop signals are bound and the full state constraints are not violated.

Remark 3: The weight tuning laws postulated in Theorem 1 are two-fold, the first component involves time-based barrier expressions coupled with the SVD of the gradient, while the latter segment embodies the sigma modification term, introduced to relax the persistence of excitation (PE) condition.

Remark 4: The SVD based MNN used in this paper exhibits superior approximation abilities in contrast to single-layer NNs, thereby enhancing the control effectiveness and reducing errors. This improvement is consistently demonstrated in our simulations.

Next, a LL approach referred to as EWC is introduced.

B. Lifelong Learning for Multilayer NN Controller

Catastrophic forgetting, where learning performance on previous tasks can degrade with new tasks, is a challenge for NNs in continual learning. To address this, the method of LL [13] has been proposed, allowing NNs to learn continuously without interfering with previously learned tasks. One technique for LL in NNs is EWC [13], which avoids catastrophic forgetting by adding a penalty term to the loss function to keep the network parameters near optimal parameters of prior tasks, preserving previously learned knowledge.

However, the application of EWC [13] has been limited to offline scenarios and has not yet been applied to real-time control systems. Therefore, the MNN weight tuning is modified to include the EWC terms.

The performance function is given by

$$L = L_b + \frac{\Psi_w}{2}\|\hat{W} - \hat{W}^*_{\tau_i}\|^2_{F_w} + \frac{\Psi_v}{2}\|\hat{V} - \hat{V}^*_{\tau_i}\|^2_{F_v}, \quad (17)$$

where L is the overall loss function, L_b is the loss just for task B, $\|\hat{W} - \hat{W}^*_{\tau_i}\|^2_{F_w}$ denotes $(\hat{W} - \hat{W}^*_{\tau_i})^\top F_w(\hat{W} - \hat{W}^*_{\tau_i})$, $\|\hat{V} - \hat{V}^*_{\tau_i}\|^2_{F_v}$ denotes $(\hat{V} - \hat{V}^*_{\tau_i})^\top F_v(\hat{V} - \hat{V}^*_{\tau_i})$, F_w and F_v are Fisher Information Matrices (FIM) corresponding to the output and hidden layers respectively, as outlined in [14]. The vectors $\hat{W}$ and $\hat{V}$ are used to represent the parameters that need to be optimized. Furthermore, the optimized parameters that are bounded from the previous task are denoted by $\hat{W}^*_{\tau_i}$ and $\hat{V}^*_{\tau_i}$. The outcome of EWC is a regularization term to be incorporated with the NN weight tuning to enable LL without forgetting.

By invoking the gradient of (17), the regularization term for the weight update laws can be obtained as

$$\begin{aligned} -\frac{\partial}{\partial \hat{W}}(L) &= -\bar{\mu}_1 \Psi_w F_w(\hat{W} - \hat{W}^*_{\tau_i}), \\ -\frac{\partial}{\partial \hat{V}}(L) &= -\bar{\mu}_2 \Psi_v F_v(\hat{V} - \hat{V}^*_{\tau_i}), \end{aligned} \quad (18)$$

The FIM, F_w and F_v, help assess the importance of NN weights $\hat{W}$ and $\hat{V}$ in retaining memory of old tasks. The parameters Ψ_v and Ψ_w control the memory recall, with greater values retaining more memory but limiting the ability to learn new tasks. Smaller values decrease memory recall, putting it closer to simple gradient descent. $\bar{\mu}_1$ and $\bar{\mu}_2$ are the learning rates, and these terms in (18) are included in the previously defined update laws from Theorem 1 for LL in control. The online LL-based EWC for MNNs in adaptive control can be summarized as follows:

(1) Initialize a velocity vector for each weight in the NN controller, which will be used to track the change in weight values over time. (2) Compute the velocity of each weight by taking the difference between the current weight value and the previous weight value and storing it in the corresponding velocity vector during the learning process. (3) Compute the importance of each weight for the current task using a measure such as the diagonal FIM. (4) Incorporate a penalty term in the control problem that penalizes large changes in the weights that are important for the current task. (5) Use the penalty term as a regularization term in the control problem to minimize the sum of the original objective function and the penalty term using an optimization algorithm such as gradient descent. This approach helps to slow down the changes in the important weights and prevent catastrophic forgetting. Next, the following theorem is stated.

Theorem 2: Consider the hypothesis outlined in Theorem 1, assuming Assumptions 1 to 3 hold true. The weight update laws for SLMNN are given by:

$$\begin{aligned} \dot{\hat{W}}_j = \Gamma_j(&-\varphi_j\left(\hat{\sigma}_j - \dot{A}_{y,j}\hat{V}_j^\top(X_j)\right) - \gamma_{2,j}\hat{W}_j \\ &- \bar{\mu}_{1,j}\Psi_{w,j}F_{w,j}(\hat{W}_j - \hat{W}^*_{\tau_i,j})), \end{aligned} \quad (19)$$

$$\begin{aligned} \dot{\hat{V}}_j = G_j(&-\varphi_j\hat{W}_j^\top\dot{A}_{y,j}X_j - \gamma_{2,j}\hat{V}_j \\ &- \bar{\mu}_{2,j}\Psi_{v,j}F_{v,j}(\hat{V}_j - \hat{V}^*_{\tau_i,j})), \end{aligned} \quad (20)$$

These equations demonstrate an increase in error bounds while ensuring UUB of the system. The optimal weights for the previous task are denoted as $\hat{W}^*_{\tau,j}$ and $\hat{V}^*_{\tau,j}$, with their boundedness proven in Theorem 1. $F_{w,j}$ and $F_{v,j}$ represent the FIM [14], while the weight update is influenced by the learning rates, $\bar{\mu}_{1,j}$ and $\bar{\mu}_{2,j}$, as well as the design parameters, $\Psi_{w,j}$ and $\Psi_{v,j}$. The variable j represents the backstepping steps.

Remark 5: The NN weight update laws above include the terms from Theorem 1 and LL-terms.

IV. Simulation Results and Discussion

In this section, we validate the theoretical design using a mobile robot tracking example. The overall representation of a mobile robot is given by

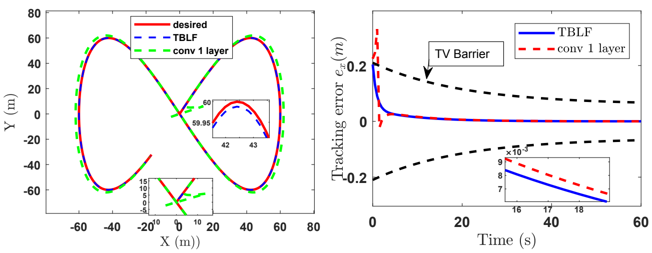
$$\begin{aligned} \dot{\eta} &= J(\eta)\nu, \\ M\dot{\nu} &= -C(\nu)\nu - D(\nu)\nu + \tau + \tau_w(t), \end{aligned} \quad (21)$$

where $\eta = [x, y, \psi]^\top$, where x, y, and ψ are the actual Cartesian position and orientation of the robot, respectively,

L is the distance from the rear axle to the front of the robot, $\nu = [v, \omega]^{\top}$, where v, and ω are linear and angular velocities, respectively, $\tau = [\tau_u, \tau_r]^{\top}$ denotes the control inputs, whereas τ_w denotes the external time-varying disturbances. In the robot dynamics, $D(\nu)$ represents damping matrix, $C(\nu)$ is the Coriolis matrix, M is the inertia matrix, and $J(\eta)$ is the rotation matrix.

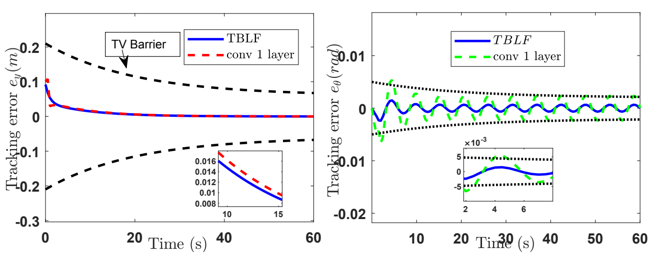
The simulation results are divided into two parts (i) single task using multilayer SVD based TBLF, (ii) multitask using multilayer SVD-based TBLF and multitask using a multilayer safe lifelong learning method.

A. Single Task using Multilayer TBLF



(a) (b)

Fig. 1: (a) Phase plane trajectories and positional tracking error for single task using TBLF and RVFLNN method

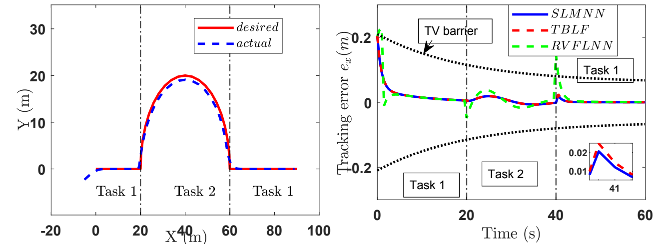


(a) (b)

Fig. 2: (a) Position tracking error e_y; (b) Orientation error e_θ for a single task using TBLF and RVFLNN method.

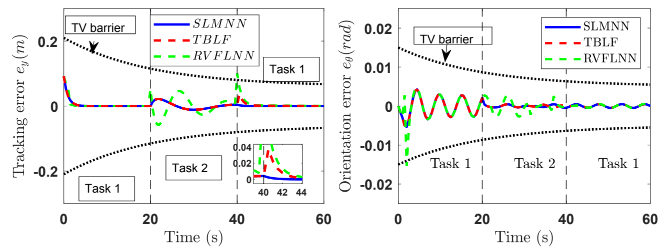
This section presents the performance of the multilayer NN based TBLF method, without the LL term (TBLF), on a single task. The task parameters, including $p_0 = [60\sin(0.01t), 60\sin(0.02t)]^T$, $m = 10kg$, $I = 5kg - m^2$, $R = 0.5m$, and the initial position $= [-5.1, -2.3, 0.5]$, as well as design parameters $\gamma = 1$, $\epsilon_0 = 0.2$, $\Psi = 10$, $K_c = 1$, $k_{d1} = k_{\theta 1} = 4$, $k_u = 3$, $k_r = 3$, $\Gamma = \text{diag}\{8\}$, $G = \text{diag}\{5\}$, and barrier constraints $\vartheta_d(t) = 0.15\exp(-0.05t) + 0.06$, $\vartheta_\theta(t) = 0.003\exp(-0.05t) + 0.002$ are defined. The hidden-layer had 10 neurons, and initial conditions of weight matrices were randomized within [0,1].

Figures 1 and 2 reveal the proposed method's superior trajectory tracking and error containment within barrier constraints, compared to the RVFLNN method shown as 'conv 1 layer' [16]. Figures 3 and 4 further illustrate that both positional tracking and orientation errors are confined within barrier limits, highlighting the improved performance of the proposed method in a single-task scenario.



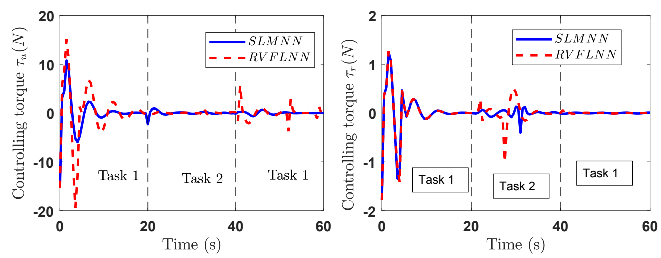
(a) (b)

Fig. 3: Performance of SLMNN, TBLF, and RVFLNN methods for multitask system:(a) Phase plane trajectories of position, (b) Positional tracking error.



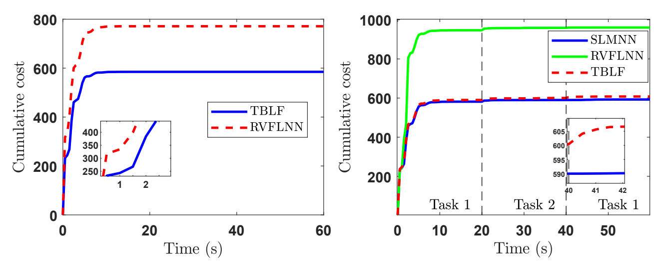
(a) (b)

Fig. 4: Performance of SLMNN, TBLF and RVFLNN:(a) Positional tracking error e_u, and (b) Orientation error e_θ.



(a) (b)

Fig. 5: Performance of SLMNN and RVFLNN methods: (a) Input torque 1 and (b) Input torque 2.



(a) (b)

Fig. 6: Cumulative cost using SLMNN, TBLF and RVFLNN methods:: (a) Single task, (b) Multitask system.

B. Multitask using Multilayer TBLF Lifelong Learning

In this part, the mobile robot tracking results using SLMNN and safe multilayer NN TBLF without LL (denoted as TBLF) are compared for multitask system, with single-layer random vector functional link NN (RVFLNN) method [16] without TBLF and LL. The simulation is run for two different tasks where trajectories are changed for each of them. Task one is run for a duration of 0 to 20 secs with trajectories and dynamics parameters as $p_0(t) = [t(k), 0]^\top, m = 10kg, I = 5kg - m^2, R = 0.5m$ and initial position= $[-5.1, -2.3, 0.5]$. Task two is run for a duration of 20 to 40 secs with desired trajectory as $p_0(t) = [40 - 20cos(t-20), sin(t-20)]^\top$, after 40 secs, task one is again run.

The design parameters are given as $\gamma = 1, \epsilon_0 = 0.2, \Psi = 10, K_c = 1, k_{d1} = k_{\theta 1} = 4, k_u = 3, k_r = 3$, Γ =diag 8, G =diag 5. The constraints of the barrier are articulated as $\vartheta_d(t) = 0.15\exp(-0.05t) + 0.06$ and $\vartheta_\theta(t) = 0.01\exp(-0.05t) + 0.05$. A sigmoid function is employed as the activation function. The hidden-layer of the NN contains ten neurons and the weight matrices were randomly initialized within the range of 0 to 1.

The phase-plane position trajectories of the mobile robot in Fig. 3a demonstrate the effectiveness of the proposed SLMNN learning approach. The robot is able to track the desired trajectory even when the dynamics and path change every 20 seconds. Fig. 3b illustrates that the tracking errors for the multilayer TBLF without LL, and SLMNN, remain within the barrier limits. The errors for the multilayer TBLF are lower as compared to RVFLNN, because of the better approximation capabilities of SVD based MNN and the utilization of barriers. However the errors are not eliminated completely during the task changes and within the tasks because of catastrophic forgetting of NNs, therefore the addition of LL (SLMNN) makes the errors lower than TBLF and RVFLNN during the task changes and within the tasks and when the robot returns to task1. In contrast, the RVFLNN method results in higher tracking errors that violate the constraints, as shown in Figs. 4a and 4b. The lower positional tracking and orientation errors in Fig. 4a and compliance to constraints further demonstrate the superiority of SLMNN over RVFLNN. Additionally, figures 5a and 5b depict that SLMNN requires less torque when compared to RVFLNN.

The cumulative cost is defined as the integral of a quadratic function of the tracking error and torque inputs over time, represented by the equation $C(t) = \int_t^\infty \left[e^T(s)Qe(s) + u^T(s)Ru(s)\right] ds$ with $Q = 4$, $R = 0.1$, and $u = \{\tau_1, \tau_2\}$. It can be observed from 6a that the cumulative cost is lower when using the SLMNN method over TBLF using MNN but without LL and RVFLNN. It is been observed that the overall cost reduction between RVFLNN and SLMNN is around 38% and between TBLF and SLMNN the cost reduction is around 6%. Therefore, it can be concluded that the performance of the SLMNN-based control technique is superior to that of the single-layer RVFLNN and TBLF without LL methods.

V. Conclusion

A novel SLMNN approach that can effectively handle time-varying constraints and uncertain dynamics is proposed. This approach utilizes an SVD of NN gradients with exploration, which improves the stability and robustness of the learning process while overcoming the vanishing gradient. The proposed approach incorporates techniques to mitigate the catastrophic forgetting that arises in multi-task scenarios when using traditional online control methods. The stability of the overall closed-loop system is rigorously analyzed through the use of Lyapunov theory, ensuring that the system signals remain bounded despite the presence of constraints and LL. The superior performance of the proposed SLMNN over the conventional single-layer NN method is demonstrated through a comprehensive simulation of a mobile robot control example. Future work will focus on further relaxing the control coefficient matrix.

References

[1] J. Zhou, C. Wen, and Y. Zhang, "Adaptive backstepping control of a class of uncertain nonlinear systems with unknown backlash-like hysteresis," *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1751–1759, 2004.

[2] X. Zhao, X. Wang, S. Zhang, and G. Zong, "Adaptive neural backstepping control design for a class of nonsmooth nonlinear systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 9, pp. 1820–1831, 2019.

[3] J. Sarangapani and F. Lewis, "Robust backstepping control of robotic systems using neural networks," *Journal of Intelligent and Robotic Systems*, vol. 23, pp. 105–128, 10 1998.

[4] N. Le Roux and Y. Bengio, "Deep belief networks are compact universal approximators," *Neural Computation*, vol. 22, no. 8, pp. 2192–2207, 2010.

[5] 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*, pp. 1–11, 2021.

[6] D. Rolnick and M. Tegmark, "The power of deeper networks for expressing natural functions," *Proc. Int. Conf. Learn. Represent.*, 05 2017.

[7] R. Sun, M. L. Greene, D. M. Le, Z. I. Bell, G. Chowdhary, and W. E. Dixon, "Lyapunov-based real-time and iterative adjustment of deep neural networks," *IEEE Control Systems Letters*, vol. 6, pp. 193–198, 2022.

[8] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022.

[9] K. P. Tee, S. Ge, and F. Tay, "Barrier lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, pp. 918–927, 2009.

[10] H. N. Rahimi, I. Howard, and L. Cui, "Neural adaptive tracking control for an uncertain robot manipulator with time-varying joint space constraints," *Mechanical Systems and Signal Processing*, vol. 112, pp. 44–60, 2018.

[11] Y.-J. Liu, L. Ma, L. Liu, S. Tong, and C. L. P. Chen, "Adaptive neural network learning controller design for a class of nonlinear systems with time-varying state constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 1, pp. 66–75, 2020.

[12] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2017.

[13] J. Kirkpatrick, R. Pascanu, N. 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, no. 13, pp. 3521–3526, 2017.

[14] I. Ganie and S. Jagannathan, "Adaptive control of robotic manipulators using deep neural networks," *Conference on Intelligent Control and Automation Sciences* $ICONS$, vol. 55, pp. 148–153, 2022.

[15] M. Roodschild, J. Gotay Sardiñas, and A. Will, "A new approach for the vanishing gradient problem on sigmoid activation," *Progress in Artificial Intelligence*, vol. 9, pp. 351–360, 2020.

[16] O. Mohareri, R. Dhaouadi, and A. B. Rad, "Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks," *Neurocomputing*, vol. 88, pp. 54–66, 2012.