

# Simultaneous Online System Identification and Control using Composite Adaptive Lyapunov-Based Deep Neural Networks

Omkar Sudhir Patil, Emily J. Griffis, Wanjiku A. Makumi, and Warren E. Dixon

**Abstract**—Although deep neural network (DNN)-based controllers are popularly used to control uncertain nonlinear dynamic systems, most results use DNNs that are pretrained offline and the corresponding controller is implemented post-training. Recent advancements in adaptive control have developed controllers with Lyapunov-based update laws (i.e., control and update laws derived from a Lyapunov-based stability analysis) for updating the DNN weights online to ensure the system states track a desired trajectory. However, the update laws are based on the tracking error, and offer guarantees on only the tracking error convergence, without providing any guarantees on system identification. This paper provides the first result on simultaneous online system identification and trajectory tracking control of nonlinear systems using adaptive updates for all layers of the DNN. A combined Lyapunov-based stability analysis is provided, which guarantees that the tracking error, state-derivative estimation error, and DNN weight estimation errors are uniformly ultimately bounded. Under the persistence of excitation (PE) condition, the tracking and weight estimation errors are shown to exponentially converge to a neighborhood of the origin, where the rate of convergence and the size of this neighborhood depends on the gains and a factor quantifying PE, thus achieving system identification and enhanced trajectory tracking performance. As an outcome of the system identification, the DNN model can be propagated forward to predict and compensate for the uncertainty in dynamics under intermittent loss of state feedback. Comparative simulation results are provided on a two-link manipulator system and an unmanned underwater vehicle system with intermittent loss of state feedback, where the developed method yields significant performance improvement compared to baseline methods.

## I. INTRODUCTION

Deep neural network (DNN)-based methods have garnered popularity as a means for identification and control of uncertain nonlinear dynamic systems. Traditional DNN-based control techniques involve initial offline system identification based on datasets gathered from experimental trials [1]–[7]. Subsequently, the identified DNN is used as a model to design

The authors are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville FL 32611-6250 USA. Email: {patilomkarsudhir, emilygriffis00, makumiw, wdixon}@ufl.edu.

This research is supported in part by Office of Naval Research Grant N00014-21-1-2481, AFOSR award number FA8651-21-F-1027, and FA9550-19-1-0169. 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.

controllers using traditional model-based control techniques. However, the weight estimates of the DNN are fixed and not updated during task execution, raising questions about the model's reliability and adaptability. Moreover, it is often implicitly assumed that minimizing a loss function would result in the DNN identifying the system dynamics. Whether a system model can be identified depends on whether the system trajectories generate information sufficient for the model to be identified, which manifests in terms of the persistence of excitation (PE) condition on the model [8]. Although the PE condition is well-studied and understood in the system identification literature for linear regression models, only a few recent works remark on the PE condition for DNNs [9]–[12]. If the model is not identified, the DNN may not generalize its performance well beyond the explored trajectories. Consequently, the controller may not accurately compensate for the uncertainty, thus hazarding the stability of the closed-loop system.

Recent results in [13]–[21] offer online weight updates for the DNN-based controllers to achieve tracking error convergence. The online weight update laws are derived from a Lyapunov-based stability analysis, and the corresponding controllers are popularly known as Lyapunov-based (Lb)-DNN controllers. These results can achieve tracking error convergence regardless of whether the PE condition is satisfied. However, the update laws in these results are based only on tracking error feedback and are primarily meant to achieve tracking error convergence. Since the parameter update law converges to zero upon convergence of the tracking error in such results, it is difficult to draw conclusions regarding the accuracy of the parameter estimate. To address this problem, incorporating a prediction error, i.e., a measure of the discrepancy between the actual dynamics and their DNN-based estimate, into the adaptation law can help with parameter estimation. It is desirable to estimate the DNN parameters to achieve system identification in addition to trajectory tracking, where the identified model can be used to perform new tasks. For example, the identified model can be used to predict and compensate for the uncertain dynamics under intermittent loss of feedback [22]–[24]. However, the prediction error is difficult or often impossible to obtain since the dynamics are unknown and the state-derivative is typically

either unavailable or noisy.

The classical result in [25] develops adaptive controllers with a composite adaptation law that incorporates both tracking and prediction errors for nonlinear systems with linear-in-parameters (LIP) uncertainties, where a low-pass filter is applied on both sides of the dynamics to eliminate the unknown state-derivative term. However, extending the composite adaptation law from [25] to nonlinear-in-parameters (NIP) uncertainties such as DNNs is challenging because the inner-layer weights are embedded in nonlinear activation functions in a nested fashion. Thus, when a low-pass filter is applied to the dynamics, the resultant expression is not separable in terms of the model parameters, which introduces technical challenges as detailed in Appendix VIII.2.

**Main Contributions.** This paper provides the first result on simultaneous online system identification and trajectory tracking control of nonlinear systems using online updates for all layers of the DNN. The development involves a composite adaptation law based on a new prediction error formulation using a dynamic state-derivative observer, which is combined with the tracking error to construct a least squares-based composite adaptation law. To address the challenges posed by the nested and NIP structure of DNNs, the Jacobian of the DNN is used in a composite adaptation law. Then, a first-order Taylor series expansion of the DNN is used in the analysis to express the prediction error in terms of the parameter estimation error. Since the adaptation laws are tightly coupled with the observer and system dynamics, a combined Lyapunov-based stability analysis is performed which guarantees the tracking, observer, and parameter estimation errors are uniformly ultimately bounded (UUB). If the PE condition is satisfied, the tracking and weight estimation errors are shown to exponentially converge to a neighborhood of the origin. Specifically, guarantees on estimating the ideal DNN parameters imply accurate system identification. Thus, the identified DNN model can generalize beyond the points encountered by the system trajectory. As a result, the composite adaptive model is suitable for systems involving intermittent loss of state feedback, where the identified DNN model can be propagated forward in time to predict the uncertain dynamics when feedback is lost, under developed sufficient dwell-time conditions. To demonstrate the performance and efficacy of the developed method on different systems, comparative simulation results are provided on two systems: a robot manipulator and an unmanned underwater vehicle (UUV) with intermittent loss of state feedback. The developed composite adaptive Lb-DNN controller yields significant performance improvement when compared to the tracking error-based adaptive Lb-DNN in [16] and state-derivative observer-based disturbance rejection controllers as baseline methods.

## II. NOTATION AND PRELIMINARIES

The space of essentially bounded Lebesgue measurable functions is denoted by  $\mathcal{L}_\infty$ . The identity matrix of size  $n$

is denoted by  $I_n$ . The pseudo-inverse of full row rank matrix  $A \in \mathbb{R}^{n \times m}$  is denoted by  $A^+$ , where Given some matrix  $A \triangleq [a_{i,j}] \in \mathbb{R}^{n \times m}$ , where  $a_{i,j}$  denotes the element in the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column of  $A$ , the vectorization operator is defined as  $\text{vec}(A) \triangleq [a_{1,1}, \dots, a_{n,1}, \dots, a_{1,m}, \dots, a_{n,m}]^\top \in \mathbb{R}^{nm}$ . In the following development, we consider a fully-connected deep neural network (DNN)  $\Phi : \mathbb{R}^{L_{\text{in}}} \times \mathbb{R}^p \rightarrow \mathbb{R}^{L_{\text{out}}}$  with  $k \in \mathbb{Z}_{>0}$  hidden layers, input size  $L_{\text{in}} \in \mathbb{Z}_{>0}$ , output size  $L_{\text{out}} \in \mathbb{Z}_{>0}$ , and total number of parameters  $p \in \mathbb{Z}_{>0}$ , where the parameters include weights and bias terms. Let  $\sigma \in \mathbb{R}^{L_{\text{in}}}$  denote the DNN input and  $\theta \in \mathbb{R}^p$  denote the concatenated vector of DNN parameters. Then, a fully-connected feed-forward DNN  $\Phi(\sigma, \theta)$  is defined using a recursive relation  $\Phi_j \in \mathbb{R}^{L_{j+1}}$  given by

$$\Phi_j \triangleq \begin{cases} V_j^\top \phi_j(\Phi_{j-1}), & j \in \{1, \dots, k\}, \\ V_j^\top \sigma_a, & j = 0, \end{cases} \quad (1)$$

where  $\Phi(\sigma, \theta) = \Phi_k$ , and  $\sigma_a \triangleq [\sigma^\top \ 1]^\top$  denotes the augmented input that accounts for the bias terms,  $V_j \in \mathbb{R}^{L_j \times L_{j+1}}$  denotes the matrix of weights and biases, and  $L_j \in \mathbb{Z}_{>0}$  denotes the number of neurons in the  $j^{\text{th}}$  layer for all  $j \in \{0, \dots, k\}$  with  $L_0 \triangleq L_{\text{in}} + 1$  and  $L_{k+1} \triangleq L_{\text{out}}$ . The vector of smooth activation functions is denoted by  $\phi_j : \mathbb{R}^{L_j} \rightarrow \mathbb{R}^{L_j}$  for all  $j \in \{1, \dots, k\}$ . The activation functions at each layer are represented as  $\phi_j \triangleq [\varsigma_{j,1} \ \dots \ \varsigma_{j,L_j-1} \ 1]^\top$ , where  $\varsigma_{j,i} : \mathbb{R} \rightarrow \mathbb{R}$  denotes the activation function at the  $i^{\text{th}}$  node of the  $j^{\text{th}}$  layer. For the DNN architecture in (1), the vector of DNN weights is defined as  $\theta \triangleq [\text{vec}(V_0)^\top \ \dots \ \text{vec}(V_k)^\top]^\top$  with size  $p = \sum_{j=0}^k L_j L_{j+1}$ . The Jacobian of the DNN with respect to the weights is denoted by  $\Phi'(\sigma, \theta) \triangleq \frac{\partial}{\partial \theta} \Phi(\sigma, \theta) \in \mathbb{R}^{n \times p}$  is represented as

$$\Phi'(\sigma, \theta) = \left[ \frac{\partial \Phi(\sigma, \theta)}{\partial \text{vec}(V_0)}, \frac{\partial \Phi(\sigma, \theta)}{\partial \text{vec}(V_1)}, \dots, \frac{\partial \Phi(\sigma, \theta)}{\partial \text{vec}(V_k)} \right],$$

where  $\frac{\partial \Phi(\sigma, \theta)}{\partial \text{vec}(V_j)} \in \mathbb{R}^{n \times L_j L_{j+1}}$ . Then, applying the property  $\frac{\partial}{\partial \text{vec}(B)} \text{vec}(ABC) = C^\top \otimes A$  to the DNN architecture in (1) yields

$$\frac{\partial \Phi(\sigma, \theta)}{\partial \text{vec}(V_j)} = \left( \prod_{l=j+1}^k V_l^\top \phi'_l(\Phi_{l-1}) \right) (I_{L_{j+1}} \otimes \varphi_j^\top), \quad (2)$$

where  $\varphi_j$  is a shorthand notation defined as  $\varphi_0 \triangleq \sigma_a$  and  $\varphi_j \triangleq \phi_j(\Phi_{j-1})$  for all  $j \in \{1, \dots, k\}$ , and the Jacobian of the activation function vector at the  $j^{\text{th}}$  layer is denoted by  $\phi'_j : \mathbb{R}^{L_j} \rightarrow \mathbb{R}^{L_j \times L_j}$  and is defined as  $\phi'_j(y) \triangleq \frac{\partial}{\partial y} \phi_j(y)$ . Specifically,  $\phi'_j$  evaluates as  $\phi'_j = \text{diag}([\varsigma'_{j,1} \ \dots \ \varsigma'_{j,L_j-1} \ 0]^\top)$ , where  $\varsigma'_{j,i}(\zeta) \triangleq \frac{\partial}{\partial \zeta} \varsigma_{j,i}(\zeta)$ , and  $\text{diag}(\cdot)$  represents the diagonalization operation which returns a diagonal matrix with the elements of its input vector arranged along the diagonal. In (2), the notation  $\prod_{l=j+1}^k$  denotes the right-to-left matrix product operation, i.e.,  $\prod_{p=1}^m A_p = A_m \dots A_2 A_1$  and  $\prod_{p=a}^m A_p = I$

if  $a > m$ , and  $\otimes$  denotes the Kronecker product. To facilitate the subsequent development and analysis, the following assumption is made regarding the activation functions of the DNN.

**Assumption 1.** For each  $j \in \{0, \dots, k\}$ , the activation function  $\phi_j$ , its Jacobian  $\phi'_j$ , and Hessian  $\phi''_j(y) \triangleq \frac{\partial^2}{\partial y^2} \phi_j(y)$  are bounded as  $\|\phi_j(y)\| \leq \alpha_1 \|y\| + \alpha_0$ ,  $\|\phi'_j(y)\| \leq \beta_0$ , and  $\|\phi''_j(y)\| \leq \gamma_0$ , respectively, where  $\alpha_0, \alpha_1, \beta_0, \gamma_0 \in \mathbb{R}_{>0}$  are known constants.

*Remark 1.* Most activation functions used in practice satisfy Assumption 1. Specifically, sigmoidal activation functions (e.g., logistic function, hyperbolic tangent etc.) have  $\|\phi_j(y)\|$ ,  $\|\phi'_j(y)\|$ , and  $\|\phi''_j(y)\|$  bounded uniformly by constants. Smooth approximations of rectified linear unit (ReLUs) such as Swish grow linearly, and hence satisfy the bound  $\|\phi_j(y)\| \leq \alpha_1 \|y\| + \alpha_0$  of Assumption 1.

The following section formulates the trajectory tracking control problem and provides a control design based on the aforementioned DNN architecture.<sup>1</sup>

### III. PROBLEM FORMULATION AND CONTROL DESIGN

Consider the second order nonlinear system

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

where  $x, \dot{x} \in \mathbb{R}^n$  denote the states with available measurements,  $\ddot{x} \in \mathbb{R}^n$  is the unknown state-derivative,  $f : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  denotes an unknown continuously differentiable drift function,  $g : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$  denotes a known locally Lipschitz control effectiveness matrix, and  $u \in \mathbb{R}^m$  denotes the control input. Let the tracking error  $e \in \mathbb{R}^n$  be defined as

$$e \triangleq x - x_d(t), \quad (4)$$

where  $x_d : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$  denotes a smooth reference trajectory that is designed to satisfy  $\|x_d(t)\| \leq \bar{x}_d$ ,  $\|\dot{x}_d(t)\| \leq \bar{\dot{x}}_d$ , and  $\|\ddot{x}_d(t)\| \leq \bar{\ddot{x}}_d$  where  $\bar{x}_d, \bar{\dot{x}}_d, \bar{\ddot{x}}_d \in \mathbb{R}_{>0}$  are user-selected constants. The control objective is to design a DNN-based adaptive controller with a composite adaptation law that achieves tracking and parameter estimation. For ease of exposition, the notation  $X \triangleq [x^\top \ \dot{x}^\top]^\top \in \mathbb{R}^{2n}$  is introduced. To aid the subsequent development, the following assumptions are made.

**Assumption 2.** The function  $g$  is full row rank, and its right pseudoinverse  $g^+ : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$  given by  $g^+(x, \dot{x}) \triangleq g(x, \dot{x})^\top (g(x, \dot{x})g(x, \dot{x})^\top)^{-1}$  is assumed to be bounded.

**Assumption 3.** The drift function  $f$  is such that there exist strictly increasing smooth functions  $\varrho_1, \varrho_2 : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{\geq 0}$  satisfying  $\left\| \frac{\partial f}{\partial x}(x, \dot{x}) \right\| \leq \varrho_1(\|X\|)$  and  $\left\| \frac{\partial f}{\partial \dot{x}}(x, \dot{x}) \right\| \leq \varrho_2(\|X\|)$ .

<sup>1</sup>A fully-connected DNN is described here for simplicity in the illustration. The following control and adaptation law development can be generalized for any network architecture  $\Phi$  with a corresponding Jacobian  $\Phi'$ . The reader is referred to [18] and [19] for extending the subsequent development to ResNets and LSTMs, respectively.

Assumption 2 implies the system is not underactuated. Many electromechanical systems satisfy this assumption, e.g., the robot manipulator and UUV considered in Section V of this paper, Stewart platforms, hexapod robots, etc. The developed method can be extended on a case-by-case basis to underactuated systems using standard nonlinear control tools (e.g., backstepping) unique for such underactuated systems. Because a universal closed-form stabilizing nonlinear controller cannot be obtained for an arbitrary underactuated, system even with perfect model knowledge, the derivation has to be done on a case-by-case basis for each specific underactuated system, depending on how  $g$  is structured. Assumption 3 is reasonable for continuously differentiable  $f$  because, for most practical systems, the bounding functions  $\varrho_1, \varrho_2$  can be constructed using basis functions such as polynomials or exponentials. This assumption is made to obtain accuracy guarantees on a high-gain state-derivative estimator constructed in the subsequent development.

#### A. Control Development

To facilitate the control development, let the auxiliary error  $r \in \mathbb{R}^n$  be defined as

$$r \triangleq \dot{e} + \alpha_1 e, \quad (5)$$

where  $\alpha_1 \in \mathbb{R}_{>0}$  denotes a constant control gain. Taking the time-derivative on both sides of (5), and substituting (3)-(5) yields

$$\dot{r} = f(x, \dot{x}) + g(x, \dot{x})u - \ddot{x}_d(t) + \alpha_1(r - \alpha_1 e). \quad (6)$$

DNNs are a powerful tool for approximating unstructured uncertainties, such as  $f$ , based on their universal function approximation capabilities on compact sets [26]. To this end, consider a compact set  $\Omega \subset \mathbb{R}^n$  that will be explicitly defined later in the development. Additionally, let  $\Phi : \mathbb{R}^{2n} \times \mathbb{R}^p \rightarrow \mathbb{R}^n$  denote a general DNN architecture, where  $p \in \mathbb{Z}_{>0}$  denotes the total number of DNN parameters. To formulate the DNN-based approximation, let the loss function  $\mathcal{L} : \mathbb{R}^p \rightarrow \mathbb{R}_{\geq 0}$  be defined as

$$\mathcal{L}(\theta) \triangleq \int_{\Omega} \left( \|f(x, \dot{x}) - \Phi(X, \theta)\|^2 + \varsigma \|\theta\|^2 \right) d\mu(x), \quad (7)$$

where  $\mu$  denotes the Lebesgue measure,  $\varsigma \in \mathbb{R}_{>0}$  denotes a regularizing constant, and the term  $\varsigma \|\theta\|^2$  represents  $L_2$  regularization (also popularly known as ridge regression in the machine learning community) [27, Sec. 7.1.1]. Additionally, a user-selected compact convex parameter search space  $\Theta \subset \mathbb{R}^p$  satisfying  $0_p \in \Theta$  is considered with a bound on the search space  $\bar{\theta} = \max_{\theta \in \Theta} \|\theta\|$ . The objective is to identify the vector of ideal DNN parameters  $\theta^* \in \Theta$  defined as

$$\theta^* \triangleq \arg \min_{\theta \in \Theta} \mathcal{L}(\theta). \quad (8)$$

For the parameter estimation problem to be well-posed, the solution  $\theta^*$  to (8) is considered to be unique. To this end, the following assumption is made.

**Assumption 4.** The loss function  $\mathcal{L}$  is strictly convex over the set  $\Theta$ .<sup>2</sup>

*Remark 2.* Notice that the universal function approximation property of DNNs was not invoked in the definition of  $\theta^*$ . The universal function approximation theorem [26, Theorem 3.1] states that the function space of DNNs is dense in the space of continuous functions  $\mathcal{C}(\Omega)$ . As a result, for any prescribed  $\bar{\varepsilon} > 0$ , there exists a DNN  $\Phi$  and a corresponding parameter  $\theta$  such that  $\max_{x \in \Omega} \|f(x, \dot{x}) - \Phi(X, \theta)\| < \bar{\varepsilon}$ , and therefore  $\int_{\Omega} \|f(x, \dot{x}) - \Phi(X, \theta)\|^2 d\mu(x) < \bar{\varepsilon}^2 \mu(\Omega)$ . However, it is not known how to construct a parameter search space  $\Theta$  for an arbitrarily prescribed accuracy  $\bar{\varepsilon}$ . Therefore, we allow  $\Theta$  to be arbitrarily constructed in the analysis, at the loss of guarantees on the approximation accuracy. Although the accuracy  $\bar{\varepsilon}$  which bounds  $\max_{x \in \Omega} \|f(x, \dot{x}) - \Phi(X, \theta)\|$  might no longer be arbitrary in this case, it would still be finite due to the continuity of  $f$  and  $\Phi$ , where minimizing the loss in (7) would achieve the best regularized approximation of  $f$ .

Due to Remark 2, the drift function can be modeled as

$$f(x, \dot{x}) = \Phi(X, \theta^*) + \varepsilon(X), \quad (9)$$

where  $\varepsilon : \mathbb{R}^{2n} \rightarrow \mathbb{R}^n$  denotes an unknown function reconstruction error that can be bounded as  $\max_{X \in \Omega} \|\varepsilon(X)\| \leq \bar{\varepsilon}$ . Based on (6) and the subsequent analysis, the control input is designed as

$$u = g^+(x, \dot{x})(\ddot{x}_d(t) - (\alpha_1 + k_r)r + (\alpha_1^2 - 1)e - \Phi(X, \hat{\theta})), \quad (10)$$

where  $k_r \in \mathbb{R}_{>0}$  denotes a constant control gain, and  $\hat{\theta} \in \mathbb{R}^p$  denotes the adaptive estimate of the DNN weights  $\theta^*$  that is developed using subsequently designed adaptation laws. Substituting (9) and (10) into (6) yields

$$\dot{r} = \Phi(X, \theta^*) - \Phi(X, \hat{\theta}) + \varepsilon(X) - e - k_r r. \quad (11)$$

### B. Composite Adaptation Law

The classical result in [25] develops a composite adaptation law using tracking and prediction errors for robot manipulators that involve linearly parameterized uncertainties in the absence of exogenous disturbances. However, for NIP uncertainties such as DNNs, the traditional development of the prediction error is not applicable and a new approach is required. Hence, an innovation of this paper is a new prediction error

<sup>2</sup>The assumption of local strict convexity allows the subsequent development to be analyzed from a convex optimization perspective, which otherwise would be non-convex due to the nested NIP structure of DNNs. Specifically, due to the strict convexity of the regularizing term  $\varsigma \|\theta\|^2$  in (7), there exists  $\sigma \in \mathbb{R}_{>0}$  which ensures  $\mathcal{L}(\theta)$  is convex for all  $\theta \in \Theta$ . Additionally, the regularizing term has other advantages such as mitigation of overfitting [27, Sec. 7.1.1]. However, selecting very high values of  $\sigma$  can be counterproductive as it can obscure the contribution of the  $\|f(x, \dot{x}) - \Phi(X, \theta)\|^2$  term to the loss function while also causing underfitting [27, Sec. 7.1.1]; therefore, there is a tradeoff between selecting low vs. high values of  $\varsigma$ . For further information on when the loss is strictly convex with a substantial contribution from the  $\|f(x, \dot{x}) - \Phi(X, \theta)\|^2$  term, the reader is referred to the identifiability conditions derived in [28, Sec. II.A].

formulation based on a dynamic state-derivative estimator that provides an estimate of the ground truth value of the drift  $f$  (c.f., [29]). The dynamic state-derivative observer is designed as

$$\begin{aligned} \dot{\hat{r}} &= g(x, \dot{x})u - \ddot{x}_d(t) + \alpha_1(r - \alpha_1 e) + \hat{f} + \alpha_2 \tilde{r}, \\ \dot{\hat{f}} &= k_f(\dot{\hat{r}} + \alpha_2 \tilde{r}) + \tilde{r}, \end{aligned} \quad (12)$$

where  $\hat{r}, \hat{f} \in \mathbb{R}^n$  denote the observer estimates of  $r$  and  $f$ , respectively,  $\tilde{r}, \tilde{f} \in \mathbb{R}^n$  denote the observer errors  $\tilde{r} \triangleq r - \hat{r}$  and  $\tilde{f} \triangleq f(x, \dot{x}) - \hat{f}$ , respectively, and  $\alpha_2, k_f \in \mathbb{R}_{>0}$  denote constant observer gains. As is typical of observer designs, observer error  $\tilde{r}$  is known because  $r$  and  $\hat{r}$  are known, and is used as feedback to the observer in (9) to estimate  $f$ . Since  $\tilde{r}$  is unknown, (12) can be implemented by integrating both sides and using the relation  $\int_{t_0}^t \dot{\tilde{r}}(\tau) d\tau = \tilde{r}(t) - \tilde{r}(t_0)$  to obtain  $\hat{f}(t) = \hat{f}(t_0) + k_f \tilde{r}(t) - k_f \tilde{r}(t_0) + \int_{t_0}^t (k_f \alpha_2 + 1) \tilde{r}(\tau) d\tau$ , where  $t_0$  denotes the initial time. Note that although  $\hat{f}$  generated by the state-derivative estimator can also be used to compensate for  $f$ , such an approach results in a robust high-gain design which does not achieve the system identification objective and can cause large overshoots in the control input.

Taking the time-derivative of  $\tilde{r}$  and  $\tilde{f}$  and substituting their definitions along with (6) and (12) yields

$$\dot{\tilde{r}} = \tilde{f} - \alpha_2 \tilde{r}, \quad \dot{\tilde{f}} = \dot{f} - k_f \tilde{f} - \tilde{r}, \quad (13)$$

where  $\dot{f} \triangleq \frac{\partial f}{\partial x} \dot{x} + \frac{\partial f}{\partial \dot{x}} \ddot{x}$ , and  $\dot{\tilde{f}}$  is derived after substituting in  $\dot{\tilde{r}}$ . Using the dynamic state-derivative estimator, the prediction error  $E \in \mathbb{R}^n$  is designed as

$$E \triangleq \hat{f} - \Phi(X, \hat{\theta}). \quad (14)$$

Then, the composite least squares adaptation law is designed as

$$\dot{\hat{\theta}} = \text{proj} \left( -k_{\hat{\theta}} \Gamma(t) \hat{\theta} + \Gamma(t) \Phi'^\top(X, \hat{\theta}) (r + \alpha_3 E) \right), \quad (15)$$

where  $\text{proj}(\cdot)$  denotes a continuous projection operator (cf. [30, Appendix E]) which ensures  $\dot{\hat{\theta}}(t) \in \mathcal{B}_{\bar{\theta}} \triangleq \{\theta \in \mathbb{R}^p : \|\theta\| \leq \bar{\theta}\}$  for all  $t \in \mathbb{R}_{\geq 0}$ ,  $\alpha_3, k_{\hat{\theta}} \in \mathbb{R}_{>0}$  denote constant gains,  $\Phi'(X, \hat{\theta}) \in \mathbb{R}^{n \times p}$  denotes the Jacobian  $\Phi'(X, \hat{\theta}) \triangleq \frac{\partial \Phi(X, \hat{\theta})}{\partial \hat{\theta}}$ . Similar development could also be used to derive the Jacobian for other DNN architectures. The term  $\Gamma : \mathbb{R}_{\geq t_0} \rightarrow \mathbb{R}^{p \times p}$  denotes a positive-definite (PD) time-varying least squares adaptation gain matrix that is a solution to [25, Eqns. (16) and (17)]

$$\frac{d}{dt} \Gamma^{-1} = \begin{cases} -\beta(t) \Gamma^{-1} & \lambda_{\Gamma, \min} < \lambda_{\min}(\Gamma) \\ +\Phi'^\top(X, \hat{\theta}) \Phi'(X, \hat{\theta}), & \text{and } \lambda_{\max}(\Gamma) < \lambda_{\Gamma, \max} \\ 0_{p \times p}, & \text{otherwise,} \end{cases} \quad (16)$$

where  $\lambda_{\Gamma, \min}, \lambda_{\Gamma, \max} \in \mathbb{R}_{>0}$  are user-prescribed lower and upper bounds on the minimum and maximum eigenvalues

of  $\Gamma$ , respectively. The term  $\beta : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  represents a bounded-gain time-varying forgetting factor designed as

$$\beta(t) \triangleq \beta_0 \left( 1 - \frac{\|\Gamma(t)\|}{\varkappa_0} \right), \quad (17)$$

where  $\beta_0 \in \mathbb{R}_{>0}$  are user-defined constants denoting the maximum forgetting rate and the bound on  $\lambda_{\max}\{\Gamma(t)\}$ , respectively. The adaptation gain in (16) is initialized to be PD such that  $\lambda_{\Gamma,\min} < \|\Gamma(t_0)\| < \varkappa_0 \leq \lambda_{\Gamma,\max}$ , and it can be shown that  $\Gamma(t)$  remains PD with  $\lambda_{\Gamma,\min} < \|\Gamma(t)\| < \varkappa_0 \leq \lambda_{\Gamma,\max}$  for all  $t \in \mathbb{R}_{\geq t_0}$  [25]. The term  $\beta(t)$  can be lower bounded as  $\beta \geq \beta_1$ , where  $\beta_1 \in \mathbb{R}_{\geq 0}$  is a constant which satisfies the properties stated in the following remark.

*Remark 3.* Consider the case when  $\Phi'(X, \hat{\theta})$  satisfies the uniform PE condition, i.e., there exists constants  $\varphi_1, \varphi_2 \in \mathbb{R}_{>0}$  for all  $\underline{t} \in \mathbb{R}_{\geq t_0}$  and some  $T \in \mathbb{R}_{>0}$  such that  $\varphi_1 I_p \leq \int_{\underline{t}}^{t+T} \Phi'^\top(X(\tau), \hat{\theta}(\tau)) \Phi'(X(\tau), \hat{\theta}(\tau)) d\tau \leq \varphi_2 I_p$  for all  $(X(t_0), \hat{\theta}(t_0)) \in \mathbb{R}^n \times \mathbb{R}^p$ . In this case, it can be shown that  $\beta_1 > 0$  [25, Sec. 4.2].

The following section shows the stability analysis for the developed DNN-based composite adaptive control method over the time-interval  $[t_0, \infty) \subseteq \mathbb{R}_{\geq 0}$ .

#### IV. STABILITY ANALYSIS

DNNs are nonlinear with respect to the weights. Designing adaptive controllers and performing stability analyses for systems that are NIP has historically been a challenging task. A method to address the NIP structure of the uncertainty, especially for DNNs, is to use a first-order Taylor series approximation [16]. Let  $\tilde{\theta} \triangleq \theta^* - \hat{\theta} \in \mathbb{R}^p$  denote the parameter estimation error. Applying Taylor's first-order theorem [31, Theorem 5.15] yields

$$\Phi(X, \theta^*) - \Phi(X, \hat{\theta}) = \Phi'(X, \hat{\theta}) \tilde{\theta} + R(X, \tilde{\theta}), \quad (18)$$

where  $R(X, \tilde{\theta}) \in \mathbb{R}^n$  denotes the Lagrange remainder. Substituting (18) into (11) yields the closed-loop error system

$$\dot{r} = \Phi'(X, \hat{\theta}) \tilde{\theta} + \Delta - e - k_r r, \quad (19)$$

where  $\Delta \in \mathbb{R}^n$  is defined as  $\Delta \triangleq R(X, \tilde{\theta}) + \varepsilon(X)$ . To facilitate the subsequent analysis, the prediction error  $E$  in (14) can be rewritten by adding and subtracting  $f$ , substituting in (9) and (18), and using the relation  $\hat{f} = f - \tilde{f}$ , which yields

$$E = \Phi'(X, \hat{\theta}) \tilde{\theta} - \tilde{f} + \Delta. \quad (20)$$

Taking the time-derivative of  $\tilde{\theta}$ , substituting (15), and then applying (20) and the relation  $\hat{\theta} = \theta^* - \tilde{\theta}$  yields the parameter estimation error dynamics

$$\begin{aligned} \dot{\tilde{\theta}} = & -\text{proj} \left( \Gamma(t) \left( k_{\hat{\theta}} + \alpha_3 \Phi'^\top(X, \hat{\theta}) \Phi'(X, \hat{\theta}) \right) \tilde{\theta} \right. \\ & + \Gamma(t) \Phi'^\top(X, \hat{\theta}) r - \alpha_3 \Gamma(t) \Phi'^\top(X, \hat{\theta}) \tilde{f} \\ & \left. + \alpha_3 \Gamma(t) \Phi'^\top(X, \hat{\theta}) \Delta - k_{\hat{\theta}} \Gamma(t) \theta^* \right). \end{aligned} \quad (21)$$

Let  $z \triangleq [e^\top \ r^\top \ \tilde{r}^\top \ \tilde{f}^\top \ \tilde{\theta}^\top]^\top \in \mathbb{R}^{4n+p}$  denote the concatenated state. It follows from using (4), (5), and the fact that  $\|x_d(t)\| \leq \bar{x}_d$  and  $\|\dot{x}_d(t)\| \leq \bar{\dot{x}}_d$  for all  $t \in \mathbb{R}_{\geq 0}$ , that the state  $X$  can be bounded as

$$\|X\| \leq (\alpha_1 + 2) \|z\| + \bar{x}_d + \bar{\dot{x}}_d. \quad (22)$$

By Taylor's theorem [31, Theorem 5.15], it can be shown that if  $\bar{H} \in \mathbb{R}_{>0}$  is a bound on the DNN Hessian,  $\left\| \frac{\partial^2 \Phi(X, \theta)}{\partial \theta} \right\| \leq \bar{H}$  for all  $X \in \Omega$  and  $\theta \in \Theta$ , then the Lagrange remainder term can be bounded as  $\|R(X, \tilde{\theta})\| \leq \frac{\bar{H}}{2} \|\tilde{\theta}\|^2$ . By examining the structure of the DNN and its Jacobian in (1) and (2), it can be shown under Assumption 1 and boundedness of the parameter set  $\Theta$  that there exists<sup>3</sup> a polynomial function  $\rho_0 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  of the form  $\rho_0(\|X\|) = a_2 \|X\|^2 + a_1 \|X\|^1 + a_0$  with constants  $a_2, a_1, a_0 \in \mathbb{R}_{>0}$  such that the DNN Hessian is bounded as  $\left\| \frac{\partial^2 \Phi(X, \theta)}{\partial \theta} \right\| \leq \rho_0(\|X\|)$ . Therefore, the Lagrange remainder term is bounded as  $\|R(X, \tilde{\theta})\| \leq \rho_0(\|X\|) \|\tilde{\theta}\|^2$ . Because  $\rho_0$  is a second order polynomial, there exists a second order polynomial  $\rho_1 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  such that  $\rho_1(\|z\|) = \rho_0((\alpha_1 + 2) \|z\| + \bar{x}_d + \bar{\dot{x}}_d)$ . Therefore, taking the norm of  $\Delta$  and using triangle inequality and the bounds  $\max_{X \in \Omega} \|\varepsilon(X)\| \leq \bar{\varepsilon}$  from Remark 2 yields

$$\begin{aligned} \|\Delta(X, \tilde{\theta})\| &\leq \rho_1(\|z\|) \|\tilde{\theta}\|^2 + \bar{\varepsilon} \\ &\leq \rho_1(\|z\|) \|z\|^2 + \bar{\varepsilon}. \end{aligned} \quad (23)$$

Since the universal function approximation property of the DNN stated in (9) holds only on the compact domain  $\Omega$ , the subsequent stability analysis requires ensuring  $X(t) \in \Omega$  for all  $t \in [t_0, \infty)$ . Due to (22),  $X$  can be restricted to  $\Omega$  by obtaining a stability result which constrains  $z$  in an appropriate compact domain. To this end, the compact domain in which  $z$  is supposed to lie,

$$\mathcal{D} \triangleq \{\iota \in \mathbb{R}^{2n+p} : \|\iota\| \leq \chi\}, \quad (24)$$

is constructed, where  $\chi \in \mathbb{R}_{>0}$  is a user-selected constant explicitly defined later in the analysis. Then using (22), it follows that if  $z \in \mathcal{D}$ , then  $X \in \Omega$ , where  $\Omega$  can now explicitly be constructed as

$$\Omega \triangleq \{\iota \in \mathbb{R}^{2n} : \|\iota\| \leq (\alpha_1 + 2) \chi + \bar{x}_d + \bar{\dot{x}}_d\}.$$

<sup>3</sup>For details on the explicit computation of such a polynomial, the reader is referred to [32, Theorem 1].

To facilitate the stability analysis, let  $V : \mathbb{R}^{4n+p} \rightarrow \mathbb{R}_{\geq 0}$  be the candidate Lyapunov function defined as

$$V(z) = \frac{1}{2}e^\top e + \frac{1}{2}r^\top r + \frac{1}{2}\tilde{r}^\top \tilde{r} + \frac{1}{2}\tilde{f}^\top \tilde{f} + \frac{1}{2}\tilde{\theta}^\top \Gamma^{-1}(t)\tilde{\theta}, \quad (25)$$

which satisfies the inequality

$$\lambda_1 \|z\|^2 \leq V(z) \leq \lambda_2 \|z\|^2, \quad (26)$$

where  $\lambda_1 \triangleq \min\{\frac{1}{2}, \frac{1}{2\lambda_{\Gamma,\max}}\} \in \mathbb{R}_{>0}$  and  $\lambda_2 \triangleq \max\{\frac{1}{2}, \frac{1}{2\lambda_{\Gamma,\min}}\} \in \mathbb{R}_{>0}$ . Taking the time-derivative of  $V(z)$ , substituting in (5), (13), (16), (19), and (21), applying the property of projection operators  $-\tilde{\theta}^\top \Gamma^{-1}(t)\text{proj}(\mu) \leq -\theta^\top \Gamma^{-1}(t)\mu$  [30, Lemma E.IV], and canceling coupling terms yields

$$\begin{aligned} \dot{V} &\leq -\alpha_1 \|e\|^2 - k_r \|r\|^2 - \alpha_2 \|\tilde{r}\|^2 - k_f \|\tilde{f}\|^2 \\ &\quad - \left(k_{\hat{\theta}} + \frac{\beta(t)}{2\lambda_{\Gamma,\max}}\right) \|\tilde{\theta}\|^2 + r^\top \Delta + \tilde{f}^\top \dot{f} \\ &\quad - \left(\alpha_3 - \frac{1}{2}\right) \tilde{\theta}^\top \Phi'^\top(X, \hat{\theta}) \Phi'(X, \hat{\theta}) \tilde{\theta} \\ &\quad + \alpha_3 \tilde{\theta}^\top \Phi'^\top(X, \hat{\theta}) (\tilde{f} - \Delta) + k_{\hat{\theta}} \tilde{\theta}^\top \theta^*. \end{aligned} \quad (27)$$

Because  $\bar{\theta} = \max_{\theta \in \Theta} \|\theta\|$  and  $\theta^* \triangleq \arg \min_{\theta \in \Theta} \mathcal{L}(\theta)$ , it follows that  $\|\theta^*\| \leq \bar{\theta}$ . Furthermore, due to the use of the projection operator,  $\|\hat{\theta}\| \leq \bar{\theta}$ . Hence,  $\|\tilde{\theta}\| \leq 2\bar{\theta}$ . By [32, Lemma 2], there exist constants  $v_1, v_2 \in \mathbb{R}_{\geq 0}$  such that  $\|\Phi'(X, \hat{\theta})\| \leq v_1 \|X\| + v_2$ . Additionally,  $\|\dot{f}\|$  can be bounded as  $\|\dot{f}\| \leq \left\| \frac{\partial f}{\partial x} \right\| \|\dot{x}\| + \left\| \frac{\partial f}{\partial \dot{x}} \right\| \|\ddot{x}\| \leq \left\| \frac{\partial f}{\partial x} \right\| (\|\dot{e}\| + \|\dot{x}_d\|) + \left\| \frac{\partial f}{\partial \dot{x}} \right\| (\|\dot{r}\| + \alpha_1 \|\dot{e}\| + \|\dot{x}_d\|)$ . Therefore, it can be shown that  $\|\dot{f}\| \leq \rho_2(\|z\|)$ , where  $\rho_2 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  is a strictly increasing function defined as

$$\begin{aligned} \rho_2(\|z\|) &\triangleq \varrho_3(\|z\|) ((\alpha_1 + 1) \|z\| + \bar{x}_d) \\ &\quad + \varrho_4(\|z\|) \left( \rho_1(\|z\|) \|z\|^2 + (\bar{x}_d v_1 + \dot{x}_d v_1 \right. \\ &\quad \left. + v_2 + \alpha_1^2 + \alpha_1 + k_r + 1 \|z\|) \right. \\ &\quad \left. + \bar{\varepsilon} + \bar{x}_d \right) + (\alpha_1 + 2) v_1 \|z\|^2, \end{aligned}$$

and  $\varrho_3, \varrho_4 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  are strictly increasing functions defined as

$$\varrho_3(\|z\|) \triangleq \varrho_1((\alpha_1 + 2) \|z\| + \bar{x}_d + \dot{x}_d)$$

and

$$\varrho_4(\|z\|) \triangleq \varrho_2((\alpha_1 + 2) \|z\| + \bar{x}_d + \dot{x}_d),$$

and  $\varrho_1, \varrho_2$  are defined in Assumption 3. Therefore, using Young's inequality, the term  $\tilde{f}^\top \dot{f}$  in (27) can be bounded as  $\tilde{f}^\top \dot{f} \leq \frac{k_f \|\tilde{f}\|^2}{2} + \frac{\rho_2^2(\|z\|)}{2k_f}$ . By first-order Taylor's theorem and the chain rule,  $\rho_2^2(\|z\|)$  can be represented as  $\rho_2^2(\|z\|) = \rho_2^2(0) + 2\rho_2(0) \rho'_2(0) \|z\| + (2\rho_2'(\tau) + 2\rho_2(\tau) \rho''_2(\tau)) \|z\|^2$

for some  $\tau \in (0, \|z\|)$ , where  $\rho'_2, \rho''_2$  denote the first and second derivatives of  $\rho_2$ , respectively. Furthermore, because  $\|z\| \leq \frac{\|z\|^2}{2} + \frac{1}{2}$ , it follows that  $2\rho_2(0) \rho'_2(0) \|z\| \leq \rho_2(0) \rho'_2(0) \|z\|^2 + \rho_2(0) \rho''_2(0)$ . Additionally, consider a strictly increasing function  $\rho_3 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  such that  $2\rho_2'(\tau) + 2\rho_2(\tau) \rho''_2(\tau) + \rho_2(0) \rho'_2(0) \leq \rho_3(\|z\|)$  for all  $\tau \in (0, \|z\|)$ , implying  $\rho_2^2(\|z\|) \leq \rho_2^2(0) + \rho_2(0) \rho'_2(0) + \rho_3(\|z\|) \|z\|^2$ . Hence,  $\tilde{f}^\top \dot{f} \leq \frac{k_f \|\tilde{f}\|^2}{2} + \frac{\rho_3(\|z\|)}{2k_f} \|z\|^2 + \frac{\rho_2^2(0) + \rho_2(0) \rho'_2(0)}{2k_f}$ .

In (27), because  $r^\top \Delta \leq \rho_1(\|z\|) \|z\|^2 \|r\| + \bar{\varepsilon} \|r\|$ , it follows by multiplying and dividing with  $\frac{k_r}{2}$  and using Young's inequality that  $\rho_1(\|z\|) \|z\|^2 \|r\| \leq \frac{k_r}{4} \|r\|^2 + \frac{\rho_1^2(\|z\|) \|z\|^4}{k_r}$  and  $\bar{\varepsilon} \|r\| \leq \frac{k_r}{4} \|r\|^2 + \frac{\bar{\varepsilon}^2}{k_r}$ . Therefore  $r^\top \Delta \leq \frac{k_r}{2} \|r\|^2 + \frac{\rho_1^2(\|z\|) \|z\|^4}{k_r} + \frac{\bar{\varepsilon}^2}{k_r}$ . Furthermore, because  $\tilde{\theta}^\top \theta^* \leq \left(\frac{\|\tilde{\theta}\|}{\sqrt{2}}\right) \sqrt{2\bar{\theta}} \leq \frac{1}{4} \|\tilde{\theta}\|^2 + \bar{\theta}^2$  by Young's inequality, it follows that  $k_{\hat{\theta}} \tilde{\theta}^\top \theta^* \leq \frac{k_{\hat{\theta}}}{4} \|\tilde{\theta}\|^2 + k_{\hat{\theta}} \bar{\theta}^2$ . As for the term  $\alpha_3 \tilde{\theta}^\top \Phi'^\top(X, \hat{\theta}) (\tilde{f} - \Delta)$ , recall the inequalities (23) and  $\|\Phi'(X, \hat{\theta})\| \leq v_1 \|X\| + v_2$ . Using these inequalities, it can be shown that  $\alpha_3 \tilde{\theta}^\top \Phi'^\top(X, \hat{\theta}) (\tilde{f} - \Delta) \leq \rho_4(\|z\|) \|z\| \|\tilde{\theta}\| + \alpha_3 \bar{\varepsilon} (v_1 \bar{x}_d + v_1 \dot{x}_d + v_2) \|\tilde{\theta}\|$ , where  $\rho_4 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  is a strictly increasing function defined as  $\rho_4(\|z\|) = \alpha_3(v_1(\alpha_1 + 2) \|z\| + v_1 \bar{x}_d + v_1 \dot{x}_d + v_2)(\|z\| + \rho_1(\|z\|) \|z\|^2) + \alpha_3 v_1(\alpha_1 + 2) \bar{\varepsilon} \|z\|$ . By Young's inequality, it follows that  $\alpha_3 \tilde{\theta}^\top \Phi'^\top(X, \hat{\theta}) (\tilde{f} - \Delta) \leq \frac{k_{\hat{\theta}}}{2} \|\tilde{\theta}\|^2 + \frac{\rho_4^2(\|z\|) \|z\|^2}{k_{\hat{\theta}}} + \frac{2(\alpha_3 \bar{\varepsilon} (v_1 \bar{x}_d + v_1 \dot{x}_d + v_2))^2}{k_{\hat{\theta}}}$ . Hence,  $\dot{V}$  can be further upper-bounded as

$$\begin{aligned} \dot{V} &\leq -(k_{\min} - \rho(\|z\|)) \|z\|^2 + c \\ &\quad - \left(\alpha_3 - \frac{1}{2}\right) \tilde{\theta}^\top \Phi'^\top(X, \hat{\theta}) \Phi'(X, \hat{\theta}) \tilde{\theta}, \end{aligned} \quad (28)$$

where  $k_{\min} \triangleq \min\{\alpha_1, \frac{k_r}{2}, \alpha_2, \frac{k_f}{2}, \frac{k_{\hat{\theta}}}{4} + \frac{\beta_1}{2\lambda_{\Gamma,\max}}\}$ ,  $\rho(\|z\|) = \frac{\rho_2^2(\|z\|) \|z\|^2}{k_r} + \frac{\rho_3(\|z\|)}{2k_f} + \frac{\rho_4^2(\|z\|)}{k_{\hat{\theta}}}$ , and  $c = \frac{\bar{\varepsilon}^2}{k_r} + \frac{\rho_2^2(0) + \rho_2(0) \rho'_2(0)}{2k_f} + \frac{2(\alpha_3 \bar{\varepsilon} (v_1 \bar{x}_d + v_1 \dot{x}_d + v_2))^2}{k_{\hat{\theta}}} + k_{\hat{\theta}} \bar{\theta}^2$ . The term  $\chi$  denoting the radius of  $\mathcal{D}$  is now explicitly defined as  $\chi \triangleq \bar{\rho}^{-1}(k_{\min} - \lambda_3 - \rho(0))$ , where  $\bar{\rho} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  is a strictly increasing invertible function defined as  $\bar{\rho}(\|z\|) \triangleq \rho(\|z\|) - \rho(0)$ , and  $\lambda_3 \in \mathbb{R}_{>0}$  is the desired rate of convergence. To facilitate the subsequent analysis, the following gain condition is introduced

$$\min \left\{ k_{\min} - \lambda_3 - \rho \left( \sqrt{\frac{\lambda_2 c}{\lambda_1 \lambda_3}} \right), \alpha_3 - \frac{1}{2} \right\} > 0. \quad (29)$$

Additionally, the set  $\mathcal{S} \triangleq \{\zeta \in \mathbb{R}^{4n+p} : \|\zeta\| \leq \sqrt{\frac{\lambda_1}{\lambda_2} \chi^2 - \frac{c}{\lambda_3}}\}$  is defined to initialize  $z$  in the subsequent analysis, where it is shown that if  $z(t_0) \in \mathcal{S} \subset \mathcal{D}$ , then  $z(t)$  exhibits exponential

converge to a neighborhood of the origin and does not escape  $\mathcal{D}$ . The following theorem states the main result of this paper.

**Theorem 1.** *Let Assumptions 1-4 and the gain condition in (29) hold. Then, for the system in (3), the DNN-based controller in (10) and the composite adaptation law in (15) ensures that  $\|z(t)\| \leq \sqrt{\frac{\lambda_2}{\lambda_1} \|z(t_0)\|^2 e^{-\frac{\lambda_3}{\lambda_2}(t-t_0)} + \frac{\lambda_2 c}{\lambda_1 \lambda_3} \left(1 - e^{-\frac{\lambda_3}{\lambda_2}(t-t_0)}\right)}$  for all  $t \in [t_0, \infty)$ , provided that  $\|z(t_0)\| \in \mathcal{S}$ .*

*Proof:* Due to the facts that  $c > 0$  and  $\lambda_1 \leq \lambda_2$  from (26), it follows from the definitions of  $\mathcal{S}$  and  $\mathcal{D}$  that  $\mathcal{S} \subset \mathcal{D}$ . As a result,  $z(t_0) \in \mathcal{S}$  implies  $z(t_0) \in \text{int}(\mathcal{D})$ , where the notation  $\text{int}(\cdot)$  denotes the interior. Therefore, because the solution  $t \mapsto z(t)$  is continuous<sup>4</sup>, there exists a time-interval  $\mathcal{I} \triangleq [t_0, t_1]$  such that  $z(t) \in \mathcal{D}$  for all  $t \in \mathcal{I}$ . In the subsequent stability analysis, we analyze the convergence properties of the solutions and also establish that  $\mathcal{I}$  can be extended to  $[t_0, \infty)$ . Consider the candidate Lyapunov function in (25). Then, using (26) and (28), when the gain condition in (29) is satisfied,  $V$  can be upper-bounded as

$$\dot{V} \leq -\frac{\lambda_3}{\lambda_2} V + c, \quad (30)$$

for all  $t \in \mathcal{I}$ . Solving the differential inequality in (30) over the time-interval  $\mathcal{I}$  yields

$$V(z(t)) \leq V(z(t_0)) e^{-\frac{\lambda_3}{\lambda_2}(t-t_0)} + \frac{\lambda_2 c}{\lambda_3} \left(1 - e^{-\frac{\lambda_3}{\lambda_2}(t-t_0)}\right), \quad (31)$$

for all  $t \in \mathcal{I}$ . Then applying (26) to (31) yields

$$\|z(t)\| \leq \sqrt{\frac{\lambda_2}{\lambda_1} \|z(t_0)\|^2 e^{-\frac{\lambda_3}{\lambda_2}(t-t_0)} + \frac{\lambda_2 c}{\lambda_1 \lambda_3} \left(1 - e^{-\frac{\lambda_3}{\lambda_2}(t-t_0)}\right)}, \quad (32)$$

for all  $t \in \mathcal{I}$ . It remains to be shown that  $\mathcal{I}$  can be extended to  $[t_0, \infty)$ . Assume for the sake of contradiction that  $\mathcal{I}$  has to be bounded, i.e., the escape time  $t_1$  is finite. Equivalently, there exists  $t_1$  for which there does not exist  $t_2 > t_1$  such that  $z(t) \in \mathcal{D}$  for all  $t \in [t_1, t_2]$ . Substituting  $t = t_1$  into (32) yields  $\|z(t_1)\| \leq \sqrt{\frac{\lambda_2}{\lambda_1} \|z(t_0)\|^2 e^{-\frac{\lambda_3}{\lambda_2}(t_1-t_0)} + \frac{\lambda_2 c}{\lambda_1 \lambda_3} \left(1 - e^{-\frac{\lambda_3}{\lambda_2}(t_1-t_0)}\right)} < \sqrt{\frac{\lambda_2}{\lambda_1} \|z(t_0)\|^2 + \frac{\lambda_2 c}{\lambda_1 \lambda_3}}$ . Because  $z(t_0) \in \mathcal{S}$ , it follows that  $\|z(t_0)\|^2 \leq \frac{\lambda_1}{\lambda_2} \chi^2 - \frac{c}{\lambda_3}$ , implying  $\|z(t_1)\| < \chi$ , and hence  $z(t_1) \in \text{int}(\mathcal{D})$ . Therefore, by continuity of  $z$ , there exists  $t_2 > t_1$  such that  $z(t) \in \mathcal{D}$  for all  $t \in [t_1, t_2]$ , violating the assumption made for contradiction. Therefore, by contradiction,  $\mathcal{I}$  can be extended to the interval  $[t_0, \infty)$ . ■

<sup>4</sup>Continuous solutions exists over some time-interval for systems satisfying Caratheodory existence conditions. According to Caratheodory conditions for the system  $\dot{y} = f(y, t)$ ,  $f$  should be locally bounded, continuous in  $y$  for each fixed  $t$ , and measurable in  $t$  for each fixed  $y$  [33, Ch.2, Theorem 1.1]. The dynamics in  $\dot{z}$  satisfy the Caratheodory conditions.

**Remark 4.** Since  $\lambda_3 = \min\{\alpha_1, \frac{k_r}{2}, \alpha_2, \frac{k_f}{2}, \frac{k_{\hat{\theta}}}{4} + \frac{\beta_1}{2\lambda_{\Gamma,\max}}\}$ , the gains  $\alpha_1, \alpha_2, k_r, k_f$ , and  $k_{\hat{\theta}}$  can be selected to be sufficiently high such that  $\lambda_3 = \frac{k_{\hat{\theta}}}{4} + \frac{\beta_1}{2\lambda_{\Gamma,\max}}$ . Since  $\beta_1$  is positive under the PE condition as mentioned in Remark 3, a larger value for  $\lambda_3$  is obtained, which implies faster exponential convergence to a smaller neighborhood of the origin. When the PE condition does not hold, convergence is not guaranteed; however, the gain  $k_{\hat{\theta}}$ , which is based on the sigma modification technique in [34, Sec. 8.4.1], ensures boundedness of all states. However, selecting a high value for  $k_{\hat{\theta}}$  can deteriorate tracking and parameter estimation performance since it yields a higher value for  $c$ .

## V. SIMULATIONS

To demonstrate the performance of the developed method, comparative simulations are performed on two different systems, i.e. a two-link manipulator and a UUV.<sup>5</sup>

### A. Two Link Manipulator

To demonstrate the performance of the developed composite adaptive Lb-DNN, comparative simulations are performed on the two-link robot manipulator model (see Appendix IX.1.1 for the dynamics) in [35] for 100 seconds. Baseline methods used for comparison include DNN-based adaptive controller with tracking error-based adaptation law developed in [16], an observer-based disturbance rejection controller [36] (i.e.,  $u = g^+(x, \dot{x}) (\ddot{x}_d - (\alpha_1 + k_r) r + (\alpha_1^2 - 1) e - \dot{f})$ ), a nonlinear proportional-derivative (PD) controller  $u = g^+(x, \dot{x}) (\ddot{x}_d - (\alpha_1 + k_r) r + (\alpha_1^2 - 1) e)$ , and nonlinear model predictive control (MPC) (see Appendix IX.1.1 for more details on baseline control methods). The comparative simulation is performed using a fully-connected DNN with 5 hidden layers and 5 neurons in each layer with hyperbolic tangent activation functions. The DNN weights are initialized randomly from the distribution  $U(-0.5, 0.5)$ . For a realistic simulation, an additive white Gaussian (AWG) measurement noise with a signal-to-noise ratio of 50 dB is considered in all state measurements.

To evaluate the tracking and drift compensation performance of the developed and baseline methods, the root mean square (RMS) values of the tracking error norm, denoted by  $\|e\|_{\text{RMS}}$ , and function approximation error along the trajectory are calculated in the steady state (i.e., in the interval [50, 100] seconds). The corresponding values are provided in Table I. Since the trajectory explored by the system essentially acts as a training dataset for the DNNs, the RMS function approximation error does not indicate whether the DNN model is overfit and how well the model generalizes over unexplored data. Thus, to evaluate the performance of the DNN beyond the trajectory, a test dataset involving 100 random datapoints

<sup>5</sup>Codes are available at: <https://github.com/patilomkarsudhir/Composite-Adaptive-Lyapunov-Based-Deep-Neural-Network>

Table I  
ROBOT MANIPULATOR PERFORMANCE COMPARISON

Adaptation Law	$\ e\ _{\text{RMS}}$ (deg)	$\ u\ _{\text{RMS}}$ (Nm)	Function error on-trajectory (rad/s <sup>2</sup> )	Mean function error on test data (rad/s <sup>2</sup> )
Tracking Error-Based	0.629	10.100	0.430	1.215
Composite	0.308	7.962	0.131	0.260
Observer-Based	0.310	10.612	0.204	N/A
Nonlinear PD	3.142	6.642	N/A	N/A
Nonlinear MPC	1.101	8.275	N/A	N/A

with values selected from the distribution  $U(-0.25, 0.25)$  is constructed, and the mean  $\|f(x, \dot{x}) - \Phi(X, \hat{\theta})\|$  across all points in the dataset is evaluated. The value of the mean  $\|f(x, \dot{x}) - \Phi(X, \hat{\theta})\|$  on the test dataset at the end of each simulation (i.e., at  $t = 100$  seconds) is then used as a metric in Table I for comparing the generalization performance of each method. To evaluate the control effort required by each controller throughout the transient and steady states, the RMS values of the control input norm,  $\|u\|$ , are calculated in the time-interval [0,100] seconds and provided in Table I for each method. As evident from Table I and Figure 1, the developed composite adaptive Lb-DNN significantly improved the tracking performance compared to tracking error-based adaptive Lb-DNN, nonlinear PD, and nonlinear MPC with comparable control effort and approximately 50%, 90%, and 70% improvements in  $\|e\|_{\text{RMS}}$ , respectively. The tracking error-based adaptive Lb-DNN exhibited more chattering in the control input due to measurement noise, which might be because the update law involved a constant high adaptation gain. In contrast, the composite update law has a decreasing gain due to the least squares approach which mitigates noise amplification resulting from the adaptation gain. Additionally, the tracking performance with the observer-based disturbance rejection method is comparable to the developed method, which is expected because the developed method also used the observer-based estimate  $\hat{f}$  to formulate the prediction error. However, notice the increased control effort due to large overshoots in the controller resulting from high gains in the state-derivative observer in (12). Although the developed composite adaptation law also used the state-derivative estimates generated by the high-gain observer, the state-derivative estimates are not directly used in the control input. Using the state-derivative estimates in the adaptation law did not cause as large overshoots in the control input because the adaptation law involves an integrator that effectively acts as a low pass filter on any overshoots in the state-derivative estimates. Furthermore, note that the observer-based controller only provides instantaneous estimates of  $f$ , due to the lack of a model. Hence, it cannot be generalized for off-trajectory points, thus not achieving the system identification objective. Additionally, despite the fact that nonlinear MPC uses model knowledge, the developed method achieved improved tracking

with reduced control effort compared to nonlinear MPC.

The evolution of the mean function approximation error on the aforementioned test dataset is shown on the right in Figure 1. The mean function approximation error with the composite method on the test dataset initially overshot followed by oscillatory behavior during the initial 10 seconds. Such a behavior is expected since the combined system goes through the initial transients, and the online data in the first few seconds based on which the DNN has learned is limited. However, after 10 s, the composite adaptation law exhibited a consistent decrease in the mean function approximation error, unlike the tracking error-based adaptation law. As a result, the composite adaptation law achieved 72.04% improvement in the final value of mean function approximation error.

1) *Ablation Study*: An ablation study is performed to demonstrate the performance of the developed method for various DNN architectures mentioned in Table II. The same set of gains are used and the weights are randomly initialized from the distribution  $U(-0.5, 0.5)$ . As evident from the percentage decrease in Table II, the developed composite adaptation law significantly improves the tracking, drift compensation, and generalization performance of the DNN across all DNN architecture with a comparable control effort. Notably, although all DNN architectures yielded acceptable performance (i.e., with  $\|e\|_{\text{RMS}}$  less than 0.5 deg) with the developed composite adaptive Lb-DNN controller, no conclusive trend was obtained to comment on the selection of appropriate size for the DNN for this application. Importantly, using DNN of a greater size did not affect the control effort.

### B. Unmanned Underwater Vehicle

Comparative simulation results are also provided for the UUV system (see Appendix IX.2 for dynamics) from [37] using the composite adaptive Lb-DNN under intermittent loss of feedback, with the same baselines as in Subsection V-A. Since the DNN identifies the system dynamics, the identified DNN could be used to predict the uncertainty when the state feedback is intermittently lost. Let  $i \in \mathbb{Z}_{\geq 0}$  denote the time index such that the state feedback is available in the time interval  $[t_{2i}, t_{2i+1})$  and unavailable in the time interval  $[t_{2i+1}, t_{2i+2})$  for all  $i \in \mathbb{Z}_{\geq 0}$ . During the time interval  $[t_{2i}, t_{2i+1})$ , when the feedback is available, the control and adaptation laws in (10) and (15) are used for all  $i \in \mathbb{Z}_{\geq 0}$ .

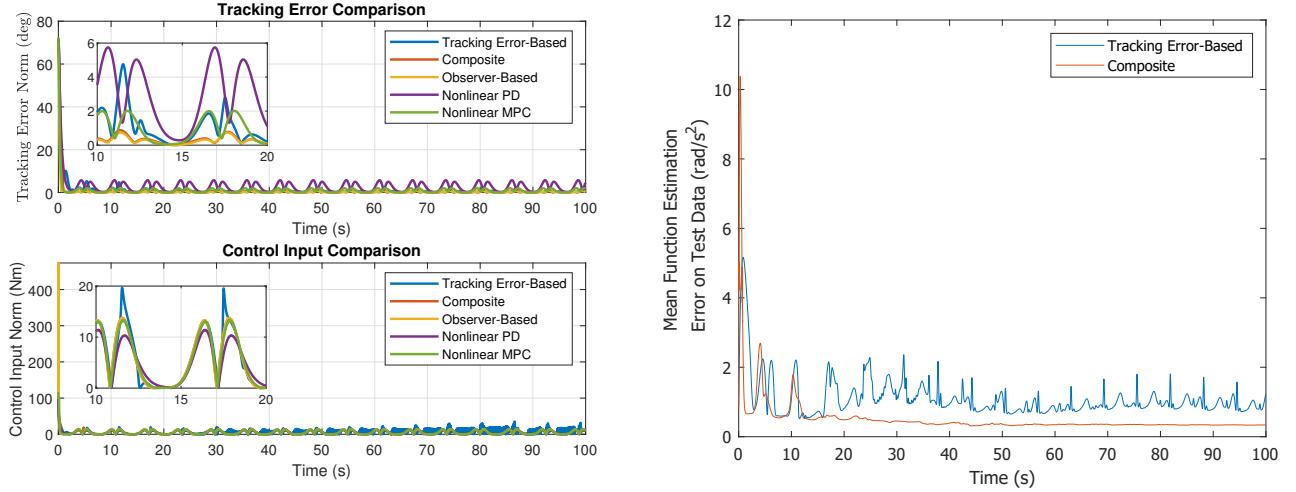


Figure 1. Left: Comparative plots of the tracking error norm and control input norm along the trajectory with the developed and baseline controllers. A zoomed view during the time interval  $[10, 20]$  is added in each subplot for visual clarity. Right: Comparative plots of the mean of function estimation error norm  $\|f(x, \dot{x}) - \Phi(X, \dot{\theta})\|$  using tracking error-based adaptation and composite adaptation on the test dataset.

Table II  
PERFORMANCE COMPARISON

Architecture	Adaptation Law	$\ e\ _{RMS}$ (deg)	$\ u\ _{RMS}$ (Nm)	Function error on-trajectory (rad/s <sup>2</sup> )	Final mean function error on test data (rad/s <sup>2</sup> )
Layers	Neurons				
3	3	Tracking Error-Based	0.731	7.484	0.430
		Composite	0.315	7.944	0.138
		% Decrease	56.97	-6.14	67.77
4	3	Tracking Error-Based	3.338	7.021	1.636
		Composite	0.338	7.957	0.154
		% Decrease	89.87	-13.33	90.58
4	4	Tracking Error-Based	0.685	7.731	0.426
		Composite	0.309	7.957	0.132
		% Decrease	54.85	-2.87	68.93
5	5	Tracking Error-Based	0.664	7.800	0.395
		Composite	0.307	7.955	0.131
		% Decrease	53.69	-1.99	66.87
5	10	Tracking Error-Based	0.351	7.940	0.192
		Composite	0.308	7.959	0.130
		% Decrease	12.41	-0.235	32.13
10	10	Tracking Error-Based	0.584	8.826	1.330
		Composite	0.307	7.965	0.130
		% Decrease	47.34	9.75	90.22

Table III  
UUV PERFORMANCE COMPARISON

	e-based	Composite	Observer-based	NMPC	NPD
RMS position tracking error norm (m)	0.201	0.152	0.158	0.254	0.186
RMS angular tracking error norm (rad)	0.037	0.012	0.024	0.054	0.028
RMS linear control input norm (N)	0.069	0.065	0.126	0.128	0.067
RMS angular control input norm (Nm)	0.041	0.035	0.072	0.036	0.032
RMS linear dynamics estimation error norm (m/s <sup>2</sup> )	4.370	2.585	19.423	N/A	N/A
RMS angular dynamics estimation error norm (rad/s <sup>2</sup> )	2.058	1.408	3.021	N/A	N/A

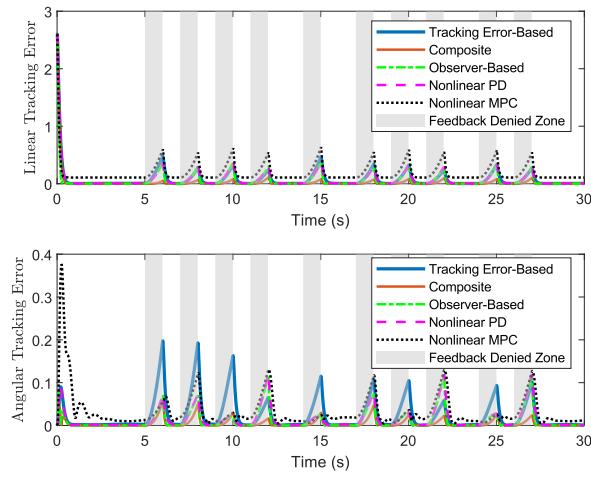


Figure 2. Comparative plots of the linear tracking error norm (m) and angular tracking error norm (rad) for the UUV. The time intervals corresponding to the feedback denied zones are marked in grey patches.

However, during the time interval  $[t_{2i+1}, t_{2i+2})$  when the state feedback is unavailable, an open-loop controller is developed as  $u = g^+(x_d(t), \dot{x}_d(t))(\ddot{x}_d(t) - \Phi(X_d(t), \hat{\theta}(t_{2i+1})))$ . The reader is referred to Appendix X for sufficient dwell-time conditions and stability analysis under which the system can stably operate under intermittent loss of feedback. For both the composite and tracking error-based adaptive Lb-DNN methods, a fully-connected DNN with 5 hidden layers with 5 neurons in each layer with hyperbolic tangent activation function was used. During feedback unavailability, the observer-based disturbance rejection and nonlinear PD controllers were designed to be  $u = g^+(x_d(t), \dot{x}_d(t))\ddot{x}_d(t)$ , with  $\dot{x}, \dot{f} = 0$  for the observer-based method, and the nonlinear MPC was designed using model predictions propagated forward over the horizon treating  $x_d$  as the current state. To simulate the performance of the system under intermittent loss of feedback, each simulation was performed for 30 seconds where the feedback was made unavailable for the time intervals [5, 6], [7, 8), [9, 10), [11, 12), [14, 15), [17, 18), [19, 20), [21, 22), [24, 25), and [26, 27) seconds, respectively. For a realistic simulation, an AWG measurement noise with a signal-to-noise ratio of 50 dB is considered in all state measurements.

Figure 2 shows comparative plots of linear tracking error and angular tracking error norms. The developed method outperforms the baseline methods in tracking the reference trajectory, especially during the loss of feedback as also evident from Table III. Additionally, Figure 3 shows the comparative plot of the linear and angular dynamics (function) estimation error norms on the left and control input norms on the right. Since tracking error-based adaptation does not involve guarantees on parameter estimation, the resulting predictions quickly diverge during absence of feedback due to model identification

errors. However, the composite adaptive Lb-DNN controller can better predict and compensate for the drift dynamics under the absence of state feedback. Additionally, when feedback is available, the state-derivative observer-based approach can yield a tracking performance comparable to the DNN-based controllers since it essentially involves a robust high-gain approach. However, the tracking performance degrades significantly in the absence of feedback using the observer-based approach when compared to the composite adaptive Lb-DNN controller. Furthermore, the observer-based approach requires significantly higher control effort as compared to both of the DNN-based adaptive controllers due to reasons discussed in Subsection V-A. Additionally, despite the fact that nonlinear MPC uses model knowledge, the developed method achieved improved tracking with reduced control effort compared to nonlinear MPC.

## VI. CONCLUSION

A composite adaptive Lb-DNN is developed for simultaneous online system identification and control, using the Jacobian of the DNN, the tracking error, and a prediction error based on a novel formulation using a dynamic state-derivative observer. A Lyapunov-based stability analysis guarantees the tracking, observer, and parameter estimation errors are UUB, with tighter bounds on these errors when the DNN's Jacobian satisfies the PE condition. Comparative simulation results demonstrate a significant improvement in tracking, function estimation and generalization capabilities with the developed method in comparison to the tracking error-based Lb-DNN in [16] and observer-based disturbance rejection controller as baseline methods.

## VII. LIMITATIONS AND SCOPE FOR FUTURE WORK

The persistence of excitation condition for identifying the parameters is restrictive as there are challenges in verifying it online. For linear regression, recent developments in the adaptive control literature such as [38]–[41] provide parameter estimation methods that guarantee parameter convergence under excitation conditions weaker than PE. All of these methods involve some form of regression extension by storing the history of the regression in memory over an interval of time. However, these methods are restricted to linear regression and have not been explored for NIP models such as DNNs yet. Thus, insights from this paper may be used in future work to develop adaptation laws for DNNs that yield parameter estimation guarantees under excitation conditions weaker than PE. Furthermore, extensions of the developed online system identification approach in optimization-based control paradigms such as MPC and reinforcement learning can be explored. Moreover, future research efforts can also investigate how to combine the developed method with control barrier functions to satisfy state and input constraints.

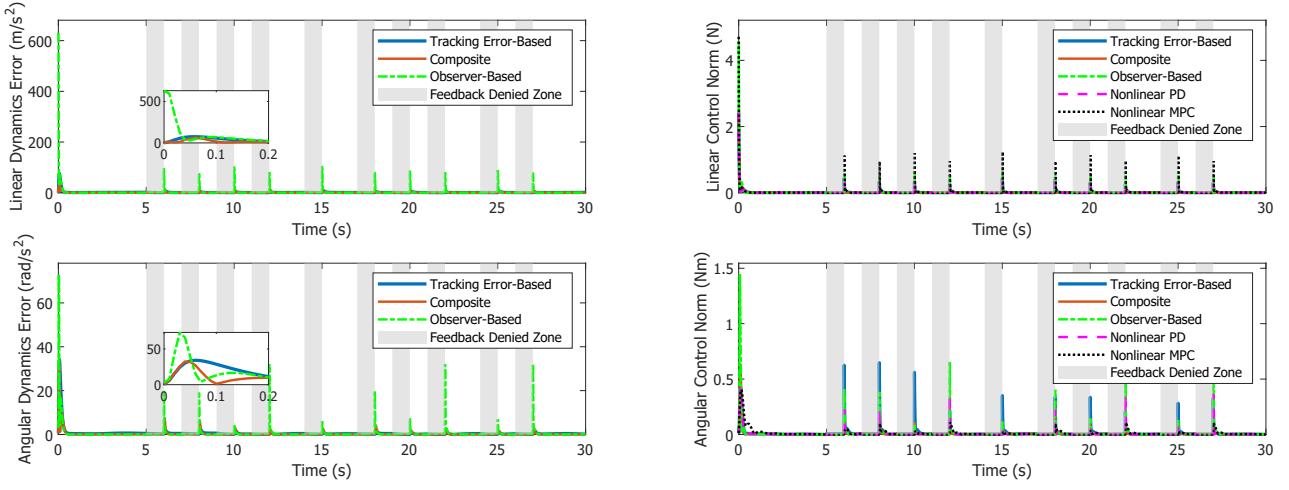


Figure 3. Left: Comparative plots of the estimation error norm (i.e.,  $\|f(x, \dot{x}) - \Phi(X, \hat{\theta})\|$  during feedback availability and  $\|f(x, \dot{x}) - \Phi(X_d(t), \hat{\theta})\|$  during loss of feedback) for the UUV with the developed and baseline methods. Right: Comparative plots of the linear and angular control input norms for the UUV. Zoomed views during the time interval [0, 0.2] seconds are added for visual clarity.

## REFERENCES

- [1] G. Shi, X. Shi, M. OConnell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural lander: Stable drone landing control using learned dynamics," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 9784–9790, 2019.
- [2] A. Punjani and P. Abbeel, "Deep learning helicopter dynamics models," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 3223–3230, 2015.
- [3] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in *Proc. IEEE Conf. Decis. Control*, pp. 4653–4660, 2016.
- [4] Q. Li, J. Qian, Z. Zhu, X. Bao, M. K. Helwa, and A. P. Schoellig, "Deep neural networks for improved, impromptu trajectory tracking of quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 5183–5189, 2017.
- [5] S. Zhou, M. K. Helwa, and A. P. Schoellig, "Design of deep neural networks as add-on blocks for improving impromptu trajectory tracking," in *Proc. IEEE Conf. Decis. Cont.*, pp. 5201–5207, IEEE, 2017.
- [6] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *Int. J. Robot. Research*, vol. 29, no. 13, pp. 1608–1639, 2010.
- [7] B. Karg and S. Lucia, "Efficient representation and approximation of model predictive control laws via deep learning," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3866–3878, 2020.
- [8] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*. Upper Saddle River, NJ: Prentice-Hall, 1989.
- [9] K. Nar and S. S. Sastry, "Persistency of excitation for robustness of neural networks," *arXiv preprint arXiv:1911.01043*, 2019.
- [10] K. Nar and S. S. Sastry, "Richness of training data does not suffice: Robustness of neural networks requires richness of hidden-layer activations," in *Proc. Workshop on Uncertainty and Robustness in Deep Learning (UDL), Int. Conf. on Machine Learning (ICML)*, 2020.
- [11] K. Sridhar, O. Sokolsky, I. Lee, and J. Weimer, "Improving neural network robustness via persistency of excitation," in *Proc. American Control Conf.*, pp. 1521–1526, IEEE, 2022.
- [12] A. Lamperski, "Neural network independence properties with applications to adaptive control," in *Proc. IEEE Conf. Decision and Control*, pp. 3365–3370, IEEE, 2022.
- [13] G. Joshi, J. Virdi, and G. Chowdhary, "Asynchronous deep model reference adaptive control," *Proc. PMLR Conf. Robot Learn.*, pp. 4601–4608, November 2020.
- [14] 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.
- [15] D. Le, M. Greene, W. 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.
- [16] 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.
- [17] D. M. Le, O. S. Patil, C. Nino, and W. E. Dixon, "Accelerated gradient approach for neural network-based adaptive control of nonlinear systems," in *Proc. IEEE Conf. Decis. Control*, pp. 3475–3480, 2022.
- [18] O. Patil, D. Le, E. Griffis, and W. E. Dixon, "Lyapunov-based deep residual neural network (ResNet) adaptive control," *IEEE Access*, vol. 13, pp. 117943–117952, 2025.
- [19] 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.
- [20] 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.
- [21] D. Muthirayan and P. P. Khargonekar, "Memory augmented neural network adaptive controllers: Performance and stability," *IEEE Trans. Autom. Contr.*, vol. 68, no. 2, pp. 825–838, 2023.
- [22] H.-Y. Chen, Z. Bell, P. Deptula, and W. E. Dixon, "A switched systems approach to path following with intermittent state feedback," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 725–733, 2019.
- [23] 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," *arXiv preprint arXiv:2402.00671*, 2024.
- [24] Z. Bell, R. Sun, K. Volle, P. Ganesh, S. Nivison, and W. E. Dixon, "Target tracking subject to intermittent measurements using attention deep neural networks," *IEEE Control Syst. Lett.*, vol. 7, pp. 379–384, 2023.
- [25] J. J. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, pp. 509–519, July 1989.
- [26] P. Kidger and T. Lyons, "Universal approximation with deep narrow networks," in *Conf. Learn. Theory*, pp. 2306–2327, 2020.

- [27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, vol. 1. MIT Press, 2016.
- [28] R. G. Hart, O. S. Patil, Z. I. Bell, and W. E. Dixon, “System identification and control using lyapunov-based deep neural networks without persistent excitation: A concurrent learning approach,” *arXiv preprint arXiv:2505.10678*, 2025.
- [29] 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, pp. 3594–3601, July 2017.
- [30] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*. New York: John Wiley & Sons, 1995.
- [31] W. Rudin, *Principles of Mathematical Analysis*. McGraw-Hill, 1976.
- [32] O. S. Patil, B. C. Fallin, C. F. Nino, R. G. Hart, and W. E. Dixon, “Bounds on deep neural network partial derivatives with respect to parameters,” *arXiv preprint arXiv:2503.21007*, 2025.
- [33] E. A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations*. McGraw-Hill, 1955.
- [34] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, 1996.
- [35] M. de Queiroz, J. Hu, D. Dawson, T. Burg, and S. Donepudi, “Adaptive position/force control of robot manipulators without velocity measurements: Theory and experimentation,” *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 27-B, no. 5, pp. 796–809, 1997.
- [36] J. Han, “From PID to active disturbance rejection control,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 900–906, 2009.
- [37] N. Fischer, D. Hughes, P. Walters, E. Schwartz, and W. E. Dixon, “Nonlinear RISE-based control of an autonomous underwater vehicle,” *IEEE Trans. Robot.*, vol. 30, pp. 845–852, Aug. 2014.
- [38] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, “Concurrent learning adaptive control of linear systems with exponentially convergent bounds,” *Int. J. Adapt. Control Signal Process.*, vol. 27, no. 4, pp. 280–301, 2013.
- [39] 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, pp. 283–290, Jan. 2018.
- [40] Y. Pan and H. Yu, “Composite learning robot control with guaranteed parameter convergence,” *Automatica*, vol. 89, pp. 415–419, Mar. 2018.
- [41] 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, 2021.
- [42] F. L. Lewis, “Neural network control of robot manipulators,” *IEEE Expert*, vol. 11, no. 3, pp. 64–75, 1996.
- [43] 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, 1996.
- [44] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*. Boston, MA: Kluwer Academic Publishers, 2002.
- [45] O. S. Patil, D. M. Le, E. Griffis, and W. E. Dixon, “Deep residual neural network (ResNet)-based adaptive control: A Lyapunov-based approach,” in *Proc. IEEE Conf. Decis. Control*, pp. 3487–3492, 2022.
- [46] P. Patre, W. Mackunis, M. Johnson, and W. E. Dixon, “Composite adaptive control for Euler-Lagrange systems with additive disturbances,” *Automatica*, vol. 46, no. 1, pp. 140–147, 2010.
- [47] P. 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, 2010.
- [48] M. O’Connell, G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, “Neural-fly enables rapid learning for agile flight in strong winds,” *Sci. Robotics*, vol. 7, no. 66, p. eabm6597, 2022.
- [49] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhauser: Boston, 2003.

## APPENDICES

### VIII. RELATED WORK

#### A. On Neural Network-Based Adaptive Control

Classical results [42]–[44] develop adaptive controllers for neural networks with a single hidden layer, where online up-

dates are performed for the input and output layer weights. In recent results [13]–[15], adaptive controllers were developed for DNNs. In these results, the outer-layer weights of the DNN are updated in real-time using Lyapunov-based adaptation laws, whereas the inner-layer weights are updated either using iterative batch updates on discrete intervals of time [13]–[15], or using a modular design [15]. Since the inner-layer weight updates in [13]–[15] happen using batch updates, the updates are essentially performed offline. In [15], the weights are updated online but the update laws are selected arbitrarily and not by a stability-driven approach. In [16], Lyapunov-based adaptation laws are developed for all layers of a fully-connected DNN (i.e., so-called Lb-DNN methods). Since the control and adaptation laws are derived from a Lyapunov-based stability analysis, the development is guaranteed to ensure stability of the closed-loop system. More recent Lb-DNN results develop Lyapunov-based adaptation laws for more complex architectures, specifically, deep residual networks (ResNets) [45] and long short-term memory (LSTM) networks [19]. However, as stated in the manuscript, the updates are based solely on tracking error feedback and are primarily meant to achieve tracking error convergence. These results do not achieve guarantees on parameter estimation and system identification.

#### B. On Composite Adaptive Control

The classical result in [25] develops adaptive controllers with a composite adaptation law that includes both tracking and prediction errors for nonlinear systems with linear-in-parameters (LIP) uncertainties. The result in [25] constructs a form of the prediction error using the swapping technique (also known as input or torque filtering), where a low-pass filter is applied on both sides of the dynamics to eliminate the unknown state-derivative term. For a brief illustration of the swapping technique in [25], consider the system

$$\dot{x} = Y(x)\theta + u,$$

where  $Y(x)$  is the regressor,  $\theta$  is the vector of unknown parameters, and  $u$  is the control input. If the state-derivative  $\dot{x}$  could be measured, the system can be expressed in terms of the linear regression equation  $\dot{x} - u = Y(x)\theta$ . Then the corresponding prediction error with an adaptive estimate  $\hat{\theta}$  could be developed as  $\epsilon = \dot{x} - u - Y(x)\hat{\theta}$ , which can be expressed linearly in terms of the parameter estimation error  $\tilde{\theta} = \theta - \hat{\theta}$  as  $\epsilon = Y(x)\tilde{\theta}$ . However,  $\dot{x}$  measurements are typically either unavailable or extremely noisy. To avoid using state-derivative information, [25] applied a low-pass filter on both sides of the dynamics, which results in the filtered regression

$$e^{-\beta t} * (\dot{x} - u) = (e^{-\beta t} * Y(x))\theta,$$

where  $*$  denotes the convolutional integral operation (i.e.,  $a(t) * b(t) = \int_0^t a(t-\tau)b(\tau)d\tau$ ) and  $e^{-\beta t}$  is the impulse

response of the low-pass filter with a positive constant decay rate  $\beta$ . Since  $e^{-\beta t} * \dot{x} = x(t) - x(0)e^{-\beta t} + \beta e^{-\beta t} * x$ , the filtered regression can be expressed as

$$x(t) - x(0)e^{-\beta t} + \beta e^{-\beta t} * x - e^{-\beta t} * u = (e^{-\beta t} * Y(x)) \theta,$$

which is implementable without using state-derivative information. The prediction error for the filtered regression can be developed as

$$\begin{aligned} \epsilon &= x(t) - x(0)e^{-\beta t} + \beta e^{-\beta t} * x \\ &\quad - e^{-\beta t} * u - (e^{-\beta t} * Y(x)) \hat{\theta} \\ &= (e^{-\beta t} * Y(x)) \theta - (e^{-\beta t} * Y(x)) \hat{\theta} \\ &= (e^{-\beta t} * Y(x)) \tilde{\theta}. \\ &= Y_f \tilde{\theta}, \end{aligned}$$

where  $Y_f = (e^{-\beta t} * Y(x))$ . Since  $\epsilon$  is linear in  $\tilde{\theta}$ , a composite adaptation law can be developed with a  $Y_f^\top \epsilon$  term which would yield negative  $\tilde{\theta}$  terms in the corresponding Lyapunov-based stability analysis. However, yielding this form of  $\epsilon$  using a filtered regression was possible because the uncertainty  $Y(x)\theta$  is linear in terms of  $\theta$ , which allowed  $\theta$  to be separable from  $e^{-\beta t} * Y(x)$  in the filtered regression. If  $Y(x)\theta$  is replaced by terms nonlinear in  $\theta$ , such as the DNN-based approximation  $\Phi(x, \theta) + \varepsilon(x)$ , applying a low pass filter on both sides would yield

$$e^{-\beta t} * (\dot{x} - u) = e^{-\beta t} * (\Phi(x, \theta) + \varepsilon(x)).$$

Notice that  $\theta$  is not separable from the convolutional integral in the term  $e^{-\beta t} * \Phi(x, \theta)$  since  $\Phi$  is nonlinear in  $\theta$ . As a result, the swapping technique from [25] does not apply for nonlinear in parameter uncertainties such as DNNs.

Results in [46] introduce a robust integral of the sign of the error (RISE)-based swapping technique to formulate the prediction error and design composite adaptive controllers for LIP uncertainties with additive disturbances. The RISE-based swapping technique is extended in [47] for NN-based models, but the development is restricted to single-hidden-layer NNs. Extending this for DNNs is mathematically challenging due to their nested NIP structure. Moreover, using RISE-based swapping requires additional RISE-based terms in the control input, which can debilitate the learning performance of the adaptive feedforward term. Notably, the results in [46] and [47] only ensure asymptotic tracking error convergence, and no guarantees are provided on the parameter estimates under the persistence of excitation (PE) condition.

The recent result in [48] developed a new learning representation uncertainties involving a composited disturbance given by  $f(x, \dot{x}, w) = \phi(x, \dot{x}) a(w)$ , where  $\phi(\cdot)$  denotes a basis function that is learned using a DNN and  $a(w)$  denotes a set of linear parameters accounting for an unknown disturbance time-varying disturbance  $w$ . Since  $f$  is linear in terms of  $a$ , the composite adaptive approach from [25] is used to design an adaptation law  $\hat{a}$  to update the estimates of  $a$  given

by  $\hat{a}$ . To obtain a disturbance-invariant representation of  $\phi$  using DNNs, a domain adversarially invariant meta-learning (DAIML) algorithm is developed to train the DNN offline. To the best of our knowledge, this is the only existing work using a composite adaptive approach in the context of deep learning-based control. However, since the DNN learning  $\phi(x, \dot{x})$  has an NIP structure, the aforementioned challenges apply for constructing a Lyapunov-based online adaptation law.

## IX. MORE SIMULATION RESULT DETAILS

All simulations were performed in MATLAB on a desktop with 64 GB RAM and 13th Gen Intel Core i9-13900 @2.00 GHz processor.

### A. Two Link Manipulator

1) *Dynamic Model:* The two-link robot manipulator was modeled by the uncertain Euler-Lagrange dynamics

$$M(x) \ddot{x} + C(x, \dot{x}) \dot{x} + F \dot{x} = u, \quad (33)$$

where  $x \triangleq [x_1, x_2]^\top \in \mathbb{R}^2$ ,  $\dot{x} \in \mathbb{R}^2$ , and  $\ddot{x} \in \mathbb{R}^2$  denote the vector of angular position, velocity, and acceleration of joints, respectively,  $M(x) \in \mathbb{R}^{2 \times 2}$  represents the inertia matrix,  $C(x, \dot{x}) \in \mathbb{R}^{2 \times 2}$  represents the centripetal-Coriolis matrix,  $F \in \mathbb{R}^{2 \times 2}$  represents friction effects, and  $u \in \mathbb{R}^2$  denotes the torque inputs. In (33), the dynamics were modeled as [35]

$$M(x) = \begin{bmatrix} p_1 + 2p_3 c_2, & p_2 + p_3 c_2 \\ p_2 + p_3 c_2, & p_2 \end{bmatrix}, \quad (34)$$

$$C(x, \dot{x}) = \begin{bmatrix} -p_3 s_2 \dot{x}_2, & -p_3 s_2 (\dot{x}_1 + \dot{x}_2) \\ p_3 s_2 \dot{x}_1, & 0 \end{bmatrix}, \quad (35)$$

$$F = \begin{bmatrix} f_1, & 0 \\ 0, & f_2 \end{bmatrix}, \quad (36)$$

where the short-hand notations  $c_2$  and  $s_2$  are defined as  $c_2 \triangleq \cos(x_2)$  and  $s_2 \triangleq \sin(x_2)$ , respectively. The nominal parameters of the two-link robot model in (34)–(36) were  $p_1 = 3.473 \text{ kg} \cdot \text{m}^2$ ,  $p_2 = 0.196 \text{ kg} \cdot \text{m}^2$ ,  $p_3 = 0.242 \text{ kg} \cdot \text{m}^2$ ,  $f_1 = 5.3 \text{ Nm} \cdot \text{sec}$ , and  $f_2 = 1.1 \text{ Nm} \cdot \text{sec}$ . The two-link manipulator dynamics can be expressed using Eq. (1) from the manuscript with  $f(x, \dot{x}) = -M^{-1}(x)(C(x, \dot{x}) \dot{x} + F \dot{x})$  and  $g(x, \dot{x}) = M^{-1}(x)$ . The gains are selected as  $\alpha_1 = 5$ ,  $\alpha_2 = 10$ ,  $\alpha_3 = 20$ ,  $\Gamma(0) = I$ ,  $k_r = 5$ ,  $k_f = 10$ ,  $k_{\hat{\theta}} = 0.0001$ ,  $\beta_0 = 10$ , and  $\varkappa_0 = 2$ . The states are initialized as  $x(0) = [1, -1]^\top \text{ rad}$  and  $\dot{x}(0) = [0, 0]^\top \text{ rad/s}$ , the initial parameter estimate  $\hat{\theta}(0)$  is selected from the uniform distribution  $U(-0.5, 0.5)$ , and the desired trajectory is  $x_d = 0.25 \exp(-\sin(t))[\sin(t), \cos(t)]^\top \text{ rad}$ . The weights are randomly initialized from the distribution  $U(-0.5, 0.5)$ . Baseline methods used for comparison include DNN-based adaptive controller with tracking error-based adaptation law developed in [16], an observer-based disturbance rejection controller [36] (i.e.,  $u = g^+(x, \dot{x}) (\ddot{x}_d - (\alpha_1 + k_r)r + (\alpha_1^2 - 1)e - \hat{f})$ ),

a nonlinear proportional-derivative (PD) controller  $u = g^+(x, \dot{x}) (\ddot{x}_d - (\alpha_1 + k_r) r + (\alpha_1^2 - 1) e)$ , and nonlinear model predictive control (MPC). The baseline DNN-based adaptive controller uses the tracking error-based adaptation law given by [16]

$$\dot{\hat{\theta}} = \text{proj} \left( -k_{\hat{\theta}} \Gamma(t_0) \hat{\theta} + \Gamma(t_0) \Phi'^\top (X, \hat{\theta}) r \right)$$

with a constant  $\Gamma$  (unlike the developed method which uses a time-varying  $\Gamma$ ), where it was selected as  $\Gamma = I$ . For a fair comparison, the set of gains common to the developed and baseline methods were selected to be exactly the same. The nonlinear MPC was designed to minimize the cost

$$\begin{aligned} J(e(t_k), r(t_k), u(t_k)) &= \sum_{i=1}^N (e(t_{k+i})^\top Q_e e(t_{k+i}) \\ &\quad + r(t_{k+i})^\top Q_r r(t_{k+i}) \\ &\quad + u(t_{k+i})^\top R u(t_{k+i})) \end{aligned}$$

subjected to the model dynamics discretized using Euler's method with a step size of 0.01 seconds. The controller was implemented using MATLAB's fmincon optimizer with  $Q_e = I$ ,  $Q_r = I$ ,  $R = 0.0001I$  and a prediction horizon of  $N = 5$  steps and bounded control input search space with upper and lower bounds of 50 and -50, respectively, for every control input.

### B. UUV System

The simulations were performed on an unmanned underwater vehicle (UUV) system that can be modeled as [37]

$$\begin{aligned} \ddot{x} &= -\bar{M}^{-1}(x) (\bar{C}(x, \dot{x}, \nu) \dot{x} + \bar{D}(x, \nu) \dot{x} + \bar{G}(x)) \\ &\quad + \bar{M}^{-1}(x) \tau_n, \end{aligned} \quad (37)$$

where  $x \in \mathbb{R}^6$  denotes a vector of position and orientation with coordinates in the earth-fixed frame,  $\dot{x} \in \mathbb{R}^6$  denotes a vector of linear and angular velocities with coordinates in the earth-fixed frame, and  $\nu \in \mathbb{R}^6$  denotes a vector of linear and angular velocities with coordinates in the body-fixed frame. The inertial effects, centripetal-Coriolis effects, hydrodynamic damping effects, gravitational effects, and control input in the earth-fixed frame can be represented by  $\bar{M} : \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$ ,  $\bar{C} : \mathbb{R}^6 \times \mathbb{R}^6 \times \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$ ,  $\bar{D} : \mathbb{R}^6 \times \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$ ,  $\bar{G} : \mathbb{R}^6 \rightarrow \mathbb{R}^6$ , and  $\tau_n : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^6$ , respectively. The velocities in the body-fixed frame can be related to the velocities in the earth-fixed frame using the relation

$$\dot{x} = J(x) \nu, \quad (38)$$

where  $J : \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$  is a Jacobian transformation matrix relating the two frames [37, Equation (2)]. Thus, the dynamics in (37) can be represented using Eq. (1) from the manuscript with

$$f(x, \dot{x}) = -\bar{M}^{-1}(x) (\bar{C}(x, \dot{x}, \nu) \dot{x} + \bar{D}(x, \nu) \dot{x} + \bar{G}(x))$$

Table IV  
UUV SYSTEM PARAMETERS [49, EQUATION (2.247)]

$m_1 = 215 \text{ kg}$	$d_{11} = 70 \text{ Nm}\cdot\text{sec}$	$d_{41} = 30 \text{ Nm}\cdot\text{sec}$
$m_2 = 265 \text{ kg}$	$d_{12} = 100 \text{ N}\cdot\text{sec}^2$	$d_{42} = 50 \text{ N}\cdot\text{sec}^2$
$m_3 = 265 \text{ kg}$	$d_{21} = 100 \text{ Nm}\cdot\text{sec}$	$d_{51} = 50 \text{ Nm}\cdot\text{sec}$
$m_4 = 40 \text{ kg}\cdot\text{m}^2$	$d_{22} = 200 \text{ N}\cdot\text{sec}^2$	$d_{52} = 100 \text{ N}\cdot\text{sec}^2$
$m_5 = 80 \text{ kg}\cdot\text{m}^2$	$d_{31} = 200 \text{ Nm}\cdot\text{sec}$	$d_{61} = 50 \text{ Nm}\cdot\text{sec}$
$m_6 = 80 \text{ kg}\cdot\text{m}^2$	$d_{32} = 50 \text{ N}\cdot\text{sec}^2$	$d_{62} = 100 \text{ N}\cdot\text{sec}^2$

and

$$g(x, \dot{x}) = \bar{M}^{-1}(x).$$

Using the kinematic transformation in (38), the earth-fixed dynamics in (37) can be expressed using body-fixed dynamics as  $\bar{M} = J^{-\top} M J^{-1}$ ,  $\bar{C} = J^{-\top} [C(\nu) - M J^{-1} \dot{J}] J^{-1}$ ,  $\bar{D} = J^{-\top} D(\nu) J^{-1}$ ,  $\bar{G} = J^{-\top} G$ , and  $\tau_n = J^{-\top} \tau_b$ , where  $M \in \mathbb{R}^{6 \times 6}$ ,  $C : \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$ ,  $D : \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$ ,  $G : \mathbb{R}^6 \rightarrow \mathbb{R}^6$ , and  $\tau_b : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^6$  denote the inertial effects, centripetal-Coriolis effects, hydrodynamic damping effects, gravitational effects, and control input in the body-fixed frame, respectively. The inertial effects, centripetal-Coriolis effects, and hydrodynamic damping effects in the body-fixed effects can be expressed as [49, Equation (2.246)]

$$M = \text{diag} \{m_1, m_2, m_3, m_4, m_5, m_6\},$$

$$\begin{aligned} D &= \text{diag} \{d_{11} + d_{12} |\nu(1)|, d_{21} + d_{22} |\nu(2)|, d_{31} + d_{32} |\nu(3)|, \\ &\quad d_{41} + d_{42} |\nu(4)|, d_{51} + d_{52} |\nu(5)|, d_{61} + d_{62} |\nu(6)|\}, \end{aligned}$$

$$V_m = \begin{bmatrix} 0 & 0 & 0 & 0 & m_3 \nu_3 & -m_2 \nu_2 \\ 0 & 0 & 0 & -m_3 \nu_3 & 0 & m_1 \nu_1 \\ 0 & 0 & 0 & m_2 \nu_2 & -m_1 \nu_1 & 0 \\ 0 & m_3 \nu_3 & -m_2 \nu_2 & 0 & m_6 \nu_6 & -m_5 \nu_5 \\ -m_3 \nu_3 & 0 & m_1 \nu_1 & -m_6 \nu_6 & 0 & m_4 \nu_4 \\ m_2 \nu_2 & -m_1 \nu_1 & 0 & m_5 \nu_5 & -m_4 \nu_4 & 0 \end{bmatrix},$$

where the numerical values of mass, inertia, and damping parameters listed in Table IV were used. The considered UUV is neutrally buoyant, thus  $G = 0_{6 \times 1}$ . The desired trajectory was selected as a helical trajectory given by  $x_d(t) = [2\cos(0.5t) \text{ m}, 2\sin(0.5t) \text{ m}, 0.1tm, 0\text{rad}, 0\text{rad}, -0.125t \text{ rad}]^\top$ , and the system was initialized with  $x(0) = [-0.5 \text{ m}, -0.5 \text{ m}, -0.5 \text{ m}, 0\text{rad}, 0\text{rad}, 0\text{rad}]^\top$  and  $\dot{x}(0) = [0_{1 \times 3} \text{ m/s}, 0_{1 \times 3} \text{ rad/s}]^\top$ . The following gains were used in the simulation:  $\alpha_1 = 5$ ,  $\alpha_2 = 10$ ,  $\alpha_3 = 40$ ,  $k_r = 20$ ,  $k_f = 20$ ,  $k_{\hat{\theta}} = 0.0001$ ,  $\Gamma(0) = 0.5I_{221}$ ,  $\beta = 10$ . The weights are randomly initialized from the distribution  $U(-0.5, 0.5)$ . Similar to the two-link manipulator, for a fair comparison, the set of gains common to the developed and baseline methods were selected to be exactly the same. The MPC was implemented in a similar manner as the two-link manipulator (see Appendix IX.1.1 for details), except with optimizer with  $Q_e = I$ ,  $Q_r = I$ ,  $R = 10I$ ,  $N = 5$  steps, and bounded control input search space with upper and lower bounds of 5 N and -5 N, respectively, for every linear control input, and 5 Nm and -5 Nm, respectively, for every angular control input, as these values were empirically found to yield the most desirable performance.

## X. DWELL-TIME CONDITIONS FOR STABLE OPERATION UNDER INTERMITTENT LOSS OF STATE FEEDBACK

Since the DNN identifies the system dynamics, the identified DNN estimates could be used to predict the uncertainty when the state feedback is intermittently lost. Let  $i \in \mathbb{Z}_{\geq 0}$  denote the time index such that the state feedback is available in the time interval  $[t_{2i}, t_{2i+1})$  and unavailable in the time interval  $[t_{2i+1}, t_{2i+2})$  for all  $i \in \mathbb{Z}_{\geq 0}$ . During the time interval  $[t_{2i}, t_{2i+1})$ , when the feedback is available, the developed composite adaptive control and adaptation laws developed in the manuscript are used for all  $i \in \mathbb{Z}_{\geq 0}$ . However, during the time interval  $[t_{2i+1}, t_{2i+2})$  when the state feedback is unavailable, the control input is designed to be an open-loop controller based on the last DNN weight estimate that was identified the feedback was available. The open-loop controller is given by

$$u = g^+(x_d(t), \dot{x}_d(t)) (\ddot{x}_d(t) - \Phi(X_d(t), \hat{\theta}(t_{2i+1}))) \quad (39)$$

Substituting (39) into  $\ddot{x} = f(x, \dot{x}) + g(x, \dot{x})u$ , subtracting  $\ddot{x}_d$  on both sides, adding and subtracting  $\Phi(X_d, \hat{\theta}(t_{2i+1}))$ , and rearranging terms yields

$$\begin{aligned} \ddot{e} &= \Phi(X, \theta^*) - \Phi(X_d(t), \hat{\theta}(t_{2i+1})) + \varepsilon(X) \\ &\quad + (g(x, \dot{x})g^+(x_d(t), \dot{x}_d(t)) - I_n)(\ddot{x}_d(t) - \Phi(X_d(t), \hat{\theta}(t_{2i+1}))) \end{aligned} \quad (40)$$

$$(41)$$

For the purpose of this section, it is assumed the drift dynamics  $f$  are globally Lipschitz with a Lipschitz constant  $\varpi \in \mathbb{R}_{>0}$ , and the control effectiveness and its pseudoinverse,  $g$  and  $g^+$ , are globally bounded functions without bounds  $\bar{g}, \bar{g}^+$  such that  $\|g(x, \dot{x})\| \leq \bar{g}$  and  $\|g^+(x, \dot{x})\| \leq \bar{g}^+$ . The global Lipschitzness of  $f$  is assumed in order to rule out the possibility of the drift dynamics causing finite-time escape during the absence of state-feedback. Such an assumption is reasonable since finite-time escape is usually not inherent to the uncontrolled dynamics for most practical systems of interest. Additionally, assuming that  $g$  and  $g^+$  are bounded is reasonable for most practical engineering systems, since the control effectiveness term usually results from the inertia matrix or the kinematic Jacobian of the system, and systems that may have potentially singular kinematic Jacobians in practice are not considered here. Additionally, in this section, a requirement is imposed on the selected DNN  $\Phi$  to contain bounded globally Lipschitz activation functions. Thus, using bounds on  $g, g^+, \Phi(X_d, \hat{\theta}(t_{2i+1})), \tilde{\theta}(t_{2i+1})$ , and  $\ddot{x}_d$ , it can be shown that there exists constants  $L_U, \delta_U \in \mathbb{R}_{>0}$  such that  $\|\ddot{e}\| \leq L_U \|e\| + L_U \|\dot{e}\| + \delta_U$ . Using the relations  $r = \dot{e} + \alpha_1 e$  and  $\ddot{e} = \dot{r} - \alpha_1 \dot{e}$  yields the inequality  $\|\dot{r}\| \leq (\alpha_1^2 + L_U \alpha_1 + L_U) \|e\| + (L_U + \alpha_1) \|r\| + \delta_U$ . Additionally, since  $f$  is considered to be globally Lipschitz in this section, it follows that  $\left\| \frac{\partial f}{\partial x} \right\|, \left\| \frac{\partial f}{\partial \dot{x}} \right\| \leq \varpi$ . As a result, it can be shown that  $\|\dot{f}\| \leq (2\alpha_1^2 + (L_U + 1)\alpha_1 + L_U) \varpi \|e\| +$

$(L_U + 2\alpha_1 + 1) \varpi \|r\| + (\delta_U + \bar{x}_d + \bar{\ddot{x}}_d) \varpi$ . During the loss of state feedback, all observer and adaptive update laws are selected to be zero, i.e.,  $\dot{r} = 0, \dot{\hat{f}} = 0$ , and  $\dot{\hat{\theta}} = 0$ .

The growth of the Lyapunov function in (25) is examined using the bounds on  $\dot{r}$  and  $\dot{\hat{f}}$  to analyze the growth of the error states during the loss of state feedback. By successive use of Holder's and Young's inequalities, it can be shown that  $\dot{V} \leq \lambda_U V + \Delta_U$ , when feedback is unavailable, where  $\lambda_U \triangleq 2 \max(\frac{3\alpha_1^2 + L_U \alpha_1 + L_U + 1}{2} + (2\alpha_1^2 + (L_U + 1)\alpha_1 + L_U)^2 \varpi^2 + (\alpha_1^2 + L_U \alpha_1 + L_U)^2, \frac{\alpha_1^2 + (L_U + 2)\alpha_1 + 3L_U + \delta_U + 1}{2} + (L_U + 2\alpha_1 + 1)^2 \varpi^2 + (L_U + \alpha_1)^2, \frac{1}{2})$  and  $\Delta_U \triangleq \frac{\delta_U}{2} + \delta_U^2 + (\delta_U + \bar{x}_d + \bar{\ddot{x}}_d)^2 \varpi^2$ . Solving for  $V$  for yields  $V(t) \leq V(t_{2i+1}) e^{\lambda_U(t-t_{2i+1})} + \frac{\Delta_U}{\lambda_U} (e^{\lambda_U(t-t_{2i+1})} - 1)$  for all  $(t, i) \in [t_{2i+1}, t_{2i+2}) \times \mathbb{Z}_{\geq 0}$ . Then applying the bounds in (26) and taking the square root yields

$$\|z(t)\| \leq \sqrt{\frac{\lambda_2}{\lambda_1} \|z(t_{2i+1})\|^2 e^{\lambda_U(t-t_{2i+1})} + \frac{2\Delta_U}{\lambda_1 \lambda_U} (e^{\lambda_U(t-t_{2i+1})} - 1)}, \quad (42)$$

for all  $(t, i) \in [t_{2i+1}, t_{2i+2}) \times \mathbb{Z}_{\geq 0}$ .

When the system regains feedback, the condition  $z(t_{2i+2}) \in \mathcal{S}$  needs to be satisfied for the composite adaptive Lb-DNN to yield the results in Theorem 1 of the manuscript. Imposing this condition yields the following condition for maximum dwell time during which feedback can be unavailable without affecting the UUB properties of the resulting switched system,

$$(t_{2i+2} - t_{2i+1}) \leq \frac{1}{\lambda_U} \ln \left( \frac{\frac{\lambda_1}{\lambda_2} \chi^2 + \frac{2\Delta_U}{\lambda_1 \lambda_U} - \frac{c}{\lambda_3}}{\frac{\lambda_2}{\lambda_1} \|z(t_{2i+1})\|^2 + \frac{2\Delta_U}{\lambda_1 \lambda_U}} \right), \quad (43)$$

for  $(t, i) \in [t_{2i+1}, t_{2i+2}) \times \mathbb{Z}_{\geq 0}$ . The maximum dwell time in (43) should be positive for the system to sufficiently allow feedback unavailability, which holds when  $\frac{\lambda_1}{\lambda_2} \chi^2 + \frac{2\Delta_U}{\lambda_1 \lambda_U} - \frac{c}{\lambda_3} > \frac{\lambda_2}{\lambda_1} \|z(t_{2i+1})\|^2 + \frac{2\Delta_U}{\lambda_1 \lambda_U}$ . Imposing this condition on  $\|z(t_{2i+1})\|^2$  and using Theorem 1 of the manuscript yields the following condition for minimum dwell time during which the feedback should be available

$$(t_{2i+1} - t_{2i}) \geq \frac{\lambda_2}{\lambda_3} \ln \left( \frac{\frac{\lambda_2}{\lambda_1} \|z(t_{2i})\|^2}{\frac{\lambda_1^2}{\lambda_2^2} \chi^2 - \frac{\lambda_2 c}{\lambda_1 \lambda_3} - \frac{\lambda_1 c}{\lambda_2 \lambda_3}} \right), \quad (44)$$

for all  $(t, i) \in [t_{2i+1}, t_{2i+2}) \times \mathbb{Z}_{\geq 0}$ . Note that it is permissible to obtain negative values for the dwell-time in (44), since a negative minimum dwell-time for feedback availability would imply the stability guarantees hold even if feedback continues to be unavailable after the time instance  $t_{2i}$ . Additionally, the size of set  $\mathcal{D}$ , i.e.,  $\chi$  needs to be selected according to  $\chi > \sqrt{\frac{\lambda_2^3 c}{\lambda_1^3 \lambda_3} + \frac{\lambda_2 c}{\lambda_1 \lambda_3}}$ , to ensure a positive denominator in (44), thus guaranteeing the feasibility of the minimum dwell-time condition. This results in the additional gain condition  $k_{\min} > \lambda_3 + \bar{\rho} \left( \sqrt{\frac{\lambda_2^3 c}{\lambda_1^3 \lambda_3} + \frac{\lambda_2 c}{\lambda_1 \lambda_3}} \right)$ .



**Omkar Sudhir Patil** received his Bachelor of Technology (B.Tech.) degree in production and industrial engineering from Indian Institute of Technology (IIT) Delhi in 2018, where he was honored with the BOSS award for his outstanding bachelor's thesis project. In 2019, he joined the Nonlinear Controls and Robotics (NCR) Laboratory at the University of Florida under the guidance of Dr. Warren Dixon to pursue his doctoral studies. Omkar received his Master of Science (M.S.) degree in mechanical engineering in 2022 and Ph.D. in mechanical engineering in 2023 from the University of Florida. During his Ph.D. studies, he was awarded the Graduate Student Research Award for outstanding research. In 2023, he started working as a postdoctoral research associate at NCR Laboratory, University of Florida. His research focuses on the development and application of innovative Lyapunov-based nonlinear, robust, and adaptive control techniques.



**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. In 2024, she was awarded the Best Dissertation Award for the Department of Mechanical and Aerospace Engineering.



**Wanjiku A. Makumi** received her B.S. from the Joint Department of Biomedical Engineering at NC State University and UNC Chapel Hill. In 2020, she joined the Nonlinear Controls and Robotics Laboratory at the University of Florida under the guidance of Dr. Warren Dixon where she received her M.S. in mechanical engineering and her Ph.D. in aerospace engineering in 2021 and 2024, respectively. In 2025 she was awarded the Best Dissertation Award for the Department of Mechanical and Aerospace Engineering. She is a recipient of the DoD SMART Scholarship and is currently a research engineer at the Air Force Research Laboratory. Her research focuses on machine learning and adaptive control for unknown nonlinear systems.



**Prof. Warren E. Dixon** received his Ph.D. from the Department of Electrical and Computer Engineering from Clemson University. He worked as a research staff member and Eugene P. Wigner Fellow at Oak Ridge National Laboratory (ORNL) for four years until he joined the University of Florida in the Mechanical and Aerospace Engineering Department. His main research interest has been the development and application of Lyapunov-based control techniques for uncertain nonlinear systems. He is an ASME and IEEE Fellow and his collaborative work has received various early and mid-career awards, including various best paper awards.