Concurrent Learning for System Identification and Control Using Lyapunov-Based Deep Neural Networks

Rebecca G. Hart, *Graduate Student Member, IEEE*, Omkar Sudhir Patil, Zachary I. Bell, and Warren E. Dixon, *Fellow, IEEE*

***Abstract*—This letter presents the first result that enables the use of a concurrent learning (CL) adaptation law for the weights of all the layers of the DNN-based controller applied to a class of second-order control-affine systems. The developed CL-based adaptation achieves convergence of the DNN's parameter estimates to a neighborhood of their ideal values, provided the DNN's Jacobian satisfies a finite excitation condition. A Lyapunov-based stability analysis is conducted to ensure convergence of the tracking error, weight estimation errors, and observer errors to a neighborhood of the origin. Simulations demonstrated approximately 40% improvement in function approximation performance and over 65% improvement on off-trajectory evaluations compared to a baseline without CL.**

***Index Terms*—Concurrent learning, system identification, nonlinear control systems.**

I. INTRODUCTION

DEEP Neural Networks (DNNs) have become increasingly popular due to their powerful function approximation capabilities [1], [2], [3], [4], [5], [6]. Yet, most DNN approximators rely on offline training with large datasets and do not adapt online. This requirement limits their ability to handle evolving dynamics in applications such as smart materials, fluid-structure iteration, and human-machine systems. Motivated by these issues, results such as [3], [4], [5], [6], [7], [8] (termed Lyapunov-based (Lb)-DNNs) have enabled online weight adaptation for all the weights in a DNN, demonstrating improved tracking performance over shallow NN approaches or traditional offline training methods. However, these approaches largely address tracking, while system identification, important for fault detection and robust state prediction under intermittent feedback [9], remains underexplored.

Motivated by the desire to achieve parameter convergence, results such as [2], [10], [11], [12], [13] use adaptive control techniques to yield convergence of parameter estimates to a neighborhood of the actual parameters. In [13], a least squares update law resulted in parameter estimation error convergence to a neighborhood of the origin, under the persistent excitation (PE) condition. However, the PE condition on the Jacobian cannot be verified online for nonlinear systems and requires continuous exploration of system states and parameter trajectories, posing a challenge in balancing exploration and exploitation. For systems involving linear-in-parameter (LIP) uncertainty, results such as [11], [12], [14], [15], [16] achieve parameter convergence without PE by incorporating data from previously explored trajectories into the parameter update laws. In [2], a DNN was cast in LIP form by adapting only the output layer weights online while keeping the inner layers fixed, enabling the use of concurrent learning (CL). CL techniques build a history stack of past data, improving system identification and allowing off-trajectory exploration [14], [15]. This approach reduces dependence on instantaneous measurements and broadens the information used in updates. Extending CL updates to all layers of a DNN remains challenging due to the nonlinear-in-parameter (NIP) nature of its inner layers, which presents significant technical challenges in formulating the prediction error to account for the nonlinear influence of the inner layers.

This letter presents the first approach for continuous all-layer adaptation of a DNN using a CL-based update law (Lb-CL-DNN). This method relaxes the PE condition, typical in other results, and ensures parameter convergence under finite excitation (FE). The novelty developed here is to enable the update law to dynamically reconstruct the history stack and to exploit the DNN Jacobian properties to maintain accurate parameter estimates. These innovations are imperative given that, unlike LIP regressions, the DNN's Jacobian is a function of the parameter estimates and thus transient errors would result in an inaccurate history stack. A Lyapunov-based stability analysis guarantees boundedness of the tracking error, weight estimation errors, and observer errors.

Received 12 September 2025; revised 19 November 2025; accepted 8 December 2025. Date of publication 18 December 2025; date of current version 31 December 2025. This work was supported in part by the National Science Foundation Graduate Research Fellowship under Grant DGE-2236414; in part by Air Force Research Lab (AFRL) under Grant FA8651-24-1-0018; and in part by Air Force Office of Scientific Research (AFOSR) under Grant FA9550-19-1-0169. Recommended by Senior Editor S. Baldi. *(Corresponding author: Rebecca G. Hart.)*

Rebecca G. Hart, Omkar Sudhir Patil, and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: rebecca.hart@ufl.edu; patilomkarsudhir@ufl.edu; wdixon@ufl.edu).

Zachary I. Bell is with the Munitions Directorate, Air Force Research Laboratory, Eglin Air Force Base, FL 32542 USA (e-mail: zachary.bell.10@us.af.mil).

Digital Object Identifier 10.1109/LCSYS.2025.3646130

Notation and Mathematical Background

The space of essentially bounded Lebesgue measurable functions is denoted by $\mathcal{L}_\infty$. Given $A \triangleq [a_{j,i}] \in \mathbb{R}^{n\times m}$, $\text{vec}(A) \triangleq [a_{1,1},\ldots,a_{n,1},\ldots,a_{1,m},\ldots,a_{n,m}]^\top$. The Kronecker product is denoted by $\otimes$. Given any $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{m\times p}$, and $C \in \mathbb{R}^{p\times r}$, $\text{vec}(ABC) = (C^\top \otimes A)\text{vec}(B)$. 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 $\overset{\curvearrowleft}{\prod}_{p=a}^{m} A_p = I_n$, if $a > m$. The identity matrix of size $n \times n$ is denoted by I_n. Given some functions f and g function composition is denoted by $\circ$, where $(f \circ g)(x) \triangleq f(g(x))$.

A. Deep Neural Network (DNN) Model

A feedforward DNN $\Phi(X,\theta) \in \mathbb{R}^n$ can be modeled as [4]

$$\Phi(X,\theta) \triangleq \left(v_k^\top \phi_k \circ \cdots \circ v_1^\top \phi_1\right)\left(v_0^\top X_a\right), \tag{1}$$

where $\theta \triangleq [\text{vec}(v_0)^\top,\ldots,\text{vec}(v_k)^\top]^\top \in \mathbb{R}^\rho$, where $\rho \triangleq \sum_{j=0}^{k} L_j L_{j+1}$, $j \in \{0,\ldots,k\}$ and $k \in \mathbb{N}$ denotes the number of hidden layers within θ, $v_j \in \mathbb{R}^{L_j \times L_{j+1}}$ denotes the matrix of weights and biases in the j^{th} hidden layer, $L_j \in \mathbb{N}$ denotes the number of nodes within the j^{th} hidden layer for all $j \in \{0,\ldots,k\}$, and $L_0 \triangleq m+1$, where m is the dimension of the $\mathbb{R}^m$ input vector to the DNN. Where $X \in \Omega$ denotes the input to the DNN, $\Omega \subset \mathbb{R}^m$ denotes a compact set, and the augmented input $X_a \in \mathbb{R}^{m+1}$ is defined as $X_a \triangleq \begin{bmatrix} X^\top & 1 \end{bmatrix}^\top$. The vector of smooth activation functions at the j^{th} layer is denoted by $\phi_j \in \mathbb{R}^{L_j}$ and is defined as $\phi_j \triangleq [\varsigma_{j,1} \ldots \varsigma_{j,L_j-1}\ 1]^\top$, where $\varsigma_{j,y} \in \mathbb{R}$ denotes the activation function at the y^{th} node of the j^{th} layer for all $j \in \{1,\ldots,k\}$. To incorporate a bias term into the DNN model in (1), the input X and the activation functions ϕ_j are augmented with a 1 for all $j \in \{1,\ldots,k\}$.

To facilitate the development of the online weight adaptation laws, the DNN model in (1) can also be represented recursively using the shorthand notation Φ_j as [4]

$$\Phi_j \triangleq \begin{cases} v_j^\top \phi_j(\Phi_{j-1}), & j \in \{1,\ldots,k\}, \\ v_0^\top X_a & j = 0, \end{cases}$$

where $\Phi(X,\theta) = \Phi_k$.

The Jacobian of the feedforward DNN, denoted $\Phi'(X,\theta)$, can be represented as $\Phi'(X,\theta) \triangleq [\Phi_0'(X,\theta),\ldots,\Phi_j'(X,\theta)] \in \mathbb{R}^{n\times\rho}$, where $\Phi_j' \triangleq \frac{\partial \Phi(X,\theta)}{\text{vec}(v_j)} \in \mathbb{R}^{n\times L_j L_{j+1}}$, for all $j \in \{0,\ldots,k\}$. The terms Φ_0' and Φ_j' can be expressed as [4]

$$\Phi_0'(X,\theta) \triangleq \left(\overset{\curvearrowleft}{\prod_{l=1}^{k}} v_l^\top \phi_l'\right)\left(I_{L_1} \otimes X_a^\top\right),$$

$$\Phi_j'(X,\theta) \triangleq \left(\overset{\curvearrowleft}{\prod_{l=j+1}^{k}} v_l^\top \phi_l'\right)\left(I_{L_{j+1}} \otimes \phi_j^\top\right),$$

for all $j \in \{1,\ldots,k\}$, where the activation function at the j^{th} layer and its Jacobian are expressed using the shorthand notations $\phi_j \triangleq \phi_j(\Phi_{j-1}(X,\theta))$ and $\phi_j' \triangleq \phi_j'(\Phi_{j-1}(X,\theta))$, respectively, and $\phi_j' : \mathbb{R}^{L_j} \to \mathbb{R}^{L_j\times L_j}$ is defined as $\phi_j'(y) \triangleq \frac{\partial}{\partial \varrho}\phi_j(\varrho)|_{\varrho=y}$, for all $y \in \mathbb{R}^{L_j}$. For each $j \in \{0,\ldots,k\}$, the activation function ϕ_j, its Jacobian ϕ_j', and Hessian $\phi_j''(y) \triangleq \frac{\partial^2}{\partial y^2}\phi_j(y)$ are bounded as $\|\phi_j(y)\| \le \mathfrak{a}_1\|y\| + \mathfrak{a}_0$, $\|\phi_j'(y)\| \le \mathfrak{b}_0$, $\|\phi_j''(y)\| \le \mathfrak{c}_0$, where $\mathfrak{a}_0, \mathfrak{a}_1, \mathfrak{b}_0, \mathfrak{c}_0 \in \mathbb{R}_{\ge 0}$ are known constants.[1]

Let the DNN parameters which yield the best approximation of an unknown function $f(x,\dot{x})$ be defined using the loss function $\mathcal{L} : \mathbb{R}^\kappa \to \mathbb{R}_{\ge 0}$ given as

$$\mathcal{L}(\theta) \triangleq \int_\Omega \left(\|f(x,\dot{x}) - \Phi(X,\theta)\|^2 + \sigma\|\theta\|^2\right) d\mu(x_i),$$

where μ denotes the Lebesgue measure, $\sigma \in \mathbb{R}_{>0}$ denotes a regularizing constant, and the term $\sigma\|\theta\|^2$ represents L_2 regularization, also known as ridge regression [17, Sec. 7.1.1]. A bounded parameter search space $\Theta \triangleq \{\theta \in \mathbb{R}^\rho : \|\theta\| \le \bar{\theta}\}$ is considered where $\bar{\theta} \in \mathbb{R}_{>0}$ is a user-selected bound. Therefore, the desired DNN parameter vector $\theta^* \in \Theta$ is formally defined as

$$\theta^* \triangleq \underset{\theta\in\Theta}{\arg\min}\ \mathcal{L}(\theta). \tag{2}$$

For the solution of (2) to be unique we make the following assumption.

Assumption 1: The loss function $\mathcal{L}$ is strictly convex over the set Θ.

II. Control Design

Consider a control-affine[2] nonlinear dynamic system modeled as

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

where $x, \dot{x} \in \mathbb{R}^n$ denotes the available generalized position and velocity, $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes an unknown continuously differentiable function, and $u \in \mathbb{R}^n$ denotes a control input. The tracking control objective is to simultaneously track a user-defined reference trajectory $x_d \in \mathbb{R}^n$ and learn the unknown function $f(x,\dot{x})$ online using a DNN starting from the initial time t_0. Therefore, $f(x,\dot{x})$ can be modeled as

$$f(x,\dot{x}) = \Phi\left(X,\theta^*\right) + \varepsilon(X), \tag{4}$$

where the input to the DNN is denoted $X \triangleq [x^\top, \dot{x}^\top]^\top$. Due to the continuity of $f(x,\dot{x})$ and Φ and the constraint of a bounded search space, the function approximation error ε which bounds $\sup_{x\in\Omega}\|f(x,\dot{x}) - \Phi(X,\theta)\|$ exists but may not be prescribable as stated in the universal function approximation theorem [18, Th. 3.1].

The reference trajectory is assumed to satisfy $\|x_d\| \le \overline{x_d}$, $\|\dot{x}_d\| \le \overline{\dot{x}_d}$, and $\|\ddot{x}_d\| \le \overline{\ddot{x}_d}$, where $\overline{x_d}, \overline{\dot{x}_d}, \overline{\ddot{x}_d} \in \mathbb{R}_{>0}$ are known constants. To quantify the tracking objective, the trajectory and auxiliary tracking errors $e, r \in \mathbb{R}^n$ are defined as

$$e \triangleq x - x_d, \quad r \triangleq \dot{e} + \alpha_1 e, \tag{5}$$

[1] Most activation functions, e.g., sigmoidal, and rectified linear unit (ReLUs), satisfy the form of these bounds.

[2] A second-order, control-affine system with matched uncertainties is intentionally considered to narrowly focus on the primary contribution. We believe future work could extend this method to address more general systems, such as those with unmatched uncertainties, by incorporating classical techniques, as no fundamental technical barriers are foreseen.

where $\alpha_1 \in \mathbb{R}_{>0}$ is a user-selected control gain. Taking the time derivative of (5), applying (3), and (4) yields the open-loop error system

$$\dot{r} = \Phi\big(X, \theta^*\big) + \varepsilon(X) + u - \ddot{x}_d + \alpha_1 \dot{e}. \tag{6}$$

The subsequent development considers the concurrent tracking control and system identification problem. The parameter identification objective is quantified using the parameter estimation error defined as $\tilde{\theta} \triangleq \theta^* - \hat{\theta}$, where $\hat{\theta} \in \mathbb{R}^{\rho}$ represents the parameter estimates. Using the definitions of $\Phi(x, \theta^*)$ and $\Phi(x, \hat{\theta})$, a first-order Taylor series approximation-based model of the estimation error is used to obtain [4]

$$\Phi\big(x, \theta^*\big) - \Phi\Big(x, \hat{\theta}\Big) = \Phi'\Big(x, \hat{\theta}\Big)\tilde{\theta} + R\Big(x, \tilde{\theta}\Big), \tag{7}$$

where $R(x, \tilde{\theta})$ represents the Lagrange remainder term.

A. Closed-Loop Error System and Control Law Development

Since $f(x, \dot{x})$ in (6) is unknown, we are motivated to develop an approximation that can be used as a feedforward control element. Based on the subsequent stability analysis, the control input is designed as

$$u = \ddot{x}_d - \Phi\Big(X, \hat{\theta}\Big) - k_1 r - e - \alpha_1 \dot{e}, \tag{8}$$

where $k_1 \in \mathbb{R}_{>0}$ is a user-defined control gain. Substituting (4) and (8) into (6), applying (7), and canceling cross terms yields

$$\dot{r} = \Phi'\Big(X, \hat{\theta}\Big)\tilde{\theta} + R\Big(x, \tilde{\theta}\Big) + \varepsilon(X) - k_1 r - e. \tag{9}$$

B. Dynamic State-Derivative Observer

Motivated by the desire to incorporate previous state information into the update law in a CL style, the update law is augmented with a history stack containing the error between the calculated control input and a reconstructed version of the control input. To reconstruct (3), an observer is developed to estimate the unmeasurable state $\ddot{x}$. The dynamic state-derivative observer is designed as

$$\dot{\hat{r}} = \hat{\Delta} - \ddot{x}_d + \alpha_1 \dot{e} + \alpha_2 \tilde{r}, \tag{10}$$
$$\dot{\hat{\Delta}} = \tilde{r} + k_\Delta \tilde{\Delta} - \dot{u}, \tag{11}$$

where $\hat{r}, \hat{\Delta} \in \mathbb{R}^n$ denote the observer estimates for r and $\ddot{x}$, respectively, $\tilde{r}, \tilde{\Delta} \in \mathbb{R}^n$ denote the observer errors, $\tilde{r} = r - \hat{r}$ and $\tilde{\Delta} = \ddot{x} - \hat{\Delta}$, respectively $\alpha_2, k_\Delta \in \mathbb{R}_{>0}$ denote constant observer gains, and $\dot{u}$ is the unmeasurable derivative of the control input. Taking the time derivative of $\tilde{r}$ and $\tilde{\Delta}$ and applying (10) and (11) yields

$$\dot{\tilde{r}} = \tilde{\Delta} - \alpha_2 \tilde{r}, \tag{12}$$
$$\dot{\tilde{\Delta}} = \dot{f}(x, \dot{x}) + \dot{u} - \tilde{r} - k_\Delta \tilde{\Delta} - \dot{u}, \tag{13}$$

where $\dot{f}(x, \dot{x}) \triangleq \frac{d}{dt} f(x, \dot{x}) = \frac{\partial f}{\partial x}\dot{x} + \frac{\partial f}{\partial \dot{x}}\ddot{x}$ and $\dot{u} \triangleq \frac{d}{dt}u$. The observer error $\tilde{r}$ is known because r and $\hat{r}$ are known. Because $\tilde{\Delta} = \dot{\tilde{r}} + \alpha_2 \tilde{r}$, the implementable form of (11) can be obtained by integrating on both sides and using the relation $\int_{t_0}^{t} \dot{\tilde{r}} = \tilde{r}(t) - \tilde{r}(t_0)$, which yields $\hat{\Delta}(t) = \hat{\Delta}(t_0) + k_\Delta(\tilde{r}(t) - \tilde{r}(t_0)) - (u(t) - u(t_0)) + \int_{t_0}^{t} (k_\Delta \alpha_2 + 1)\tilde{r}(\tau) d\tau$.

Let the concatenated error vectors $z \in \mathbb{R}^{2n}$ and $\zeta \in \mathbb{R}^{\psi}$ be defined as $z \triangleq [\tilde{r}^\top, \tilde{\Delta}^\top]^\top$ and $\zeta \triangleq [r^\top, e^\top, \tilde{\theta}^\top]^\top$, respectively, where $\psi \triangleq 2n+\rho$. Additionally, let the open and connected set $\mathcal{D} \subset \mathbb{R}^{\psi}$ be defined as $\mathcal{D} \triangleq \{\zeta \in \mathbb{R}^{\psi} : \|\zeta\| < \chi\}$, where $\chi \in \mathbb{R}_{>0}$ denotes a subsequently defined known upper bound. The following lemma establishes a bound on $\dot{f}(x, \dot{x})$ when $\zeta \in \mathcal{D}$ to facilitate the convergence analysis for the observer.

Lemma 1: For all $\zeta \in \mathcal{D}$, there exists a constant $\delta_f \in \mathbb{R}_{>0}$ such that the bound $\|\dot{f}(x, \dot{x})\| \leq \delta_f$ holds.

Proof: See Appendix. ■

Let $\Lambda_1 \triangleq \min\{k_\Delta, 2\alpha_2\}$. The following lemma establishes the convergence properties of the observer error system in (12) and (13).[3]

Lemma 2: Consider the observer given by (10) and (11). The observer error is bounded in the sense that $\|z(t)\| \leq \sqrt{\|z(t_0)\|^2 e^{-\Lambda_1(t-t_0)} + \frac{\delta_f^2}{\Lambda_1}(1 - e^{-\Lambda_1(t-t_0)})}$ for all $t \in \mathbb{R}_{\geq 0}$, provided $\zeta \in \mathcal{D}$. Furthermore, suppose the initial acceleration error is bounded by a known constant, $\|\tilde{\Delta}(t_0)\| \leq \bar{\Delta}_0$ where $\bar{\Delta}_0 \in \mathbb{R}_{>0}$. The observer can achieve a prescribed accuracy $\delta_\Delta \in \mathbb{R}_{>0}$ with the settling time $t_\Delta \triangleq t_0 + \frac{1}{\Lambda_1}\ln(\frac{k_\Delta \Lambda_1 \bar{\Delta}_0^2 - \delta_f^2}{k_\Delta \Lambda_1 \delta_\Delta^2 - \delta_f^2})$, provided the feasibility gain condition $k_\Delta \Lambda_1 > \frac{\delta_f^2}{\delta_\Delta^2}$ is satisfied.

Proof: See Appendix. ■

Remark 1: Using Lemma 2, the observer errors, $\tilde{\Delta}$ and $\tilde{r}$ will converge to the prescribed ultimate bound δ_Δ after the time t_Δ has passed.

III. CL Adaptation Laws For Adaptive Control

The adaptation law is developed by extending established CL techniques to the NIP structure of the DNN. The shorthand notation $\Phi'(X, \hat{\theta})$ in (14) denotes the Jacobian of the DNN at the current state and weight estimates, and $\Phi'(X_i, \hat{\theta})$ represents the Jacobian of the DNN using the current weight estimates and the previous state X_i, where $X_i \triangleq X(t_i)$ and $t_i \in [t_\Delta, t]$, represents states within this time interval. Compared to other CL approaches which begin gathering data immediately, the history stack is only constructed after $t \geq t_\Delta$ where the observer settling time t_Δ is defined in Lemma 2. This time-based condition ensures that the user-defined number of data points $N \in \mathbb{N}$ which construct the history stack are gathered only after the observer has reached its ultimate bound which is required to ensure accuracy of the stored data. The implementable form of the CL-DNN update law is designed as

$$\dot{\hat{\theta}} = \mathrm{proj}\Bigg(\Gamma\Bigg(\Phi'^\top\Big(X, \hat{\theta}\Big)r - \gamma_1 \sum_{i=1}^{N} \Phi'^\top\Big(X_i, \hat{\theta}\Big)\big(u_i - \hat{u}_i\big) - \gamma_2 \hat{\theta}\Bigg)\Bigg), \tag{14}$$

where $\gamma_1, \gamma_2 \in \mathbb{R}_{>0}$ denote user-selected adaptation gains, and $\Gamma \in \mathbb{R}^{\rho \times \rho}$ denotes a positive-definite time-varying least

[3] When the observer is activated with the system at rest, the initial velocity and acceleration are both zero, simplifying the calculation and eliminating the need for an upper bound. If the initial acceleration is not exactly known, a conservative upperbound can be used and will result in a longer settling time t_Δ.

squares adaptation gain matrix

$$\frac{d}{dt}\Gamma^{-1} = \begin{cases} -\beta\Gamma^{-1} + \gamma_1\left(\sum_{i=1}^{N} \Phi'^{\top}\left(X_i, \hat{\theta}\right)\Phi'\left(X_i, \hat{\theta}\right)\right), \\ \quad \text{if } \lambda_{\Gamma,\min} < \lambda_{\min}(\Gamma) \text{ and } \lambda_{\max}(\Gamma) < \lambda_{\Gamma,\max} \\ 0, \qquad \text{otherwise} \end{cases} \tag{15}$$

where β represents a user selected forgetting factor $\beta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, and $\lambda_{\Gamma,\min}$, $\lambda_{\Gamma,\max}$ are user-selected bounds for the minimum and maximum eigenvalues of Γ, respectively. The adaptation gain in (15) is initialized to be PD and it can be shown that $\Gamma(t)$ remains PD for all $t \in \mathbb{R}_{\geq 0}$ [19]. A reconstructed estimate of the calculated control input can be determined as

$$\hat{u}_i = \hat{\Delta}_i - \Phi\left(X_i, \hat{\theta}\right). \tag{16}$$

An analytical form of the implementable weight adaptation law in (14) can be obtained from (3), (4), (7), (14), and (16) as

$$\dot{\hat{\theta}} = \text{proj}\Bigg(\Gamma\Bigg(\Phi'^{\top}\left(X, \hat{\theta}\right)r - \gamma_2\hat{\theta} + \gamma_1\sum_{i=1}^{N}\Phi'^{\top}\left(X_i, \hat{\theta}\right)\Phi'\left(X_i, \hat{\theta}\right)\tilde{\theta} + \gamma_1\sum_{i=1}^{N}\Phi'^{\top}\left(X_i, \hat{\theta}\right)\left(R\left(x, \tilde{\theta}\right) + \varepsilon(X_i) - \tilde{\Delta}_i\right)\Bigg)\Bigg). \tag{17}$$

IV. Stability Analysis

Using the structure of the DNN, the following bounds can be established

$$\left\|R\left(X, \tilde{\theta}\right)\right\| \leq \rho_1(\|\zeta\|)\|\zeta\|^2, \quad \left\|\widehat{\Phi}'\left(X_i, \hat{\theta}\right)\right\| \leq \rho_2(\|X_i\|), \tag{18}$$

where $\rho_1, \rho_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denote strictly-increasing functions. Due to the fact that $R(X, \tilde{\theta})$, $\Phi'(X_i, \hat{\theta})$, and ε are bounded, then the following bounds can be established

$$\left\|\Phi'^{\top}\left(X_i, \hat{\theta}\right)R\left(X_i, \tilde{\theta}\right)\right\| \leq \rho_3(\|\zeta_i\|)\|\zeta\|^2, \quad \left\|\Phi'^{\top}\left(X_i, \hat{\theta}\right)\varepsilon(X_i)\right\| \leq \rho_4(\|X_i\|), \tag{19}$$

where $\rho_3, \rho_4 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denote strictly-increasing functions. Let $\rho_\Delta(\|\zeta\|) \geq \frac{1}{2k_1}\rho_1^2(\|\zeta\|)\|\zeta\|^2 + \frac{\gamma_1^2}{2}\sum_{i=1}^{N}\rho_3^2(\|\zeta_i\|)\|\zeta\|^2$, where $\rho_\Delta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denotes an invertible strictly-increasing function. Since the approximation capabilities of DNNs holds on a compact domain Ω, the subsequent stability analysis requires ensuring $X(t) \in \Omega$ for all $t \in [t_0, \infty)$. This is achieved by yielding a stability result which constrains ζ in a compact domain. Therefore, consider the compact domain $\mathcal{D} \triangleq \{\sigma \in \mathbb{R}^{\psi} : \|\sigma\| \leq \chi\}$ in which ζ is supposed to lie to develop Theorem 1. It follows that if $\|\zeta\| \leq \chi$ then X can be bounded as $\|X\| \leq (\alpha+2)\chi + \overline{x_d} + \overline{\dot{x}_d}$. Therefore, select $\Omega \triangleq \{\sigma \in \mathbb{R}^{2n} : \|\sigma\| \leq (\alpha_1+2)\chi + \overline{x_d} + \overline{\dot{x}_d}\}$. Then $\zeta \in \mathcal{D}$ implies $X \in \Omega$. Let $\lambda \triangleq \min(\frac{k_1}{2} - \frac{1}{2}, \alpha_1, \frac{\gamma_1}{2}(\lambda_{\min}(\sum_{i=1}^{N}\Phi'^{\top}(X_i, \hat{\theta})\Phi'(X_i, \hat{\theta})) - 1 - N\delta_\Delta) - \frac{1}{2} + \frac{\gamma_2}{2})$ and $\lambda_d \in \mathbb{R}_{>0}$ denote the desired convergence rate. To ensure that arbitrary initial conditions are always included, the user-selected constant χ is selected as $\chi \triangleq \rho_\Delta^{-1}(\lambda - \lambda_d)$. Then, it follows that $z(t_0) \in \mathcal{D} = \{\zeta \in \mathbb{R}^{2n+p} : \|\zeta\| \leq \chi\}$ is always satisfied. Because the solution $t \mapsto \zeta(t)$ is continuous due to the Caratheodory existence conditions [20, Ch.2, Th. 1.1], there exists a time-interval $\mathcal{I} \triangleq [t_0, t_1)$ such that $\|\zeta(t)\| < \mathcal{D}$ for all $t \in \mathcal{I}$. It follows that $X(t) \in \Omega$ for all $t \in \mathcal{I}$, therefore the universal function approximation property holds over this time interval. In the subsequent stability analysis, we analyze the convergence properties of the solutions and also establish that $\mathcal{I}$ can be extended to $[t_0, \infty)$. Let the set $\mathcal{S} \subset \mathcal{D}$ be the set of stabilizing initial conditions defined as

$$\mathcal{S} \triangleq \left\{\sigma \in \mathbb{R}^{\psi} : \|\zeta(t_0)\| < \sqrt{\frac{\beta_1}{\beta_2}\rho_\Delta^{-1}(\lambda - \lambda_d)^2 - \frac{\iota}{\lambda_d}}\right\},$$

where $\beta_1 \triangleq \min\{\frac{1}{2}, \frac{1}{2}\frac{1}{\lambda_{\Gamma,\max}}\}$, $\beta_2 \triangleq \max\{\frac{1}{2}, \frac{1}{2}\frac{1}{\lambda_{\Gamma,\min}}\}$, and $\iota \triangleq \frac{1}{2}\bar{\varepsilon}^2 + \frac{\gamma_2}{2}\bar{\theta}^2 + \frac{\gamma_1}{2}\sum_{i=1}^{N}\rho_4^2(\|X_i\|) + \frac{\gamma_1 N\delta_\Delta}{2}\sum_{i=1}^{N}\rho_2^2(\|X_i\|)$.

To guarantee that the adaptive update law in Theorem 1 achieves the parameter identification, i.e., $\|\tilde{\theta}\|$ is minimized, the history stack needs to be sufficiently rich, i.e., the system needs to be excited over a finite duration of time as specified in the following assumption.

Assumption 2: There exists $\lambda_e > 0$ and there exists $T > t_\Delta$ for all $t \geq T$ such that $\lambda_{\min}\{\sum_{i=1}^{N}\Phi'^{\top}(X_i, \hat{\theta})\Phi'(X_i, \hat{\theta})\} \geq \lambda_e$.

Consider the Lyapunov candidate function $\mathcal{V} : \mathbb{R}^{\psi} \to \mathbb{R}_{\geq 0}$ defined as

$$\mathcal{V}(\zeta) = \frac{1}{2}r^{\top}r + \frac{1}{2}e^{\top}e + \frac{1}{2}\tilde{\theta}^{\top}\Gamma^{-1}\tilde{\theta}, \tag{20}$$

which satisfies the inequality $\beta_1\|\zeta\|^2 \leq \mathcal{V}(\zeta) \leq \beta_2\|\zeta\|^2$. Taking the time derivative of $\mathcal{V}(\zeta)$, and applying (5) and (9) yields

$$\dot{\mathcal{V}} = r^{\top}\left(\Phi'\left(X, \hat{\theta}\right)\tilde{\theta} + R\left(X, \tilde{\theta}\right) + \varepsilon(X) - k_1 r - e\right) - e^{\top}\alpha_1 e + \tilde{\theta}^{\top}\Gamma^{-1}\dot{\tilde{\theta}} + \frac{1}{2}\tilde{\theta}^{\top}\left(\frac{d}{dt}\Gamma^{-1}\right)\tilde{\theta}. \tag{21}$$

Using (15), the last term in (21) can be bounded as

$$\frac{1}{2}\tilde{\theta}^{\top}\left(\frac{d}{dt}\Gamma^{-1}\right)\tilde{\theta} \leq \frac{1}{2}\tilde{\theta}^{\top}\gamma_1\left(\sum_{i=1}^{N}\Phi'^{\top}\left(X_i, \hat{\theta}\right)\Phi'\left(X_i, \hat{\theta}\right)\right)\tilde{\theta}. \tag{22}$$

Theorem 1 provides convergence guarantees for the tracking and parameter estimation errors using the update law in (17).

Theorem 1: Let the gain conditions $k_\Delta\Lambda_1 > \frac{\delta_f^2}{\delta_\Delta^2}$ and $\lambda_d > 0$ be satisfied, and $\|\zeta(0)\| \in \mathcal{S}$. For the dynamical system in (3), the controller in (8) and the adaptation law developed in (17) ensures the concatenated error vector ζ is bounded in the sense that $\|\zeta(t)\| \leq \sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 e^{-\frac{\lambda_d}{\beta_2}(t-t_0)} + \frac{\beta_2\iota}{\beta_1\lambda_d}(1 - e^{-\frac{\lambda_d}{\beta_2}(t-t_0)})}$, for all $t \in \mathbb{R}_{\geq 0}$.

Proof: Consider the candidate Lyapunov function in (20). From (17), (21), and using (22), $\dot{\mathcal{V}}$ can be upper bounded as

$$\dot{\mathcal{V}} \leq -r^{\top}k_1 r - e^{\top}\alpha_1 e + r^{\top}\left(R\left(X, \tilde{\theta}\right) + \varepsilon(X)\right) - \tilde{\theta}^{\top}\frac{\gamma_1}{2}\sum_{i=1}^{N}\Phi'^{\top}\left(X_i, \hat{\theta}\right)\Phi'\left(X_i, \hat{\theta}\right)\tilde{\theta} - \tilde{\theta}^{\top}\gamma_2\tilde{\theta} + \tilde{\theta}^{\top}\gamma_2\theta^* - \tilde{\theta}^{\top}\gamma_1\sum_{i=1}^{N}\Phi'^{\top}\left(X_i, \hat{\theta}\right)\left(R\left(X_i, \tilde{\theta}\right) + \varepsilon(X_i) - \tilde{\Delta}_i\right).$$

Using Lemma 2, the bound $\|\tilde{\Delta}_i\| \leq \delta_\Delta$ holds for all $i \in \{1, \ldots, N\}$, provided $\zeta \in \mathcal{D}$, the data is collected after the settling time t_Δ, and the feasibility gain condition $k_\Delta \Lambda_1 > \frac{\delta_f^2}{\delta_\Delta^2}$ is satisfied. Therefore, using the bounds in (18) and (19), $\dot{\mathcal{V}}$ can be upper bounded as $\dot{\mathcal{V}} \leq -(\lambda - \rho_\Delta(\|\zeta\|))\|\zeta\|^2 + \iota$, when $\zeta \in \mathcal{D}$. Therefore, when ζ is initialized such that $\zeta(t_0) \in \mathcal{S}$, then, from the definition of $\mathcal{S}$, $\lambda > \lambda_d + \rho_\Delta(\sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2 \iota}{\beta_1 \lambda_d}})$. Recall that, because the solution $t \mapsto \zeta(t)$ is continuous, ζ cannot instantaneously escape $\mathcal{S}$ at t_0, therefore there exists a time-interval $\mathcal{I}$ such that $\zeta(t) \in \mathcal{S}$ for all $t \in \mathcal{I}$, implying $\|\zeta(t)\| < \sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2 \iota}{\beta_1 \lambda_d}}$ for all $t \in \mathcal{I}$. Because ρ_Δ is strictly increasing, $\rho_\Delta(\|\zeta(t)\|) < \rho_\Delta(\sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2 \iota}{\beta_1 \lambda_d}})$ for all $t \in \mathcal{I}$. As a result,

$$\dot{\mathcal{V}} \leq -\lambda_d \|\zeta\|^2 + \iota \tag{23}$$

for all $t \in \mathcal{I}$. Using $\beta_1\|\zeta\|^2 \leq \mathcal{V}(\zeta) \leq \beta_2\|\zeta\|^2$, $\dot{\mathcal{V}}$, and solving the differential inequality over $\mathcal{I}$ yields $\mathcal{V}(\zeta(t)) \leq \mathcal{V}(\zeta(t_0))e^{-\frac{\lambda_d}{\beta_2}(t-t_0)} + \frac{\beta_2 \iota}{\lambda_d}(1 - e^{-\frac{\lambda_d}{\beta_2}(t-t_0)})$. Using [21, Th. 3.3], $\mathcal{I}$ can be extended to the interval $[t_0, \infty)$ and $\|\zeta(t)\| < \sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2 \iota}{\beta_1 \lambda_d}}$ for all $t \in [t_0, \infty)$. Under more excitation $\lambda_{\min}\{\sum_{i=1}^{N} \widehat{\Phi}'^\top(X_i, \hat{\theta})\widehat{\Phi}'(X_i, \hat{\theta})\}$ grows, thus increasing λ_e and λ_d, tightening the ultimate bound of ζ. Hence, if $\zeta(t_0) \in \mathcal{S}$, then $\zeta(t) \in \mathcal{S} \subset \mathcal{D}$ and therefore $X \in \Omega$ for all $t \geq 0$. Using (20) and (23) implies $e, r, \tilde{\theta} \in \mathcal{L}_\infty$. The fact that $x_d, \dot{x}_d, \ddot{x}_d, e, r \in \mathcal{L}_\infty$ implies $x, \dot{x} \in \mathcal{L}_\infty$, using this and the fact $\hat{\theta} \in \mathcal{L}_\infty$ is bounded by the use of the projection operator implies u is bounded. ■

V. Simulations

Comparative simulation results are provided to demonstrate the performance of the developed method with a baseline which is tracking error based (i.e., $\gamma_1 = 0$). Simulations were performed on a 6-DOF rigid body vehicle with state vector $x = [\mathrm{x}, \mathrm{y}, \mathrm{z}, \varphi, \vartheta, \psi]^\top \in \mathbb{R}^6$. The system dynamics are represented by $f(x, \dot{x}) = [-\frac{k_L}{m}\dot{\mathrm{x}}|\dot{\mathrm{x}}|[\frac{m}{s^2}], -\frac{k_L}{m}\dot{\mathrm{y}}|\dot{\mathrm{y}}|[\frac{m}{s^2}], -g - \frac{k_L}{m}\dot{\mathrm{z}}|\dot{\mathrm{z}}|[\frac{m}{s^2}], \frac{I_{yy}-I_{zz}}{I_{zz}}\dot{\vartheta}\dot{\psi} - k_A\dot{\varphi}|\dot{\varphi}|[\frac{\mathrm{rad}}{s^2}], \frac{I_{zz}-I_{xx}}{I_{yy}}\dot{\varphi}\dot{\psi} - k_A\dot{\vartheta}|\dot{\vartheta}|[\frac{\mathrm{rad}}{s^2}], \frac{I_{xx}-I_{yy}}{I_{zz}}\dot{\varphi}\dot{\vartheta} - k_A\dot{\psi}|\dot{\psi}|[\frac{\mathrm{rad}}{s^2}]]^\top$ where $k_L = 0.25$, $k_A = 0.03$, $m = 1.5\mathrm{kg}$, $g = 9.81\frac{m}{s^2}$, $I_{xx} = 0.0348\mathrm{kg} \cdot \mathrm{m}^2$, $I_{yy} = 0.0348\mathrm{kg} \cdot \mathrm{m}^2$, $I_{yy} = 0.0977\mathrm{kg} \cdot \mathrm{m}^2$, represent the linear and angular drag coefficients, mass, gravity, X-axis, Y-axis, and Z-axis inertia, respectively. Simulations were run for 100s with an update preformed every 0.01s to demonstrate the impact of the real-time adaptation. The settling time t_Δ was selected as 5s and the history stack was constructed using a sliding window of the previous 100 data points (1s of data) with the stack being updated every 5 new data points gathered or 0.05s. The tracking objective was to track the desired trajectory $x_d = [2\sin(0.5t), 1.5\cos(0.7t), 1 + 0.8\sin(0.3t), 0.2\sin(0.4t), 0.15\cos(0.6t), 0.3\sin(0.2t)]^\top \in \mathbb{R}^6$ and tracking performance $e = [e_p^\top, e_R^\top]$ where $e_p \in \mathbb{R}^3$ and $e_R \in \mathbb{R}^3$ are the errors for the position and rotation metrics, respectively. The simulation is initialized at $x(0) = [0.1, -0.1, 0.05, 0.02, -0.02, 0.01]^\top$ and $\dot{x}(0) = [0, 0, 0, 0, 0, 0]$. The DNNs are composed of 4 layers, 12 neurons,[4] and tanh activation functions. The weights of the DNN were randomly initialized from a uniform distribution $U(-0.25, 0.25)$. The gains were selected as $\alpha_1 = 15$, $\alpha_2 = 50$, $k_1 = 3$, $k_\Delta = 20$, $\beta = 0.1$, $\gamma_1 = 0.5$, $\gamma_2 = 0.001$, and $\Gamma(0) = 0.25$. The same randomly selected initial weights and control gains were used to demonstrate the performance of the adaptation under the same conditions.

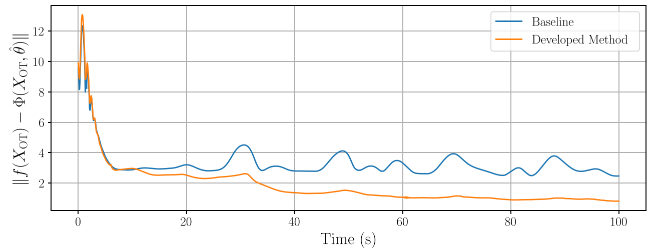

Fig. 1. Function approximation errors evaluated at off trajectory points.

TABLE I
Tracking and Function Approximation Metrics for the Developed Methods vs. Baseline Method

RMS	$\|e_R\|$ [deg]	$\|e_p\|$ [mm]	$\|\tilde{f}\|$	$\|\tilde{f}_{OT}\|$
Baseline	0.0524	8.48	0.163	3.03
CL-DNN	0.0624	9.33	0.097	1.03

The developed method improved function approximation while maintaining baseline-level control effort of 10.1 N and tracking performance. The minimal tracking difference is consistent with the function approximation capabilities of DNNs. The developed method's guaranteed parameter convergence led to a 40.54% reduction in the function approximation error which is defined as $\tilde{f} \triangleq f(X) - \Phi(X, \hat{\theta})$. To evaluate off-trajectory performance, a test set of 100 inputs denoted by X_{OT} was constructed from a uniform distribution, $U(-1, 1)$. The function approximation error for this unseen data is denoted $\tilde{f}_{OT} \triangleq f(X_{OT}) - \Phi(X_{OT}, \hat{\theta})$. The mean function approximation error for the off-trajectory points shown in Figure 1 and in Table I, which demonstrates a 65.75% improvement in off-trajectory function approximation compared to the baseline.

VI. Conclusion

This letter overcomes challenges in parameter convergence for continuous all-layer DNN adaptation, arising from the NIP nature of the inner layers by introducing a Lb-CL-DNN update law. A Lyapunov-based stability analysis guarantees ultimately bounded error convergence for both the tracking and the weight estimation errors. Simulations conducted on a 6-DOF rigid body vehicle demonstrated improvements in function approximation of 40.54% compared to the baseline method and off-trajectory simulations demonstrated 65.75% improvement for the developed method.

[4]The computational complexity of the DNN forward and backward pass is $\mathcal{O}(kL^2)$, where k is the number of layers, L is the number of neurons in each layer, and kL^2 is approximately the total number of individual weights in the DNN [13].

APPENDIX

Proof of Lemma 1: For all $\zeta \in \mathcal{D}$, $\|\zeta\| < \chi$, and hence $\|r\|, \|e\|, \|\tilde{\theta}\| < \chi$. Since $\|x\| = \|e + x_d\| \leq \|e\| + \|x_d\| \leq \|e\| + \overline{x_d}$ and $\|\dot{x}\| = \|r - \alpha_1 e + \dot{x}_d\| \leq \|r\| + \alpha_1\|e\| + \overline{\dot{x}_d}$, the bounds $\|x\| \leq \chi + \overline{x_d}$ and $\|\dot{x}\| \leq (\alpha_1 + 1)\chi + \overline{\dot{x}_d}$ hold for all $\zeta \in \mathcal{D}$. Moreover, using (3) and (8), $\ddot{x} = f(x, \dot{x}) - \Phi(X, \hat{\theta}) - k_1 r - e - \alpha_1 \dot{e} + \ddot{x}_d$. Using the structure of the DNN in (1), and because $\hat{\theta}, \theta^* \in \Theta$, it follows that there exists $\bar{\Phi} \in \mathbb{R}_{>0}$ such that $\|\Phi(X, \theta^*)\|, \|\Phi(X, \hat{\theta})\| \leq \bar{\Phi}$. Hence, using (4) yields $\|f(x, \dot{x}) - \Phi(X, \hat{\theta})\| \leq 2\overline{\Phi} + \bar{\varepsilon}$. Therefore, the definition (5) and bound $\|\ddot{x}_d\| \leq \overline{\ddot{x}_d}$ yields $\|\ddot{x}\| \leq \bar{\ddot{x}} \triangleq 2\overline{\Phi} + \bar{\varepsilon} + ((k_1 + \alpha_1) + (\alpha_1^2 + 1))\chi + \overline{\ddot{x}_d}$ for all $\zeta \in \mathcal{D}$. Since $f(x, \dot{x})$ is continuously differentiable, there exist constants $\varrho_1, \varrho_2 \in \mathbb{R}_{>0}$ such that $\|\frac{\partial f}{\partial x}\| \leq \varrho_1$ and $\|\frac{\partial f}{\partial \dot{x}}\| \leq \varrho_2$ for all $\zeta \in \mathcal{D}$. Therefore, $\|\dot{f}(x, \dot{x})\| \leq \|\frac{\partial f}{\partial x}\|\|\dot{x}\| + \|\frac{\partial f}{\partial \dot{x}}\|\|\ddot{x}\| \leq \varrho_1((\alpha_1 + 1)\chi + \overline{\dot{x}_d}) + \varrho_2\bar{\ddot{x}}$. Thus, selecting $\delta_f \triangleq \varrho_1((\alpha_1+1)\chi+\overline{\dot{x}_d})+\varrho_2\bar{\ddot{x}}$ yields $\|\dot{f}(x, \dot{x})\| \leq \delta_f$ for all $\zeta \in \mathcal{D}$. ■

Proof of Lemma 2

Proof: Consider the candidate Lyapunov function $\mathcal{V}_\Delta(z) \triangleq \frac{1}{2}\tilde{\Delta}^\top\tilde{\Delta} + \frac{1}{2}\tilde{r}^\top\tilde{r}$. Taking the derivative, using (12), (13), and canceling cross terms yields $\dot{\mathcal{V}}_\Delta(z) \leq -\tilde{\Delta}^\top k_\Delta \tilde{\Delta} - \tilde{r}^\top \alpha_2 \tilde{r} + \tilde{\Delta}^\top \dot{f}(x, \dot{x})$. Using Young's inequality and Lemma 1, $\dot{\mathcal{V}}_\Delta(z)$ can be further upper-bounded as $\dot{\mathcal{V}}_\Delta \leq -\Lambda_1\mathcal{V}_\Delta + \frac{\delta_f^2}{2k_\Delta}$ provided $\zeta \in \mathcal{D}$. Therefore, $\mathcal{V}_\Delta(z(t)) \leq \mathcal{V}_\Delta(z(t_0))e^{-\Lambda_1(t-t_0)} + \frac{\delta_f^2}{2k_\Delta\Lambda_1}(1 - e^{-\Lambda_1(t-t_0)})$ and $\|z\| \leq \sqrt{\|z(t_0)\|^2 e^{-\Lambda_1(t-t_0)} + \frac{\delta_f^2}{k_\Delta\Lambda_1}(1 - e^{-\Lambda_1(t-t_0)})}$, provided $\zeta \in \mathcal{D}$. Furthermore, if $\|\tilde{\Delta}(t_0)\| \leq \bar{\Delta}_0$ and $\hat{r}(t_0) = r(t_0)$ (because $r(t_0)$ can be measured), then $\|z(t_0)\| \leq \bar{\Delta}_0$. Then $\|z\| \leq \sqrt{\bar{\Delta}_0^2 e^{-\Lambda_1(t-t_0)} + \frac{\delta_f^2}{k_\Delta\Lambda_1}(1 - e^{-\Lambda_1(t-t_0)})}$. For the prescribed accuracy δ_Δ, using the differential inequality, the settling time $t_\Delta = t_0 + \frac{1}{\Lambda_1}\ln(\frac{k_\Delta\Lambda_1\bar{\Delta}_0^2 - \delta_f^2}{k_\Delta\Lambda_1\delta_\Delta^2 - \delta_f^2})$ is obtained after imposing $\delta_\Delta \geq \sqrt{\bar{\Delta}_0^2 e^{-\Lambda_1(t-t_0)} + \frac{\delta_f^2}{k_\Delta\Lambda_1}(1 - e^{-\Lambda_1(t-t_0)})}$, provided $\zeta \in \mathcal{D}$. For the settling time to be feasible, the argument of the natural logarithm needs to be positive; imposing this condition yields the feasibility gain condition $k_\Delta\Lambda_1 > \frac{\delta_f^2}{\delta_\Delta^2}$. ■

ACKNOWLEDGMENT

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

REFERENCES

[1] G. Joshi, J. Virdi, and G. Chowdhary, "Design and flight evaluation of deep model reference adaptive controller," in *Proc. AIAA Scitech Forum*, 2020, p. 1336.
[2] S. Basyal, J. Ting, K. Mishra, and B. C. Allen, "Augmentation of a Lyapunov-based deep neural network controller with concurrent learning for control-affine nonlinear systems," in *Proc. Amer. Control Conf.*, 2024, pp. 2870–2875.
[3] R. Sun, M. Greene, D. Le, Z. 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.
[4] O. Patil, D. Le, M. 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.
[5] R. Hart, E. Griffis, O. 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.
[6] E. Griffis, O. Patil, Z. 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.
[7] H. Lu, J. Wu, and W. Wang, "Adaptive deep neural network optimized control for a class of nonlinear strict-feedback systems with prescribed performance," *Int. J. Adapt. Control Signal Process.*, vol. 38, no. 12, pp. 1–24, 2024.
[8] J. Mei, H. Jian, Y. Li, W. Wang, and D. Lin, "Deep neural networks-based output-dependent intermittent control for a class of uncertain nonlinear systems," *Chaos Solitons Fractals*, vol. 185, Aug. 2024, Art. no. 114999.
[9] A. Pulido, K. Volle, K. Waters, Z. I. Bell, P. Ganesh, and J. Shin, "Uncertainty-aware guidance for target tracking subject to intermittent measurements using motion model learning," in *Proc. Amer. Control Conf.*, 2025, pp. 3140–3145.
[10] R. Ortega, S. Aranovskiy, A. A. Pyrkin, A. Astolfi, and A. A. Bobtsov, "New results on parameter estimation via dynamic regressor extension and mixing: Continuous and discrete-time cases," *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 2265–2272, May 2021.
[11] Y. Pan and H. Yu, "Composite learning robot control with guaranteed parrameter convergence," *Automatica*, vol. 89, pp. 415–419, Mar. 2018.
[12] G. Chowdhary, M. Mühlegg, J. How, and F. Holzapfel, "Concurrent learning adaptive model predictive control," in *Advances in Aerospace Guidance, Navigation and Control*. Berlin, Germany: Springer, 2013, pp. 29–47.
[13] W. Makumi, O. S. Patil, and W. E. Dixon, "Lyapunov-based adaptive deep system identification for approximate dynamic programming," *Automatica*, vol. 180, Oct. 2025, Art. no. 112462.
[14] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence using finite excitation," *Int. J. Adapt. Control Signal Process.*, vol. 33, pp. 1775–1787, Dec. 2019.
[15] R. Kamalapurkar, B. Reish, G. Chowdhary, and W. E. Dixon, "Concurrent learning for parameter estimation using dynamic state-derivative estimators," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3594–3601, Jul. 2017.
[16] S. B. Roy, S. Bhasin, and I. N. Kar, "Combined MRAC for unknown MIMO LTI systems with parameter convergence," *IEEE Trans. Autom. Control*, vol. 63, no. 1, pp. 283–290, Jan. 2018.
[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[18] P. Kidger and T. Lyons, "Universal approximation with deep narrow networks," in *Proc. Conf. Learn. Theory*, 2020, pp. 2306–2327.
[19] J. J. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, pp. 509–519, Jul. 1989.
[20] E. A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations*. New York, NY, USA: McGraw-Hill, 1955.
[21] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Hoboken, NJ, USA: Prentice Hall, 2002.