IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 June 2025, accepted 25 June 2025, date of publication 30 June 2025, date of current version 11 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3584253

RESEARCH ARTICLE

Lyapunov-Based Deep Residual Neural Network (ResNet) Adaptive Control

OMKAR SUDHIR PATIL, DUC M. LE, EMILY J. GRIFFIS, AND WARREN E. DIXON, (Fellow, IEEE)
Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA
Corresponding author: Omkar Sudhir Patil (patilomkarsudhir@ufl.edu)

This work was supported in part by the Office of Naval Research under Grant N00014-21-1-2481 and in part by Air Force Office of Scientific Research (AFOSR) under Award FA8651-21-F-1027 and Award FA9550-19-1-0169.

ABSTRACT Deep Neural Network (DNN)-based controllers have emerged as a tool to compensate for unstructured uncertainties in nonlinear dynamical systems. A recent breakthrough in the adaptive control literature provides a Lyapunov-based approach to derive weight adaptation laws for each layer of a fully-connected feedforward DNN-based adaptive controller. However, deriving weight adaptation laws from a Lyapunov-based analysis remains an open problem for deep residual neural networks (ResNets). This paper provides the first result on Lyapunov-derived weight adaptation for a ResNet-based adaptive controller. A nonsmooth Lyapunov-based analysis is provided to guarantee asymptotic tracking error convergence. Comparative Monte Carlo simulations are provided to demonstrate the performance of the developed ResNet-based adaptive controller. The ResNet-based adaptive controller shows a 64% improvement in the tracking and function approximation performance, in comparison to a fully-connected DNN-based adaptive controller.

INDEX TERMS Deep neural networks, ResNets, adaptive control, Lyapunov-based methods.

I. INTRODUCTION

Deep Neural Network (DNN)-based controllers have emerged as a tool to compensate for unstructured uncertainties in nonlinear dynamical systems. The success of DNN-based controllers is powered by the ability of a DNN to approximate any continuous function over a compact domain [1]. A popular DNN-based control method is to first perform a DNN-based offline system identification using sampled input-output datasets that are collected by conducting experiments [2, Sec. 6.6]. Then, using the identified DNN, a feedforward term is constructed to compensate for uncertainty in the system. However, the DNN weight estimates are not updated during task execution, and hence, such an approach involves static implementation of the DNN-based feedforward term. Since there is no continued learning with most DNN methods, questions arise regarding how well the training dataset matches the actual uncertainties in the system and the value or quality of the static feedforward model. This strategy motivates the desire for a large training dataset, but such data can be expensive or impossible to obtain, including the need for higher order derivatives that are typically not measurable.

Unlike offline methods, a closed-loop adaptive feedforward term can be developed by deriving real-time DNN weight adaptation laws from a Lyapunov-based stability analysis. Various classical results [3], [4], [5], [6], [7] use Lyapunov-based techniques to develop weight adaptation laws, but only for single-hidden-layer networks. These results do not provide weight adaptation laws for DNNs with more than one hidden layer, since there are mathematical challenges posed by the nested and nonlinear parameterization of a DNN that preclude the development of inner-layer weight adaptation laws. Recent results such as [8], [9], [10] develop Lyapunov-based adaptation laws for the output-layer weights of a DNN. However, to update the inner-layer weights, results in [8], [9], and [10] collect datasets over discrete time-periods and iteratively identify the inner-layer weights using offline training algorithms. To circumvent offline identification of the inner-layer weights, the result in [11] provides a real-time

[0]The associate editor coordinating the review of this manuscript and approving it for publication was Yizhang Jiang.

inner-layer weight adaptation scheme based on a modular approach. However, modular designs only offer constraints on the adaptation laws and do not provide constructive insights on designing the adaptation laws.

Our recent work in [12] provides the first result on Lyapunov-derived weight adaptation laws for each layer of a DNN-based adaptive controller. To address the challenges posed by the nested and nonlinearly-parameterized structure of the DNN, a recursive representation of the DNN is developed. Then, a first-order Taylor series approximation is recursively applied for each layer. Using a Lyapunov-based stability analysis, the inner- and outer-layer weight adaptation laws are designed to cancel coupling terms that result from the approximation strategy. Although the result in [12] provides Lyapunov-derived weight adaptation laws for the DNN, the development is restricted to fully-connected DNNs.

There are several limitations associated with standard DNN architectures such as fully-connected and convolutional DNNs. Deeper networks typically suffer from the problem of vanishing or exploding gradients, i.e., the rate of learning using a gradient-based update rule is highly sensitive to the magnitude of DNN weights. Challenges faced from the vanishing or exploding gradient problem are ubiquitous to both offline training [13] and real-time weight adaptation [12]. Additionally, in applications such as image recognition, the performance of a DNN is found to initially improve by increasing the depth of the DNN. However, as the depth exceeds a threshold, performance rapidly degrades [14].

To overcome the vanishing or exploding gradient problem and the degradation of performance with the increasing depth of a DNN, results in [14] introduce shortcut connections across layers, i.e., a feedforward connection between layers that are separated by more than one layer. DNNs with a shortcut connection are known as deep residual neural networks (ResNets). Offline results in [15] and [16] offer mathematical explanations for why ResNets perform better than non-residual DNNs. In [15], the parameterization of a non-residual DNN is shown to cause difficulties in training DNN layers to approximate the identity function. As explained in [15], for a DNN to achieve a good training accuracy, the DNN layers must be able to approximate the identity function well. Since a shortcut connection in ResNets is represented using an identity function, ResNets provide an improved performance when compared to non-residual DNNs. Additionally, the result in [16] provides explanations from Lyapunov stability theory on why ResNets are easier to train offline using the gradient descent algorithm as compared to non-residual DNNs. The shortcut connections in ResNets facilitate the stability of the equilibria of gradient descent dynamics for a larger set of step sizes or initial weights as compared to non-residual DNNs.

Although there has been significant research across various applications involving ResNets [14], [17], [18], [19], [20], the approximation power of ResNets has not yet been explored for adaptive control problems. Developing a ResNet-based adaptive feedforward control term with real-time weight adaptation laws is an open problem. Although real-time weight adaptation laws are developed for fully-connected feedforward DNNs in [12], the shortcut connections in ResNets pose additional mathematical challenges. Unlike fully-connected DNNs, the shortcut connection prevents a recursive application of Taylor series approximation for each layer of the ResNet. As a result, it is difficult to generate the coupling terms that are generated using the approximation strategy in [12], that can be canceled using the weight adaptation laws in the Lyapunov-based analysis.

Our preliminary work in [21] and this paper provide the first result on Lyapunov-derived adaptation laws for the weights of each layer of a ResNet-based adaptive controller for uncertain nonlinear systems. To overcome the mathematical challenges posed by the residual network architecture, the ResNet is expressed as a composition of building blocks that involve a shortcut connection across a fully-connected DNN. Then, a constructive Lyapunov-based approach is provided to derive weight adaptation laws for the ResNet using the gradient of each DNN building block. A nonsmooth Lyapunov-based analysis is provided to guarantee asymptotic tracking error convergence. Unlike our preliminary work in [21], which involved a ResNet with only one shortcut connection, this paper provides weight adaptation laws for a general ResNet that has an arbitrary number of shortcut connections. The development of adaptation laws for ResNets with an arbitrary number of shortcut connections is challenging due to the complexity of the architecture. This challenge is addressed by constructing a recursive representation of the ResNet which involves a composition of an arbitrary number of building blocks. Then, based on the recursive representation of the ResNet architecture, a first-order Taylor series approximation is applied, which is then utilized to yield the Lyapunov-based adaptation laws. Additionally, unlike our preliminary work in [21] which did not provide simulations, this paper provides comparative Monte Carlo simulations to demonstrate the performance of the developed ResNet-based adaptive controller, and the results are compared with an equivalent fully-connected DNN-based adaptive controller [12]. Since the performance of ResNet and DNN-based adaptive controllers is sensitive to weight initialization, the Monte Carlo approach is used to provide a fair comparison between the two architectures. In the Monte Carlo comparison, 10,000 simulations are performed, where the initial weights in each simulation are selected from a uniform random distribution, and a cost function is evaluated for each simulation. Then, the simulation results yielding the least cost for both architectures are compared. The ResNet-based adaptive controller shows a 64% improvement in the tracking and function approximation performance, in comparison to a fully-connected DNN-based adaptive controller.

Notation and Preliminaries:

The space of essentially bounded Lebesgue measurable functions is denoted by $\mathcal{L}_\infty$. The right-to-left matrix product

operator is represented by $\overset{\curvearrowleft}{\prod}$, i.e., $\overset{\curvearrowleft}{\prod_{p=1}^{m}} A_p = A_m \ldots A_2 A_1$ and $\overset{\curvearrowleft}{\prod_{p=a}^{m}} A_p = I$ if $a > m$. The Kronecker product is denoted by $\otimes$. Function compositions are denoted using the symbol $\circ$, e.g., $(g \circ h)(x) = g(h(x))$, given suitable functions g and h. The notation $\overset{a.a.t.}{(\cdot)}$ denotes that the relation $(\cdot)$ holds for almost all time (a.a.t.). Consider a Lebesgue measurable and locally essentially bounded function $h : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$. The Filippov set-valued map for h at $(y, t) \in \mathbb{R}^n \times \mathbb{R}_{\geq 0}$ is defined as the intersection of convex closures of values attained by h in every neighborhood of y omitting sets of measure zero, i.e.,

$$K[h](y,t) \triangleq \bigcap_{\delta>0} \bigcap_{\mathrm{m}\mathbb{S}=0} \overline{\mathrm{co}}\, h(\mathbb{B}(y,\delta) \setminus \mathbb{S}, t),$$

where $\bigcap_{\mathrm{m}\mathbb{S}=0}$ denotes the intersection over all sets $\mathbb{S} \subset \mathbb{R}^n$ of Lebesgue measure zero [22]. Then, the function $y : \mathcal{I} \to \mathbb{R}^n$ is called a Filippov solution of $\dot{y} = h(y, t)$ on the interval $\mathcal{I} \subseteq \mathbb{R}_{\geq 0}$ if y is absolutely continuous on $\mathcal{I}$ and $\dot{y} \overset{a.a.t.}{\in} K[h](y, t)$.[1] Given $w \in \mathbb{R}$ and some functions f and g, the notation $f(w) = \mathcal{O}^m(g(w))$ means that there exists some constants $M \in \mathbb{R}_{>0}$ and $w_0 \in \mathbb{R}$ such that $\|f(w)\| \leq M \|g(w)\|^m$ for all $w \geq w_0$. Given some matrix $A \triangleq [a_{i,j}] \in \mathbb{R}^{n \times m}$, where $a_{i,j}$ denotes the element in the i^{th} row and j^{th} column of A, the vectorization operator is defined as $\mathrm{vec}(A) \triangleq [a_{1,1}, \ldots, a_{1,m}, \ldots, a_{n,1}, \ldots, a_{n,m}]^T \in \mathbb{R}^{nm}$. The p-norm is denoted by $\|\cdot\|_p$, where the subscript is suppressed when $p = 2$. The Frobenius norm is denoted by $\|\cdot\|_F \triangleq \|\mathrm{vec}(\cdot)\|$. Given any $A \in \mathbb{R}^{p \times a}$, $B \in \mathbb{R}^{a \times r}$, and $C \in \mathbb{R}^{r \times s}$, the vectorization operator satisfies the property [24, Proposition 7.1.9]

$$\mathrm{vec}(ABC) = (C^T \otimes A)\mathrm{vec}(B). \tag{1}$$

Differentiating (1) on both sides with respect to vec (B) yields the property

$$\frac{\partial}{\partial \mathrm{vec}(B)} \mathrm{vec}(ABC) = (C^T \otimes A). \tag{2}$$

II. UNKNOWN SYSTEM DYNAMICS AND CONTROL DESIGN

Consider a control-affine nonlinear dynamic system modeled as

$$\dot{x} = f(x) + u, \tag{3}$$

where $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ denotes any arbitrary Filippov solution[2] to (3), $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes an unknown differentiable drift vector field, and $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ denotes a control input.[3] Let the tracking error $e : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ be defined as

$$e \triangleq x - x_d, \tag{4}$$

where $x_d : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ denotes a continuously differentiable reference trajectory. The reference trajectory is designed such that $\|x_d(t)\| \leq \overline{x_d}$ $\forall t \in \mathbb{R}_{\geq 0}$ and $\dot{x}_d \in \mathcal{L}_\infty$, where $\overline{x_d} \in \mathbb{R}_{>0}$ is a constant. The control objective is to design a ResNet-based adaptive controller that achieves asymptotic tracking error convergence.

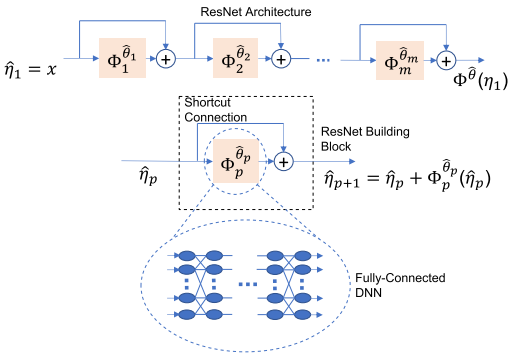

FIGURE 1. Illustration of the ResNet architecture in (6). The ResNet is shown at the top of the figure and is composed of building blocks that involve a shortcut connection across a fully-connected DNN component. The fully-connected DNN component for the p^{th} building block (bottom) is denoted by $\Phi_p^{\hat{\theta}_p}$ for all $p \in \{1, \ldots, m\}$, where the input and the vector of weights of Φ_p are denoted by η_p and $\hat{\theta}_p$, respectively. Then the output of the p^{th} building block after considering the shortcut connection is represented by $\eta_{p+1} = \eta_p + \Phi_p^{\hat{\theta}_p}(\eta_p)$ for all $p \in \{1, \ldots, m-1\}$, and the output of the ResNet is $\eta_m + \Phi_m^{\hat{\theta}_m}(\eta_m)$.

A. ResNet ARCHITECTURE

The unknown drift vector field f can be approximated using a ResNet. A ResNet is modeled using building blocks that involve a shortcut connection across a fully-connected DNN [14]. Let $\Phi_p : \mathbb{R}^{L_{p,0}} \times \mathbb{R}^{L_{p,0} \times L_{p,1}} \times \ldots \times \mathbb{R}^{L_{p,k_p} \times L_{p,k_p+1}} \to \mathbb{R}^{L_{p,k_p+1}}$ denote the p^{th} fully-connected DNN block defined as $\Phi_p(\eta_p, V_{p,0}, \ldots, V_{p,k_p}) \triangleq \left(V_{p,k_p}^T \phi_{p,k_p} \circ \ldots \circ V_{p,1}^T \phi_{p,1}\right)\left(V_{p,0}^T \eta_p\right)$ for all $p \in \{1, \ldots, m\}$, where $\eta_p \in \mathbb{R}^{L_{p,0}}$ denotes the input of Φ_p, $k_p \in \mathbb{Z}_{>0}$ denotes the number of hidden layers in Φ_p, and $m \in \mathbb{Z}_{>0}$ denotes the number of building blocks. Additionally, $L_{p,j} \in \mathbb{Z}_{>0}$ denotes the number of nodes, and $V_{p,j} \in \mathbb{R}^{L_{p,j} \times L_{p,j+1}}$ denotes the weight matrix in the j^{th} layer of Φ_p for all $(p, j) \in \{1, \ldots, m\} \times \{0, \ldots, k_p\}$. Similarly, $\phi_{p,j} : \mathbb{R}^{L_{p,j}} \to \mathbb{R}^{L_{p,j}}$ denotes a vector of smooth activation functions.[4] If the ResNet involves multiple types of activation

[1] For a detailed discussion on Filippov solutions and its comparison with other notions of solutions to discontinuous differential equations, such as Aizerman-Pyatnitskii and Gelig-Leonov-Yakubovich, the reader is referred to [23].

[2] The Filippov solutions may not be unique. In this paper, we analyze the convergence properties of all Filippov solutions beginning from a set of initial conditions.

[3] The control effectiveness term is omitted to better focus on the specific contributions of this paper without loss of generality. The method in [8] can be used with the developed method in the case where the system involves an uncertain control effectiveness term.

[4] For the case of DNNs with nonsmooth activation functions (e.g., rectified linear unit (ReLU), leaky ReLU, maxout etc.), the reader is referred to [12] where a switched analysis is provided to account for the nonsmooth nature of activation functions. To better focus on our main contribution without loss of generality, we restrict our attention to smooth activation functions.

functions at each layer, then $\phi_{p,j}$ may be represented as $\phi_{p,j} \triangleq \left[\varsigma_{p,j,1} \cdots \varsigma_{p,j,L_{p,j}}\right]^T$, where $\varsigma_{p,j,i} : \mathbb{R} \to \mathbb{R}$ denotes the activation function at the i^{th} node of the j^{th} layer of Φ_p.[5] All the weights of Φ_p can be represented by the vector $\theta_p \triangleq \left[\text{vec}(V_{p,0})^T \ \ldots \ \text{vec}(V_{p,k_p})^T\right]^T \in \mathbb{R}^{\sum_{j=0}^{k_p} L_{p,j}L_{p,j+1}}$. The fully-connected block Φ_p can be expressed as $\Phi_p = V_{p,k_p}^T \varphi_{p,k_p}$, where $\varphi_{p,0} : \mathbb{R}^{L_{p,0}} \to \mathbb{R}^{L_{p,0}}$ and $\varphi_{p,j} : \mathbb{R}^{L_{p,0}} \times \mathbb{R}^{L_{p,0}\times L_{p,1}} \times \ldots \times \mathbb{R}^{L_{p,j-1}\times L_{p,j}} \to \mathbb{R}^{L_{p,j}}\ \forall j \in \{1, \ldots, k_p\}$ denote the recursive relation defined as

$$\varphi_{p,j} \triangleq \begin{cases} \phi_{p,j}\left(V_{p,j-1}^T \varphi_{p,j-1}\right), & j \in \{1, \ldots, k_p\}, \\ \eta_p, & j = 0. \end{cases} \tag{5}$$

The arguments of $\varphi_{p,j}$ are suppressed for notational brevity. Let $\phi'_{p,j} : \mathbb{R}^{L_{p,j}} \to \mathbb{R}^{L_{p,j}\times L_{p,j}}$ be defined as $\phi'_{p,j}(y) \triangleq \frac{\partial}{\partial y}\phi_{p,j}(y)\ \forall y \in \mathbb{R}^{L_{p,j}}$. The short-hand notation $\Phi_p^{\theta_p}(\eta_p) \triangleq \Phi_p(\eta_p, V_{p,0}, V_{p,1}, \ldots, V_{p,k_p})$ is defined for notational brevity in the subsequent development. Then the output of the p^{th} building block is given by $\eta_p + \Phi_p^{\theta_p}(\eta_p)$, where the addition of the input term η_p represents the shortcut connection across Φ_p. As shown in Figure 1, the ResNet $\Phi : \mathbb{R}^n \times \mathbb{R}^{\sum_{p=1}^{m}\sum_{j=0}^{k_p} L_{p,j}L_{p,j+1}} \to \mathbb{R}^n$, which defines the mapping $(\eta_1, \theta) \mapsto \Phi^\theta(\eta_1)$, is modeled as [14]

$$\Phi^\theta(\eta_1) \triangleq \eta_m + \Phi_m^{\theta_m}(\eta_m), \tag{6}$$

where $\theta \triangleq \left[\theta_1^T \ \ldots \ \theta_m^T\right]^T \in \mathbb{R}^{\sum_{p=1}^{m}\sum_{j=0}^{k_p} L_{p,j}L_{p,j+1}}$ denotes the vector of weights for the entire ResNet, and η_m is evaluated using the recursive relation

$$\eta_p = \begin{cases} \eta_{p-1} + \Phi_{p-1}^{\theta_{p-1}}(\eta_{p-1}), & p \in \{2, \ldots, m\}, \\ x, & p = 1. \end{cases} \tag{7}$$

The recursive relation in (7) has valid dimensions under the constraint $L_{1,0} = L_{1,k_1+1} = L_{2,0} = L_{2,k_2+1} = \ldots = L_{m,0} = L_{m,k_m+1} = n$. To facilitate the subsequent development, the following assumption is made.

Assumption 1: The function space of ResNets given by (6) is dense in $\mathcal{C}(\Omega)$ with respect to the supremum norm, where $\mathcal{C}(\Omega)$ denotes the space of functions continuous over the compact set $\Omega \subset \mathbb{R}^n$.

Remark 1: Assumption 1 implies that ResNets satisfy the universal function approximation property that is well-known for various DNN architectures [1]. The universal function approximation property of ResNets is a common assumption that is widely used in the deep learning literature, and has been rigorously established for ResNets with specific activation functions in [25] and [26].

Consider any vector field $f \in \mathcal{C}(\Omega)$ and a prescribed accuracy $\overline{\varepsilon} \in \mathbb{R}_{>0}$. Then by Assumption 1, there exists a ResNet Φ with sufficiently large $m, k_p, L_{p,j}\ \forall (p,j) \in \{1, \ldots, m\} \times \{0, \ldots, k_p\}$ and a corresponding vector of ideal weights $\theta^* \in \mathbb{R}^{\sum_{p=1}^{m}\sum_{j=0}^{k_p} L_{p,j}L_{p,j+1}}$ such that $\sup_{x\in\Omega} \left\| f(x) - \Phi^{\theta^*}(x) \right\| \leq \overline{\varepsilon}$. Therefore, the drift vector field $x \mapsto f(x)$ can be modeled as[6]

$$f(x) = \Phi^{\theta^*}(x) + \varepsilon(x), \tag{8}$$

when $x \in \Omega$, where $\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$ denotes an unknown function reconstruction error that can be bounded as $\sup_{x\in\Omega} \|\varepsilon(x)\| \leq \overline{\varepsilon}$.

To facilitate the subsequent analysis, the following assumption is made (cf., [27, Assumption 1]).

Assumption 2: There exists a known constant $\overline{\theta} \in \mathbb{R}_{>0}$ such that the unknown ideal ResNet weights can be bounded as $\|\theta^*\| \leq \overline{\theta}$.

B. CONTROL AND ADAPTATION LAWS

The ResNet-based model in (8) can be leveraged to approximate the unknown drift vector field f. However, since the ideal weights are unknown, adaptive weight estimates are developed. The adaptive weight estimate for the j^{th} layer of Φ_p is denoted by $\hat{V}_{p,j} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{L_{p,j}\times L_{p,j+1}}\ \forall (p,j) \in \{1, \ldots, m\} \times \{0, \ldots, k_p\}$. The weight estimate for the p^{th} building block $\hat{\theta}_p : \mathbb{R}_{\geq 0} \to \mathbb{R}^{\sum_{j=0}^{k_p} L_{p,j}L_{p,j+1}}$ is defined as $\hat{\theta}_p = \left[\text{vec}(\hat{V}_{p,0})^T, \ldots, \text{vec}(\hat{V}_{p,k_p})^T\right]^T$ for all $p \in \{1, \ldots, m\}$, the weight estimate for the ResNet $\hat{\theta} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{\sum_{p=1}^{m}\sum_{j=0}^{k_p} L_{p,j}L_{p,j+1}}$ is defined as $\hat{\theta} \triangleq \left[\hat{\theta}_1^T \ \ldots \ \hat{\theta}_m^T\right]^T$, and the ResNet-based adaptive estimate of $f(x)\ \forall x \in \Omega$ is denoted by $\Phi^{\hat{\theta}}(x)$. The weight estimation error $\tilde{\theta} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{\sum_{p=1}^{m}\sum_{j=0}^{k_p} L_{p,j}L_{p,j+1}}$ is defined as $\tilde{\theta} \triangleq \theta^* - \hat{\theta}$. Based on the subsequent stability analysis, the adaptation law for the weight estimates of the ResNet in (6) is designed as

$$\dot{\hat{\theta}} \triangleq \Gamma \Phi'^T e, \tag{9}$$

where $\Gamma \in \mathbb{R}^{\sum_{p=1}^{m}\sum_{j=0}^{k_p} L_{p,j}L_{p,j+1} \times \sum_{p=1}^{m}\sum_{j=0}^{k_p} L_{p,j}L_{p,j+1}}$ denotes a positive-definite adaptation gain matrix, and $\Phi' \in \mathbb{R}^{n\times \sum_{p=1}^{m}\sum_{j=0}^{k_p} L_{p,j}L_{p,j+1}}$ is a short-hand notation denoting the $\Phi' \triangleq \frac{\partial \Phi^{\hat{\theta}}(x)}{\partial \hat{\theta}}$. The term Φ' can be evaluated as follows. Let $\hat{\eta}_p \in \mathbb{R}^{L_{p,0}}$ be defined as

$$\hat{\eta}_p = \begin{cases} x, & p = 1, \\ \hat{\eta}_{p-1} + \Phi_{p-1}^{\hat{\theta}_{p-1}}(\hat{\eta}_{p-1}), & p \in \{2, \ldots, m\}. \end{cases} \tag{10}$$

Then, it follows that $\Phi^{\hat{\theta}}(x) = \hat{\eta}_m + \Phi_m^{\hat{\theta}_m}(\hat{\eta}_m)$. To facilitate the subsequent development, the short-hand notations $\Phi'_p \triangleq \left(\frac{\partial \Phi^{\hat{\theta}}(x)}{\partial \hat{\theta}_p}\right)$, $\Lambda_p \triangleq \frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \hat{\theta}_p}$, $\Lambda_{p,j} \triangleq \frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \text{vec}(\hat{V}_{p,j})}$, and $\Xi_p \triangleq \frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \hat{\eta}_p}$ are introduced. Then $\Phi' = \left[\left(\frac{\partial \Phi^{\hat{\theta}}(x)}{\partial \hat{\theta}_1}\right), \ldots, \left(\frac{\partial \Phi^{\hat{\theta}}(x)}{\partial \hat{\theta}_m}\right)\right]$ can be expressed as

$$\Phi' \triangleq \left[\Phi'_1, \ \ldots, \ \Phi'_m\right]. \tag{11}$$

[5] Bias terms are omitted for simplicity of the notation.

[6] If f has different input-output dimensions, i.e., $f : \mathbb{R}^\mu \to \mathbb{R}^n$ with the input dimension μ, the ResNet architecture can be modified with an extra fully-connected DNN block $\Phi_0^{\theta_0}(x) \in \mathbb{R}^\mu$ at the input to account for the difference in input and output domains.

Using the chain rule, the term Φ'_p can be computed as

$$\Phi'_p = \left(\prod_{l=p+1}^{\curvearrowleft m} (I_n + \Xi_l) \right) \Lambda_p, \ \forall p \in \{1, \ldots, m\}. \tag{12}$$

In (12), the terms Λ_p and Ξ_p, for all $p \in \{1, \ldots, m\}$, can be computed as follows. Since $\hat{\theta}_p = \left[\text{vec}(\hat{V}_{p,0})^T, \ldots, \text{vec}(\hat{V}_{p,k_p})^T\right]^T$, it follows that $\frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \hat{\theta}_p} = \left[\left(\frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \text{vec}(\hat{V}_{p,0})}\right), \ldots, \left(\frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \text{vec}(\hat{V}_{p,k_p})}\right)\right]$. Therefore, using the definitions of Λ_p and $\Lambda_{p,j}$ yields

$$\Lambda_p = \begin{bmatrix} \Lambda_{p,0} & \Lambda_{p,1} & \ldots & \Lambda_{p,k_p} \end{bmatrix}, \forall p \in \{1, \ldots, m\}. \tag{13}$$

For brevity in the subsequent development, the short-hand notations $\hat{\varphi}_{p,j} \triangleq \varphi_{p,j}(\hat{\eta}_p, \hat{V}_{p,0}, \ldots, \hat{V}_{p,j})$ and $\hat{\varphi}'_{p,j} \triangleq \varphi'_{p,j}(\hat{\eta}_p, \hat{V}_{p,0}, \ldots, \hat{V}_{p,j})$ are introduced. Using (5), the chain rule, and the property of vectorization operators in (2), the terms $\Lambda_{p,0}$ and $\Lambda_{p,j}$ in (13) can be computed as

$$\Lambda_{p,0} = \left(\prod_{l=1}^{\curvearrowleft k_p} \hat{V}_{p,l}^T \hat{\varphi}'_{p,l} \right) (I_{L_{p,1}} \otimes \hat{\eta}_p^T), \tag{14}$$

and

$$\Lambda_{p,j} = \left(\prod_{l=j+1}^{\curvearrowleft k_p} \hat{V}_{p,l}^T \hat{\varphi}'_{p,l} \right) (I_{L_{p,j+1}} \otimes \hat{\varphi}_{p,j}^T), \tag{15}$$

for all $(p, j) \in \{1, \ldots, m\} \times \{1, \ldots, k_p\}$, respectively. Similarly, the term Ξ_p can be computed as

$$\Xi_p = \left(\prod_{l=1}^{\curvearrowleft k_p} \hat{V}_{p,l}^T \hat{\varphi}'_{p,l} \right) \hat{V}_{p,0}^T, \ \forall p \in \{1, \ldots, m\}. \tag{16}$$

Remark 2: If Φ_p suffers from the vanishing gradient problem, i.e., $\|\Xi_l\|_F \approx 0$ for all $l \in \{p+1, \ldots, m\}$, then $\Phi'_p = \left(\prod_{l=p+1}^{\curvearrowleft m} (I_n + \Xi_l) \right) \Lambda_p \approx \Lambda_p$. For an equivalent fully-connected DNN [12], i.e., in absence of shortcut connections, $\left\|\Phi'_p\right\|_F = \left\| \left(\prod_{l=p+1}^{\curvearrowleft m} \Xi_l \right) \Lambda_p \right\| \approx 0$. Thus, the shortcut connection circumvents the vanishing gradient problem in the ResNet when Φ_p has a vanishing gradient.

Based on the subsequent stability analysis, the control input is designed as

$$u \triangleq \dot{x}_d - \Phi^{\hat{\theta}}(x) - \sigma_e e - \sigma_s \text{sgn}(e), \tag{17}$$

where $\sigma_e, \sigma_s \in \mathbb{R}_{>0}$ are constant control gains, and $\text{sgn}(\cdot)$ denotes the vector signum function.

III. STABILITY ANALYSIS

To facilitate the subsequent analysis, let $z \triangleq [e^T, \tilde{\theta}^T]^T \in \mathbb{R}^\Psi$ denote a concatenated state, where $\Psi \triangleq n + \Sigma_{p=1}^m \Sigma_{j=0}^{k_p} L_{p,j} L_{p,j+1}$. Consider the candidate Lyapunov function $\mathcal{V}_L : \mathbb{R}^\Psi \to \mathbb{R}_{\geq 0}$ defined as

$$\mathcal{V}_L(z) \triangleq \frac{1}{2} e^T e + \frac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta}, \tag{18}$$

which satisfies the inequality $\alpha_1 \|z\|^2 \leq \mathcal{V}_L(z) \leq \alpha_2 \|z\|^2$, where $\alpha_1, \alpha_2 \in \mathbb{R}_{>0}$ are known constants. The universal function approximation property in (8) holds only on the compact domain Ω; hence, the subsequent stability analysis requires ensuring $x(t) \in \Omega$ for all $t \in \mathbb{R}_{\geq 0}$. This is achieved by yielding a stability result which constrains z in a compact domain. Consider the compact domain $\mathcal{D} \triangleq \{\varsigma \in \mathbb{R}^\Psi : \|\varsigma\| < \kappa\}$ in which z is supposed to lie, where $\kappa \in \mathbb{R}_{>0}$ is a bounding constant. The subsequent analysis shows that $z(t) \in \mathcal{D}$ for all $t \in \mathbb{R}_{\geq 0}$, if z is initialized within the set $\mathcal{S} \triangleq \{\varsigma \in \mathbb{R}^\Psi : \|\varsigma\| < \sqrt{\frac{\alpha_1}{\alpha_2}} \kappa\}$.

Taking the time-derivative of (4), substituting in (3) and (17), and substituting in (8) yields the closed-loop error system

$$\dot{e} = \Phi^{\theta^*}(x) - \Phi^{\hat{\theta}}(x) + \varepsilon(x) - \sigma_e e - \sigma_s \text{sgn}(e). \tag{19}$$

The ResNet in (6) is nonlinear in terms of the weights. Adaptive control design for nonlinearly parameterized systems is known to be a difficult problem [28]. A number of adaptive control methods have been developed to address the challenges posed by a nonlinear parameterization [3], [12], [28], [29], [30], [31], [32], [33]. In particular, first-order Taylor series approximation-based techniques have shown promising results for neural network-based adaptive controllers [3], [5], [12]. Specifically, the result in [12] uses a first-order Taylor series approximation to derive weight adaptation laws for a fully-connected DNN-based adaptive controller. Thus, motivation exists to explore a Taylor series approximation-based design to derive adaptation laws for the ResNet. For the ResNet in (6), a first-order Taylor series approximation-based error model is given by [27, Eq. 22]

$$\Phi^{\theta^*}(x) - \Phi^{\hat{\theta}}(x) = \Phi' \tilde{\theta} + \mathcal{O}\left(\left\| \tilde{\theta} \right\|^2 \right), \tag{20}$$

where $\mathcal{O}\left(\left\| \tilde{\theta} \right\|^2 \right)$ denotes higher-order terms. Since $\|x_d(t)\| \leq \overline{x_d}$ for all $t \in \mathbb{R}_{\geq 0}$, x can be bounded as $\|x\| \leq \|e + x_d\| \leq \|z\| + \|x_d\| \leq \kappa + \overline{x_d}$, based on the definition of $\mathcal{D}$, when $z \in \mathcal{D}$. Hence, since the ResNet is smooth, there exists a known constant $\overline{\Delta} \in \mathbb{R}_{>0}$ such that $\left\| \mathcal{O}\left(\left\| \tilde{\theta} \right\|^2 \right) \right\| \leq \overline{\Delta}$, when $z \in \mathcal{D}$. Then, substituting (20) into (19), the closed-loop error system can be expressed as

$$\dot{e} = \Phi' \tilde{\theta} + \mathcal{O}\left(\left\| \tilde{\theta} \right\|^2 \right) + \varepsilon(x) - \sigma_e e - \sigma_s \text{sgn}(e). \tag{21}$$

Then, using (9) and (21) yields

$$\dot{z} = h(z, t), \tag{22}$$

where $h : \mathbb{R}^{\Psi} \times \mathbb{R}_{\geq 0} \to \mathbb{R}^{\Psi}$ is defined as

$$h(z,t) \triangleq \begin{bmatrix} \left(\Phi'\tilde{\theta} + \mathcal{O}\left(\left\|\tilde{\theta}\right\|^2\right) + \varepsilon(x)\right) \\ -\sigma_e e - \sigma_s \text{sgn}(e) \\ -\Gamma \Phi'^T e \end{bmatrix}. \tag{23}$$

Based on the nonsmooth analysis technique in [34], the following theorem establishes the invariance properties of Filippov solutions to (22) and provides guarantees of asymptotic tracking error convergence for the system in (3).

Theorem 1: For the dynamical system in (3), the controller in (17) and the adaptation law in (9) ensure asymptotic tracking error convergence in the sense that $z, u, \hat{\theta} \in \mathcal{L}_\infty$ and $\lim_{t\to\infty} \|e(t)\| = 0$, provided Assumptions 1 and 2 hold, $z(0) \in \mathcal{S}$, and the following gain condition is satisfied:

$$\sigma_s > \bar{\varepsilon} + \bar{\Delta}. \tag{24}$$

Proof: Let $\partial \mathcal{V}_L$ denote the Clarke gradient of $\mathcal{V}_L$ defined in [35, p. 39]. Since $z \mapsto \mathcal{V}_L(z)$ is continuously differentiable,[7] $\partial \mathcal{V}_L(z) = \{\nabla \mathcal{V}_L(z)\}$, where ∇ denotes the standard gradient operator. Based on (23) and the chain rule in [37, Thm 2.2], it can be verified that $t \to \mathcal{V}_L(z(t))$ satisfies the differential inclusion

$$\begin{aligned} \dot{\mathcal{V}}_L &\overset{a.a.t.}{\in} \bigcap_{\xi \in \partial \mathcal{V}_L(z)} \xi^T K\left[h\right](z,t) \\ &= \nabla \mathcal{V}_L(z)^T K\left[h\right](z,t) \\ &= e^T\left(\mathcal{O}\left(\left\|\tilde{\theta}\right\|^2\right) + \varepsilon(x)\right) - \sigma_e \|e\|^2 \\ &\quad - \sigma_s e^T K\left[\text{sgn}\right](e) + e^T \Phi'\tilde{\theta} - \tilde{\theta}^T \Phi'^T e, \end{aligned} \tag{25}$$

for all $z \in \mathcal{D}$. Note that the terms $e^T\Phi'\tilde{\theta}$ and $\tilde{\theta}^T\Phi'^T e$ cancel. This cancellation of terms is the key motivation for designing the adaptive update law in (9). Using the fact that $e^T K\left[\text{sgn}\right](e) = \|e\|_1$, (25) can be bounded as

$$\dot{\mathcal{V}}_L \overset{a.a.t.}{\leq} -\sigma_e \|e\|^2 + e^T\left(\mathcal{O}\left(\left\|\tilde{\theta}\right\|^2\right) + \varepsilon(x)\right) - \sigma_s \|e\|_1, \tag{26}$$

for all $z \in \mathcal{D}$. Based on Holder's inequality, triangle inequality, and the fact that $\|e\| \leq \|e\|_1$, the following inequality can be obtained: $e^T\left(\mathcal{O}\left(\left\|\tilde{\theta}\right\|^2\right) + \varepsilon(x)\right) \leq \|e\|_1\left(\left\|\mathcal{O}\left(\left\|\tilde{\theta}\right\|^2\right)\right\| + \|\varepsilon(x)\|\right) \leq \left(\bar{\varepsilon} + \bar{\Delta}\right)\|e\|_1$. Then, provided the gain condition in (24) is satisfied, the right-hand side of (26) can be upper-bounded as

$$\dot{\mathcal{V}}_L \overset{a.a.t.}{\leq} -\sigma_e \|e\|^2, \tag{27}$$

for all $z \in \mathcal{D}$. Based on (27), invoking [34, Corollary 2] yields $z \in \mathcal{L}_\infty$ and $\lim_{t\to\infty} \|e(t)\| = 0$, when $z \in \mathcal{D}$. Using (27), $\alpha_1 \|z(t)\|^2 \leq \mathcal{V}_L(z(t)) \leq \mathcal{V}_L(z(0)) \leq \alpha_2 \|z(0)\|^2$, when $z(t) \in \mathcal{D}$. Thus, $\|z(t)\| < \sqrt{\frac{\alpha_2}{\alpha_1}}\|z(0)\|$, when $z(t) \in \mathcal{D}$. Therefore, $z(t) \in \mathcal{D}$ is always satisfied if $\|z(0)\| \leq \sqrt{\frac{\alpha_1}{\alpha_2}}\kappa$, i.e., $z(0) \in \mathcal{S}$. To show $x \in \Omega$ for ensuring the universal function approximation holds, consider the set $\Upsilon \subseteq \Omega$ defined as $\Upsilon \triangleq \{\zeta \in \Omega : \|\zeta\| \leq \kappa + \overline{x_d}\}$. Since $\|z\| \leq \kappa$ implies $\|e\| \leq \kappa$, the following relation holds: $\|x\| \leq \|e + x_d\| \leq \kappa + \overline{x_d}$. Therefore, $x(t) \in \Upsilon \subseteq \Omega$ for all $t \in \mathbb{R}_{\geq 0}$. Additionally, due to the facts that $(x, \hat{\theta}) \to \Phi^{\hat{\theta}}(x)$ is smooth, $x \in \Omega$, and $\hat{\theta} \in \mathcal{B}$, it follows that $\Phi^{\hat{\theta}}(x)$ is bounded. Since each term on the right-hand side of (17) is bounded, the control input $u \in \mathcal{L}_\infty$. Since $\phi_{p,j}$ and $\phi'_{p,j}$ are smooth for all $(p,j) \in \{1,\ldots,m\} \times \{0,\ldots,k_p\}$, it follows from (11)-(16) that Φ' is bounded. Then, every term on the right-hand side of (9) is bounded, and hence, $\dot{\hat{\theta}}$ is bounded. ■

Remark 3: If the ResNet is used to approximate the desired drift $f(x_d)$ instead of the actual drift $f(x)$, the control design and analysis method in our preliminary work in [21] can be used with the developed method to yield asymptotic tracking error convergence for any value of the initial condition $e(0) \in \mathbb{R}^n$.

Remark 4: If the sliding-mode term $\sigma_s \text{sgn}(e)$ is removed from the control input, the adaptation law in (9) can be modified with standard robust modification techniques such as sigma modification or e-modification [38, Ch. 8], where a uniformly ultimately bounded tracking result can be obtained without requiring knowledge of the bounds $\bar{\theta}$, $\bar{\Delta}$, and $\bar{\varepsilon}$. Note that the design using a sliding-mode term is not a key feature of our contribution, but only one approach we selected to develop the ResNet-based controller. As demonstrated in the last paragraph of the simulation section, the sliding-mode term can even be omitted while still achieving a good tracking performance.

Remark 5: The time and memory complexity of the approach is $\mathcal{O}(\Psi)$, i.e., growing linearly with the total number of parameters, similar to shallow NNs and fully-connected DNNs. Thus, the ResNet does not incur any significant additional computational cost compared to other architectures of the same size.

IV. SIMULATIONS

Monte Carlo simulations are provided[8] to demonstrate the performance of the developed ResNet-based adaptive controller, and the results are compared with a fully-connected DNN-based adaptive controller [12]. The system in (3) is considered with the state dimension $n = 10$. The unknown drift vector field in (3) is modeled as $f(x) = Ay(x)$, where $A \in \mathbb{R}^{n\times 6n}$ is a random matrix with all elements belonging to the uniform random distribution $U(0, 0.1)$, and $y(x) \triangleq [x^T, \tanh(x)^T, \sin(x)^T, \text{sech}(x)^T, (x \odot x)^T, (x \odot x \odot x)^T]^T$, where $\odot$ denotes the element-wise product operator. All elements of the initial state $x(0)$ are selected from the distribution $U(0, 2)$. The reference trajectory is

[7] For discontinuous candidate Lyapunov functions, the approach from [36] can be used in lieu of the continuous development in [37].

[8] Codes for the simulations are provided at https://github.com/patilomkarsudhir/Lyapunov-based-ResNet-Adaptive-Control/tree/main

selected as $x_d(t) = [0.5 + \sin(\omega_1 t), \ldots, 0.5 + \sin(\omega_n t)]$, where $\omega_1, \ldots, \omega_n \sim U(0, 20)$. The configuration of the ResNet in (6) is selected with 20 hidden layers, a shortcut connection across each hidden layer, and 10 neurons in each layer. The hyperbolic tangent activation function is used in each node of the ResNet. The results are compared with an equivalent fully-connected DNN-based adaptive control, i.e., the same configuration as the ResNet but without shortcut connections. The control and adaptation gains are selected as $\sigma_e = 2$, $\sigma_s = 2$, and $\Gamma = I_{\sum_{p=1}^{m}\sum_{j=0}^{kp} L_{p,j}L_{p,j+1}}$.

The performance of both the ResNet and the fully-connected DNN-based adaptive controller is sensitive to initial weights. To account for the sensitivity of performance to weight initialization, the initial weights for each method are obtained using a Monte Carlo method. In the Monte Carlo method, 10,000 simulations are performed, where the initial weights in each simulation are selected from $U(-0.05, 0.05)$, and the cost $J = \int_0^{10} \left(e^T(t)Qe(t) + u^T(t)Ru(t)\right) dt$ is evaluated in each simulation with $Q = I_{10}$ and $R = 0.01I_{10}$. For a fair comparison between the ResNet and the fully-connected DNN, the simulation results yielding the least J for each architecture are compared.[9]

TABLE 1. Performance comparison.

Architecture	$\|e_{\text{rms}}\|$	$\|\tilde{f}_{\text{rms}}\|$	$\|u_{\text{rms}}\|$
ResNet	0.329	3.395	24.332
ResNet (single shortcut connection)	0.375	5.727	23.836
Fully-Connected	0.912	9.636	24.816
Shallow NN with 10 neurons	0.727	6.187	24.681
Shallow NN with 100 neurons	0.560	4.979	24.662

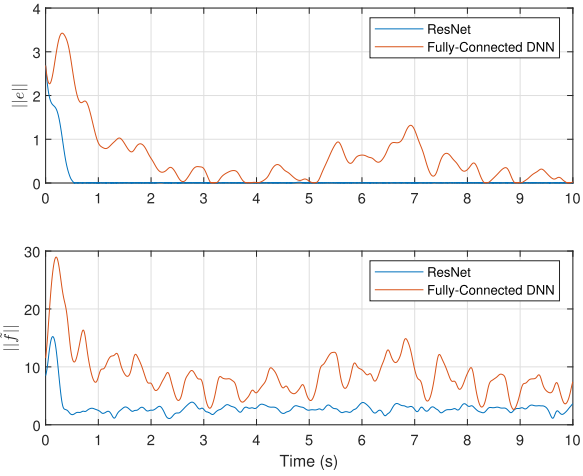

FIGURE 2. Plots of the tracking error norm $\|e\|$ and function approximation error norm $\|\tilde{f}\|$ with ResNet and fully-connected DNN-based adaptive controller.

Table 1 provides the norm of the root mean square (RMS) tracking error, function approximation error, and control input given by $\|e_{\text{rms}}\|$, $\|\tilde{f}_{\text{rms}}\|$, and $\|u_{\text{rms}}\|$, respectively. In comparison to the fully-connected DNN, the ResNet shows 63.93% and 64.77% decrease in the norms of the tracking and function approximation errors, respectively. As shown in Figure 2, the fully-connected DNN exhibits a comparatively poor tracking and function approximation performance. As mentioned in Remark 2, fully-connected DNNs suffer from the vanishing gradient problem. Thus, the fully-connected DNN weights remain approximately constant as shown in Figure 3. Consequently, the fully-connected DNN-based feedforward term fails to compensate for the uncertainty in the system which yields a relatively poor tracking and function approximation. In contrast to the fully-connected DNN, the presence of shortcut connections in the ResNet eliminates the vanishing gradient problem as mentioned in Remark 2. As a result, the ResNet weights are able to compensate for the system uncertainty as shown in Figure 3 which yields improved tracking and function approximation performance. Additionally, the ResNet requires approximately the same control effort as the fully-connected DNN. Therefore, the ResNet improves the tracking performance without requiring a higher control effort in comparison to the fully-connected DNN.

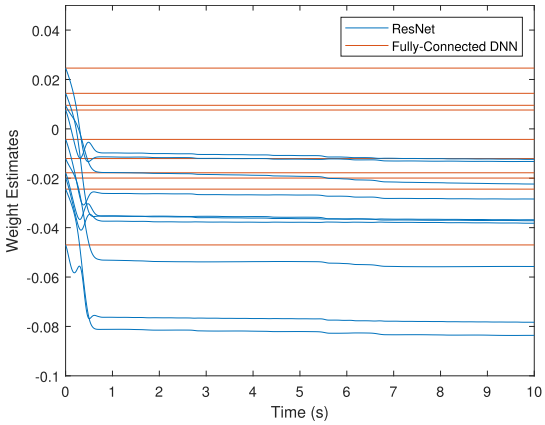

FIGURE 3. Plot of the weight estimates of the ResNet and fully-connected DNN. There are a total of 2,000 individual weights in each architecture. For better visualization, 10 arbitrarily selected weights are shown. The fully-connected DNN weights adapt slowly due to the problem of vanishing gradients. However, the ResNet weights are able to adapt faster since the ResNet does not have vanishing gradients.

Additionally, to demonstrate performance comparison of the ResNet with shallow NNs, comparative simulations two different configurations are used for the shallow NN architecture given by $V_1^T \phi(V_0^T x)$. In the first configuration, 10 neurons are used in the hidden layer of the shallow NN, i.e., the same number of neurons as in hidden layer of the ResNet. In the second configuration, we use 100 neurons in the hidden layer of the shallow NN, which yields the same total number of individual weights as the ResNet (i.e., 2000). As evident from Figure 4, the ResNet significantly outperforms both of the shallow NN configurations. Both of the shallow NN configurations exhibit overshoot in the tracking and function approximation errors, unlike the ResNet. The ResNet achieves rapid tracking error

[9]Although we use the Monte Carlo approach in this section to tune the initial weights, it can also be used to tune the controller gains using the same cost function.

convergence in approximately 0.5 seconds. In comparison, both of the shallow NN configurations fail to demonstrate exact tracking error convergence, despite containing the same robust state-feedback gains as the ResNet-based controller.

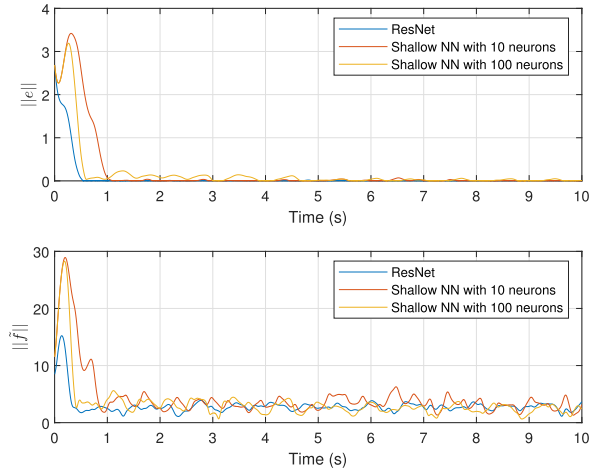

FIGURE 4. Comparative plots of the tracking error norm $\|e\|$ and the function approximation error norm $\left\|\tilde{f}\right\|$ with the ResNet and a shallow NN with 10 neurons.

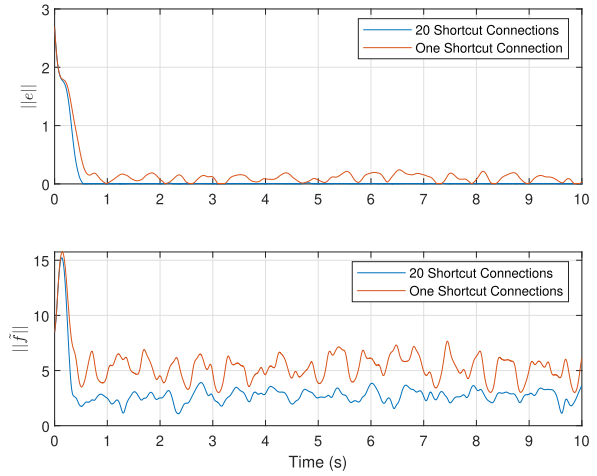

FIGURE 5. Comparative plots of the tracking error norm $\|e\|$ and the function approximation error norm $\left\|\tilde{f}\right\|$ with the proposed ResNet architecture (i.e., shortcut connection across every layer, 20 shortcut connections in total) and the ResNet with only one shortcut connection from our preliminary work in [21].

Additional simulation results are provided to examine the benefits of including more shortcut connections in the ResNet compared to our preliminary work in [21] which included only one shortcut connection. For the baseline method from [21], we consider a single shortcut connection beginning from the input layer and ending at the output layer of the ResNet. Figure 5 shows the comparative plots of the tracking error and function approximation error norms with respect to time, where the developed ResNet (i.e., with a shortcut connection across each layer, 20 shortcut connections in total) is found to provide improved tracking and function approximation performance. As evident from Table 1. the developed ResNet provided 12.27 % and 40.72% reduction in the RMS tracking and function approximation error norms, respectively, while requiring a comparable control effort. Such an improvement was expected because including more shortcut connections implies better mitigation of the vanishing gradient problem, as discussed in Remark 2.

Note that in practice, one can tune a ResNet to contain sufficiently large number of layers, neurons, and shortcut connections, which would yield a small value of the approximation error ε, which can be easily compensated using the sliding-mode term to yield asymptotic tracking. Additionally, to avoid using a sliding-mode term or the knowledge of the bound on θ^*, other standard robust modification techniques such as sigma modification or e-modification [38, Ch. 8] can be used in the adaptation law, where a uniformly ultimately bounded (UUB) tracking result can be obtained, as stated in Remark 4. The sliding-mode term is used to show an asymptotic tracking result, but it is not central to the main development. Figure 6 demonstrates the simulation results where the sliding-mode term $\sigma_s \mathrm{sgn}(e)$ is omitted and the e-modification based update law $\dot{\hat{\theta}} = -\sigma_\theta \|e\| \hat{\theta} + \Gamma \Phi'^T e$ is used with $\sigma_\theta = 1$ and $\sigma_e = 20$. As evident, the ResNet-based controller with the e-modification is able to track the desired trajectory with an ultimate bound of 0.15 on the tracking error, and root mean square (RMS) values of 0.269, 4.669, and 23.194 for the tracking error, function approximation error, and control input norms, respectively.

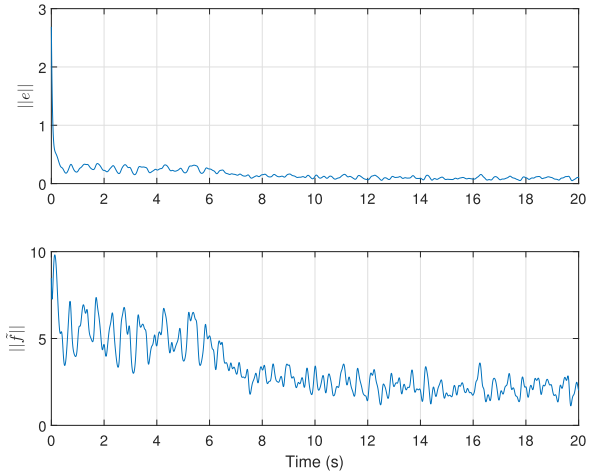

FIGURE 6. Plots demonstrating the tracking and function approximation errors with the ResNet-based controller, using sigma modification in the adaptation law.

V. CONCLUSION AND FUTURE WORK

This paper provided the first stability-driven adaptation laws for the weights of each layer of a ResNet-based adaptive controller with an arbitrary number of shortcut connections. Unlike fully-connected DNNs, the ResNet does

not exhibit the vanishing gradient problem. Comparative Monte Carlo simulations were provided where the developed ResNet-based adaptive controller provided approximately 64% improvement in the tracking and function approximation performance, in comparison to a fully-connected DNN-based adaptive controller with the same number of hidden layers and neurons.

Future work can explore incorporating a long short-term memory component in the ResNet architecture, based on our recent work [39], to model uncertainties with long-term temporal dependencies. Additionally, composite adaptive methods and physics-informed approaches (e.g., [40]) can be explored that incorporate a prediction error of the uncertainty, in addition to the tracking error, in the adaptation law.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

REFERENCES

[1] P. Kidger and T. Lyons, "Universal approximation with deep narrow networks," in *Proc. Conf. Learn. Theory*, Jan. 2020, pp. 2306–2327.

[2] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge, U.K.: Cambridge Univ. Press, 2019.

[3] F. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural net robot controller: Structure and stability proofs," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 388–399, Mar. 1996.

[4] F. L. Lewis, S. Jagannathan, and A. Yesildirak, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Philadelphia, PA, USA: CRC Press, 1998.

[5] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*. Boston, MA, USA: Kluwer Academic, 2002.

[6] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, "Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and RISE feedback control structure," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2180–2185, Oct. 2008.

[7] P. M. Patre, S. Bhasin, Z. D. Wilcox, and W. E. Dixon, "Composite adaptation for neural network-based controllers," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 944–950, Apr. 2010.

[8] 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 Syst. Lett.*, vol. 6, pp. 193–198, 2022.

[9] G. Joshi and G. Chowdhary, "Deep model reference adaptive control," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 4601–4608.

[10] G. Joshi, J. Virdi, and G. Chowdhary, "Asynchronous deep model reference adaptive control," in *Proc. Conf. Robot Learn.*, Jan. 2020.

[11] D. M. Le, M. L. Greene, W. A. Makumi, and W. E. Dixon, "Real-time modular deep neural network-based adaptive control of nonlinear systems," *IEEE Control Syst. Lett.*, vol. 6, pp. 476–481, 2022.

[12] O. S. Patil, D. M. Le, M. L. Greene, and W. E. Dixon, "Lyapunov-derived control and adaptive update laws for inner and outer layer weights of a deep neural network," *IEEE Control Syst. Lett.*, vol. 6, pp. 1855–1860, 2022.

[13] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[15] M. Hardt and T. Ma, "Identity matters in deep learning," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2017.

[16] K. Nar and S. Sastry, "Residual networks: Lyapunov stability and convex decomposition," 2018, *arXiv:1803.08203*.

[17] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2790–2798.

[18] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2018, pp. 527–542.

[19] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1181–1193, May 2019.

[20] T. Tan, Y. Qian, H. Hu, Y. Zhou, W. Ding, and K. Yu, "Adaptive very deep convolutional residual network for noise robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 26, no. 8, pp. 1393–1405, Aug. 2018.

[21] O. S. Patil, D. M. Le, E. J. Griffis, and W. E. Dixon, "Deep residual neural network (ResNet)-based adaptive control: A Lyapunov-based approach," in *Proc. IEEE 61st Conf. Decis. Control (CDC)*, Dec. 2022, pp. 3487–3492.

[22] A. F. Filippov, *Differential Equations With Discontinuous Right-Hand Sides*. Boston, MA, USA: Kluwer Academic, 1988.

[23] G. A. Leonov, M. A. Kiseleva, N. V. Kuznetsov, and O. A. Kuznetsova, "Discontinuous differential equations: Comparison of solution definitions and localization of hidden Chua attractors," *IFAC-PapersOnLine*, vol. 48, no. 11, pp. 408–413, 2015.

[24] D. S. Bernstein, *Matrix Mathematics*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[25] H. Lin and S. Jegelka, "ResNet with one-neuron hidden layers is a universal approximator," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[26] P. Tabuada and B. Gharesifard, "Universal approximation power of deep residual neural networks via nonlinear control theory," in *Proc. Int. Conf. Learn. Represent.*, May 2021.

[27] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 388–399, Mar. 1996.

[28] A. M. Annaswamy, F. P. Skantze, and A.-P. Loh, "Adaptive control of continuous time systems with convex/concave parametrization," *Automatica*, vol. 34, no. 1, pp. 33–49, Jan. 1998.

[29] A. Kojić, A. M. Annaswamy, A.-P. Loh, and R. Lozano, "Adaptive control of a class of nonlinear systems with convex/concave parameterization," *Syst. Control Lett.*, vol. 37, no. 5, pp. 267–274, Aug. 1999.

[30] W. Lin and C. Qian, "Adaptive control of nonlinearly parameterized systems: The smooth feedback case," *IEEE Trans. Autom. Control*, vol. 47, no. 8, pp. 1249–1266, Aug. 2002.

[31] W. Lin and C. Qian, "Adaptive control of nonlinearly parameterized systems: A nonsmooth feedback framework," *IEEE Trans. Autom. Control*, vol. 47, no. 5, pp. 757–774, May 2002.

[32] Z. Qu, R. A. Hull, and J. Wang, "Globally stabilizing adaptive control design for nonlinearly-parameterized systems," *IEEE Trans. Autom. Control*, vol. 51, no. 6, pp. 1073–1079, Jun. 2006.

[33] S. Basu Roy, S. Bhasin, and I. N. Kar, "Robust gradient-based adaptive control of nonlinearly parametrized plants," *IEEE Control Syst. Lett.*, vol. 1, no. 2, pp. 352–357, Oct. 2017.

[34] N. Fischer, R. Kamalapurkar, and W. E. Dixon, "LaSalle-Yoshizawa corollaries for nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2333–2338, Sep. 2013.

[35] F. H. Clarke, *Optimization and Nonsmooth Analysis*. Philadelphia, PA, USA: SIAM, 1990.

[36] V. A. Yakubovich, G. A. Leonov, and A. K. Gelig, *Stability of Stationary Sets in Control Systems With Discontinuous Nonlinearities*, vol. 14. Singapore: World Scientific, 2004.

[37] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 39, no. 9, pp. 1910–1914, Sep. 1994.

[38] P. Ioannou and J. Sun, *Robust Adaptive Control*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[39] E. J. Griffis, O. S. Patil, Z. I. Bell, and W. E. Dixon, "Lyapunov-based long short-term memory (Lb-LSTM) neural network-based control," *IEEE Control Syst. Lett.*, vol. 7, pp. 2976–2981, 2023.

[40] R. G. Hart, E. J. Griffis, O. Sudhir Patil, and W. E. Dixon, "Lyapunov-based physics-informed long short-term memory (LSTM) neural network-based adaptive control," *IEEE Control Syst. Lett.*, vol. 8, pp. 13–18, 2024.

OMKAR SUDHIR PATIL received the B.Tech. degree in production and industrial engineering from Indian Institute of Technology Delhi (IIT Delhi), in 2018, the M.S. degree in mechanical engineering, in 2022, and the Ph.D. degree in mechanical engineering from the University of Florida, Gainesville, FL, USA, in 2023.

In 2019, he joined the Nonlinear Controls and Robotics (NCR) Laboratory, University of Florida, under the guidance of Dr. Warren Dixon to pursue his Ph.D. studies. In 2023, he started working as a Postdoctoral Research Associate at the NCR Laboratory, University of Florida. His research interests include the development and application of innovative Lyapunov-based nonlinear, robust, and adaptive control techniques. He was honored with the BOSS Award for his outstanding bachelor's thesis project at IIT. During his Ph.D. studies, he was awarded the Graduate Student Research Award for outstanding research.

DUC M. LE received the Ph.D. degree in mechanical engineering from the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA, in 2022. During his graduate studies, his research was on nonlinear controls and autonomy, with a focus on adaptive control and switched system control in a variety of applications. He joined Aurora Flight Sciences, a Boeing Company, as an Aerospace Controls Researcher, in 2023. His research interests include, but are not limited to, adaptive control, deep learning, and guidance navigation and control (GNC).

EMILY J. GRIFFIS received the B.S. and M.S. degrees in mechanical engineering and the Ph.D. degree from the University of Florida, in May 2020, December 2021, and May 2024, respectively. In 2020, she joined the Nonlinear Controls and Robotics Laboratory, University of Florida, under the supervision of Dr. Warren Dixon to pursue her Ph.D. degree. Her research interests include using adaptive control and deep learning to study Lyapunov-based control of nonlinear and uncertain systems.

WARREN E. DIXON (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Clemson University, Clemson, SC, USA, in 2000. After working at the Oak Ridge National Laboratory as an Eugene P. Wigner Fellow and a Research Staff Member, he joined the University of Florida, in 2004, where he is currently the Dean's Leadership Professor and the Department Chair at the Department of Mechanical and Aerospace Engineering. His main research interests include the development and application of Lyapunov-based control techniques for uncertain nonlinear systems. He is an ASME and IEEE Fellow for contributions to adaptive control of uncertain nonlinear systems. His work has been acknowledged by various early and mid-career awards and best paper awards. His work has been acknowledged by various early and mid-career awards and best paper awards.