Adaptive Output Feedback Control Using Lyapunov-Based Deep Recurrent Neural Networks (Lb-DRNNs)

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

***Abstract*—Unlike traditional feedforward neural networks, recurrent neural networks (RNNs) possess a recurrent connection that allows them to retain past information. This internal memory enables RNNs to effectively model and capture time-varying and accumulative effects observed in dynamic systems, which cannot be achieved by static feedforward neural networks, making RNNs more suitable for tasks, such as state estimation and output feedback (OFB) control. Motivated by the dynamic behavior of RNNs, this article develops an adaptive Lyapunov-based deep RNN (Lb-DRNN) OFB controller for uncertain nonlinear systems. Specifically, an Lb-DRNN observer is designed to adaptively estimate the unknown states of the system and is integrated into an OFB control framework. To ensure real-time adaptation and online learning, the DRNN weights are dynamically adjusted using Lyapunov-based adaptation laws. A Lyapunov-based stability analysis proves asymptotic estimation and tracking error convergence. Validation simulation experiments on an unmanned underwater vehicle system yielded a 31.85% and 86.21% improvement in normalized linear and angular tracking error, respectively, when compared to the shallow RNN-based OFB controller.**

***Index Terms*—Adaptive control, deep learning, Lyapunov methods.**

I. Introduction

Neural networks (NNs) have been widely used as a feedforward estimate of unstructured uncertainties in closed-loop control methods (cf., results, such as [1], [2], [3], [4], [5], [6], [7], [8], [9], and [10]). However, typical NN-based control methods rely on full access to the system states, which can be impractical or impossible for many real-world systems. Motivated to relax full state feedback requirements, results, such as [8] and [9], develop adaptive NN-based output feedback (OFB) controllers, but these results are restricted to feedforward NNs with a single hidden layer. Feedforward NNs have a static structure that only has access to current state information. Previous findings have demonstrated that incorporating a memory component capable of accessing previous state information reduces the required training data and accelerates the learning process [10], [11], [12]. Building on these insights, results, such as [13], augment static NN-based controllers with an external memory, resulting in faster learning and improved function approximation. Although this approach introduces a working memory to the NN, the NN remains static and feedforward, with the augmented memory being external to the NN architecture. Recurrent neural networks (RNNs) are a dynamic type of NN that are specifically designed to handle sequential or temporal data. Unlike feedforward NNs, which process inputs independently and have no internal memory, RNNs have a recurrent connection that allows them to maintain information from previous time steps. This internal memory allows RNNs to capture and model time-varying and cumulative effects present in certain dynamic systems that feedforward NNs cannot. This ability to capture dynamic behavior makes RNNs better suited to construct state observers for OFB control of uncertain nonlinear systems compared to traditional feedforward NNs.

Motivated by the improved memory capabilities of RNNs, previous results, such as [14], [15], [16], and [17], develop RNN-based observers and OFB controllers. While the RNN architectures developed in [15] and [17] are implemented in observers, these results do not consider an OFB control architecture. RNN architectures in results, such as [14], are implemented in an OFB controller, but are trained offline by minimizing a loss function based on prior collected datasets. Offline training techniques do not provide explicit guarantees of stability, and there is a risk that the trained controller may become unstable. In contrast, stability-driven adaptation ensures stability of the control system throughout its operation, adapts to changes in the system dynamics in real time, and is more robust against system uncertainties and disturbances compared to offline approaches [18]. Offline training techniques require a sufficiently rich dataset for training, which may not always be feasible. Thus, stability-driven adaptive control methods are particularly advantageous in dynamic and uncertain environments where accurate modeling and obtaining prior knowledge of the system dynamics may be challenging or impractical. Results, such as [9] and [16], focus on the design and analysis of adaptive RNN architectures that use Lyapunov-based analysis techniques. However, these results, such as other previous RNN-based OFB controllers developed in [16], [17], and [19], are restricted to RNNs with a single hidden layer.

While traditional shallow NNs are capable of approximating nonlinear functions, deep NNs (DNNs) have grown in recent popularity due to their improved performance (cf., [20]). Specifically, DNNs have been shown to be exponentially more efficient than shallow NNs regarding the number of total weights required to achieve comparable function approximation performance [21]. Thus, motivation exists to develop an adaptive deep RNN (DRNN) architecture. However, the more complex structure and nested nonlinearities of DRNNs make developing stability-driven weight adaptation laws difficult. Our preliminary work in [15] develops an adaptive DRNN-based observer using Lyapunov-stability-driven techniques but does not incorporate the developed adaptive RNN architecture and observer state estimates into an OFB control input. Constructing an OFB controller using the DRNN-based observer is technically challenging due to the combined objectives of state estimation and tracking. As a result, the stability analysis involves coupling between the estimation and tracking errors,

Received 1 November 2024; revised 14 May 2025; accepted 16 July 2025. Date of publication 31 July 2025; date of current version 30 December 2025. This work was supported in part by the AFRL under Grant FA8651-21-F-1027 and Grant FA8651-21-F-1025, in part by the AFOSR under Grant FA9550-19-1-0169, and in part by the Office of Naval Research under Grant N00014-21-1-2481. Recommended by Associate Editor A. Bobtsov. *(Corresponding author: Emily J. Griffis.)*

The authors are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: emilygriffis00@ufl.edu; patilomkarsudhir@ufl.edu; makumiw@ufl.edu; wdixon@ufl.edu).

Digital Object Identifier 10.1109/TAC.2025.3594611

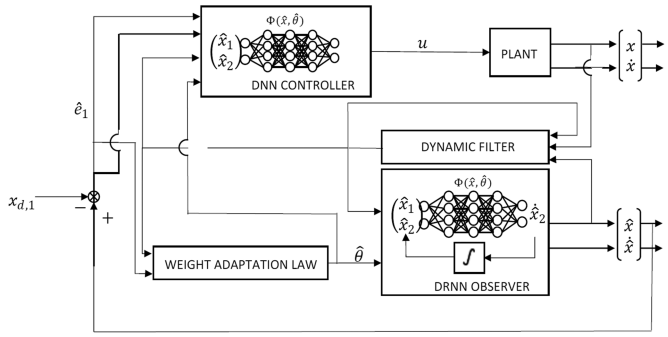

Fig. 1. Block diagram of adaptive DRNN-based OFB controller.

which introduces mathematical challenges in deriving the control and adaptation laws.

In this article, an OFB controller is developed for uncertain nonlinear systems using adaptive DRNNs. Specifically, the contributions of this article are as follows.

1) Inspired by the dynamic nature and memory capabilities of RNNs, a DRNN observer is designed to adaptively estimate the unknown states of the system and is incorporated into a control framework.
2) Unlike our preliminary work in [15], the developed OFB controller is designed to achieve a twofold control objective: asymptotic estimation of the unmeasurable states and asymptotic tracking control.
3) Lyapunov-based stability-driven adaptation laws are developed to adjust the DRNN weights online based on the tracking and observer errors (hence, we will refer to the architecture as Lyapunov-based (Lb-). The developed adaptation laws allow the Lb-DRNN to adapt, learn, and control the system based on the system output in real time.
4) A Lyapunov-based stability analysis shows the developed observer and the overall control design yield asymptotic estimation and trajectory tracking.

Validation simulation experiments were performed on an unmanned underwater vehicle (UUV) system that resulted in a 31.85% and 86.21% improvement in linear and angular tracking error, respectively, when compared to the shallow RNN-based OFB controller.

I. Notations and Preliminaries

The space of essentially bounded Lebesgue measurable functions is denoted by $\mathcal{L}_\infty$. The Kronecker product is denoted by $\otimes$. The 1-norm and the convolution operator are denoted by $\|\cdot\|_1$ and $*$, respectively. The vectorization operator is denoted by $\text{vec}(\cdot)$, i.e., given $A \triangleq [a_{i,j}] \in \mathbb{R}^{n\times m}$, $\text{vec}(A) \triangleq [a_{1,1},\ldots,a_{n,1},\ldots,a_{1,m},\ldots,a_{n,m}]^\top$. The vectorization operator satisfies the following properties [22, Prop. 7.1.9]. Given any $A \in \mathbb{R}^{p\times a}$, $B \in \mathbb{R}^{a\times r}$, and $C \in \mathbb{R}^{r\times s}$

$$\text{vec}(ABC) = \left(C^\top \otimes A\right)\text{vec}(B) \tag{1}$$

and consequently

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

Function compositions are denoted using the symbol $\circ$, e.g., given suitable functions g and h, $(g\circ h)(x) = g(h(x))$. The right-to-left matrix product operator is represented by $\overleftarrow{\prod}$, i.e., $\overleftarrow{\Pi}_{p=a}^{m} A_p = A_m \ldots A_{a+1}A_a$ and $\overleftarrow{\Pi}_{p=a}^{m} A_p = I$ if $a > m$, where I denotes the identity matrix. The pseudoinverse of the full row rank matrix $A \in \mathbb{R}^{n\times m}$ is denoted by A^+, where $A^+ \triangleq A^\top(AA^\top)^{-1}$. The Filippov set-valued map defined in [23, eq. (2b)] is denoted by $K[\cdot]$. Consider a Lebesgue measurable and locally essentially bounded function $h : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$. Then, the function $y : \mathcal{I} \to \mathbb{R}^n$ is called a *Filippov solution* of $\dot{y} = h(y,t)$ on the interval $\mathcal{I} \subseteq \mathbb{R}_{\geq 0}$ if y is absolutely continuous on $\mathcal{I}$ and $\dot{y} \overset{a.a.t}{\in} K[h](y,t)$, where $a.a.t.$ denotes almost all time.

II. Problem Formulation

A. Model Dynamics

Consider a second-order nonlinear system modeled as[1]

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x) + g(x_1)\,u \\ y &= x_1 \end{aligned} \tag{3}$$

where $x \triangleq [x_1^\top\ x_2^\top]^\top \in \mathbb{R}^{2n}$ and $u \in \mathbb{R}^m$ denote the generalized state and control input of the system, respectively, $f : \mathbb{R}^{2n} \to \mathbb{R}^n$ denotes a continuously differentiable function, $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ denotes a continuous function, and $y \in \mathbb{R}^n$ denotes the measurable output of the system. The control objective is to design an adaptive DRNN controller to track a desired trajectory $x_{d,1} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ despite the system uncertainty and unavailability of the state x_2. It is assumed that the desired trajectory $x_{d,1}$ is designed to be sufficiently smooth such that $|x_{d,1}(t)| \leq \overline{x_d}$, $|\dot{x}_{d,1}(t)| \leq \overline{\dot{x}_d}$, and $|\ddot{x}_{d,1}(t)| \leq \overline{\ddot{x}_d}$ for all $t \in \mathbb{R}_{\geq 0}$, where $\overline{x}_d, \overline{\dot{x}_d}, \overline{\ddot{x}_d} \in \mathbb{R}_{>0}$ denote known constants. The following assumptions facilitate the subsequent observer development and control design.

Assumption 1: The function g is known and full row rank.[2]
Assumption 2: The system in (3) is observable.
Assumption 3: The output state x_1 is measurable.

B. DRNN Model

An adaptive DRNN architecture is developed to estimate the unknown model dynamics in (3) in real time and is implemented in a subsequently designed OFB-based controller, as shown in Fig. 1. Analytic, stability-driven weight adaptation laws adjust the weights of the DRNN in real time using state estimation and tracking errors, providing a continuous learning method.[3]

Motivated by estimating continuous-time dynamics, continuous-time RNN architectures were developed in results, such as [24], by converting existing discrete-time shallow RNNs to continuous-time using Euler's method. Using Euler's method and the discrete-time DRNN representation in [25], a continuous-time DRNN can be modeled in

[1]For ease of exposition, a second-order nonlinear system was considered. However, the error system development in [15] can be used with the developed adaptive DRNN OFB controller for control of Nth order strict nonlinear systems with an unmeasurable state x_N.

[2]The control development in [3] can be used to account for an uncertain, linearly parameterizable control effectiveness g.

[3]Offline training methods (e.g., Adam and Levenberg–Marquardt algorithm) could be used to initialize the DRNN if pretraining data are available. If no data are available, the DRNN can also be initialized with random weights or using pretraining data taken from a similar system (i.e., transfer learning).

continuous-time as

$$\dot{h} = -bh + W_k^\top \phi_k \circ \ldots \circ W_1^\top \phi_1 \circ W_0^\top y \tag{4}$$

where $b \in \mathbb{R}_{>0}$ denotes a user-selected time constant, $y \in \mathbb{R}^{L_{k+1}+2n+1}$ denotes a concatenated state vector defined as $y \triangleq [h^\top \; x_a^\top]^\top$, $x_a \in \mathbb{R}^{2n+1}$ denotes the augmented input defined as $x_a \triangleq [x^\top \; 1]^\top$, and $h \in \mathbb{R}^{L_{k+1}}$ denotes the hidden state. The weight matrices and smooth activation functions[4] at the jth layer are denoted by $W_j^\top \in \mathbb{R}^{L_{j+1} \times L_j}$ and $\phi_j : \mathbb{R}^{L_j} \to \mathbb{R}^{L_j}$ for all $j \in \{0, \ldots, k\}$ and $j \in \{1, \ldots, k\}$, respectively, where $\theta \triangleq [\text{vec}(W_0)^\top \ldots \text{vec}(W_k)^\top]^\top \in \mathbb{R}^{\sum_{j=0}^{k} L_j L_{j+1}}$ and $L_j \in \mathbb{R}_{>0}$ for all $j \in \{0, \ldots, k\}$ denotes the number of neurons in the jth layer. Since the hidden state is an input to the first hidden layer, $W_0^\top \in \mathbb{R}^{L_1 \times (L_{k+1}+2n+1)}$. To incorporate a bias term, x_a and ϕ_j are augmented with 1 for all $j \in \{1, \ldots, k\}$. For DNNs with multiple types of activation functions at each layer, ϕ_j may be modeled as $\phi_j \triangleq [\varsigma_{j,1}, \varsigma_{j,2}, \ldots, \varsigma_{j,L_{j-1}}, 1]^\top$, where $\varsigma_{j,i} : \mathbb{R} \to \mathbb{R}$ for all $j \in \{1, \ldots, k\}$ and $i \in \{1, \ldots, L_j\}$ denotes the activation function at the ith node of the jth layer.

A recursive representation of the adaptive Lb-DRNN architecture can be modeled as

$$\Phi_j = \begin{cases} W_j^\top \phi_j \left(\Phi_{j-1}\right), & j = \{1, \ldots, k\} \\ W_j^\top y, & j = 0 \end{cases} \tag{5}$$

where $\Phi_j(x, \theta) \in \mathbb{R}^{L_{j+1}}$ denotes the output of the jth layer defined as $\Phi_j \triangleq W_j^\top \phi_j \circ W_{j-1}^\top \phi_{j-1} \circ \ldots \circ W_1^\top \phi_1 \circ W_0^\top y$ for all $j \in \{0, \ldots, k\}$. From (5), the Lb-DRNN in (4) can be represented as $\dot{h} = -bh + \Phi(x, \theta)$, where $\Phi(x, \theta) \triangleq \Phi_k(x, \theta)$.

The Lb-DRNN architecture in (4) can then be used to model the unknown system dynamics in (3) using the unknown state x_2 as the hidden state h. Using [26, Thm. 3.2], the universal function approximation property states that the function space of DNNs given by (5) is dense in $\mathcal{C}(\mathcal{X})$, where $\mathcal{C}(\mathcal{X})$ denotes the space of continuous functions over the set $\mathcal{X} \subseteq \mathbb{R}^{2n}$. Therefore, for any prescribed $\bar{\varepsilon} \in \mathbb{R}_{>0}$, there exist ideal weight matrices $\theta \in \mathbb{R}^{\sum_{j=0}^{k} L_j L_{j+1}}$ such that $\|f(x) - (-bx_2 + \Phi(x, \theta))\| \leq \bar{\varepsilon}$ for all $x \in \mathcal{X}$. Thus, the dynamics in (3) can be modeled using an adaptive DRNN architecture as

$$\dot{x}_2 = -bx_2 + \Phi\left(x, \theta\right) + \varepsilon(x) + g\left(x_1\right) u \tag{6}$$

where $\varepsilon : \mathbb{R}^{2n} \to \mathbb{R}^n$ denotes the function approximation error such that $\sup_{x \in \mathcal{X}} \|\varepsilon(x)\| \leq \bar{\varepsilon}$. The following typical assumption is made about the DRNN model in (4).

Assumption 4: The DRNN weights can be bounded as $\|\theta\| \leq \bar{\theta}$, where $\bar{\theta} \in \mathbb{R}_{>0}$ denotes a known constant [4, Assumption 1].[5]

III. Control Development

Since the state x_2 is not available and the dynamics in (3) are unknown and unstructured, an adaptive DRNN observer is constructed to facilitate the control development. Thus, an adaptive DRNN observer is designed to estimate x_2. To quantify the control objectives, the estimation error $\tilde{x}_1 \in \mathbb{R}^n$ and tracking error $e_1 \in \mathbb{R}^n$ are defined as

$$\tilde{x}_1 \triangleq x_1 - \hat{x}_1$$
$$e_1 \triangleq x_1 - x_{d,1} \tag{7}$$

respectively, for state estimates $\hat{x} \triangleq [\hat{x}_1^\top \; \hat{x}_2^\top]^\top$. Using the estimation and tracking errors, auxiliary estimation and tracking errors $\xi, \; r \in \mathbb{R}^n$ are defined as [28]

$$\xi \triangleq \dot{\tilde{x}}_1 + \alpha \tilde{x}_1 + \eta$$
$$r \triangleq \dot{e}_1 + \alpha e_1 + \eta \tag{8}$$

respectively, where $\alpha \in \mathbb{R}_{>0}$ denotes a user-selected constant and $\eta \in \mathbb{R}^n$ denotes the output of a dynamic filter introduced to compensate for the lack of direct measurements of the state x_2. The dynamic filter is designed as [29]

$$\eta \triangleq p - (\alpha + k_r)\, \tilde{x}_1$$
$$\dot{\nu} \triangleq p - \alpha\nu - (\alpha + k_r)\, \tilde{x}_1$$
$$\dot{p} \triangleq -(k_r + 2\alpha)\, p - \nu + \left((\alpha + k_r)^2 + 1\right) \tilde{x}_1 + e_1 \tag{9}$$

where $k_r \in \mathbb{R}_{>0}$ denotes a user-selected constant, $p \in \mathbb{R}^n$ denotes an internal filter variable, and $\nu \in \mathbb{R}^n$ denotes the auxiliary output of the filter. The filter variables p and ν are initialized such that $p(0) = (\alpha + k_r)\tilde{x}_1(0)$ and $\nu(0) = 0$, respectively.

The filter in (9) uses the estimation error $\tilde{x}_1$ and tracking error e_1 as inputs and produces the two filter outputs ν and η. The internal filter variable p is used to generate the signal η without involving the unavailable derivative of the estimation error $\dot{\tilde{x}}_1$. From (8) and (9), the dynamic filter can be related to the unknown auxiliary estimation error ξ and unknown auxiliary tracking error r as

$$\xi = \dot{e}_{\text{es}} + \alpha e_{\text{es}}$$
$$r = \dot{e}_{\text{tr}} + \alpha e_{\text{tr}} \tag{10}$$

where $e_{\text{es}} \in \mathbb{R}^n$ and $e_{\text{tr}} \in \mathbb{R}^n$ are auxiliary errors defined as $e_{\text{es}} \triangleq \tilde{x}_1 + \nu$ and $e_{\text{tr}} \triangleq e_1 + \nu$, respectively. Taking the time derivative of η and using (8) and (9) yield

$$\dot{\eta} = -(\alpha + k_r)\, \xi - \alpha\eta + \tilde{x}_1 + e_1 - \nu. \tag{11}$$

A. Observer Design

The adaptive Lb-DRNN observer for the uncertain nonlinear system in (3) is designed as[6]

$$\dot{\hat{x}}_1 = \hat{x}_2$$
$$\dot{\hat{x}}_2 = -b\hat{x}_2 + \Phi\left(\hat{x}, \hat{\theta}\right) + g\left(x_1\right) u + \beta_1 \text{sgn}\left(e_{\text{es}}\right) + \chi \tag{12}$$

where $\text{sgn}(\cdot)$ denotes the elementwise signum function, $\hat{\theta} \triangleq [\text{vec}(\widehat{W}_0)^\top, \ldots, \text{vec}(\widehat{W}_k)^\top]^\top$, $\beta_1 \in \mathbb{R}_0$ denotes a user-defined constant, and $\widehat{W}_j^\top \in \mathbb{R}^{L_{j+1} \times L_j}$ denotes the weight estimates for all $j \in \{0, \ldots, k\}$. The auxiliary term $\chi \in \mathbb{R}^n$ is designed as

$$\chi \triangleq -\left(\gamma\left(k_r + \alpha\right) + 2\alpha\right)\eta + (\gamma - \alpha)\,\nu + \tilde{x}_1 - \nu \tag{13}$$

where $\gamma \in \mathbb{R}_{>0}$ denotes a user-selected constant.

[4]The adaptive DRNN architecture in (4) does not consider nonsmooth activation functions for notational simplicity. However, the switched analysis in [2] can be used with the developed method to incorporate nonsmooth activation functions into the RNN architecture.

[5]Results, such as [27], provide in-roads to robust adaptive extensions for an uncertain bound $\bar{\theta}$.

[6]As is typical in observer design approaches (see [30]), the state x_1 is also approximated, even though it is measurable, as a means to provide feedback for the construction of the estimate for the unmeasurable state x_2.

Taking the derivative of ξ, substituting (6)–(8), (12), and (13), and adding and subtracting $\Phi(\hat{x},\theta)$ yield

$$\dot{\xi} = -b\dot{\tilde{x}}_1 + \Phi(\hat{x},\theta) - \Phi(\hat{x},\hat{\theta}) + \Phi(x,\theta) - \Phi(\hat{x},\theta) + \varepsilon(x) - \beta_1 \text{sgn}(e_{\text{es}}) - \chi + \alpha(\xi - \alpha\tilde{x}_1 - \eta) + \dot{\eta}. \tag{14}$$

To address the mathematical challenges posed by the nested nonlinearity of the DRNN architecture, a first-order Taylor series approximation-based error model is evaluated as [4]

$$\Phi(\hat{x},\theta) - \Phi\left(\hat{x},\hat{\theta}\right) = \widehat{\Phi}'\tilde{\theta} + \mathcal{O}^2\left(\hat{x},\tilde{\theta}\right) \tag{15}$$

where $\mathcal{O}^2(\hat{x},\tilde{\theta})$ denotes high-order terms and $\tilde{\theta} \triangleq \theta - \hat{\theta}$. Substituting (15) into (14) yields the closed-loop error system

$$\dot{\xi} = N_1 + \widehat{\Phi}'\tilde{\theta} - \beta_1 \text{sgn}(e_{\text{es}}) - \chi + \alpha(\xi - \alpha\tilde{x}_1 - \eta) + \dot{\eta} \tag{16}$$

where the auxiliary term $N_1 \in \mathbb{R}^n$ is defined as $N_1 \triangleq -b\dot{\tilde{x}}_1 + \Phi(x,\theta) - \Phi(\hat{x},\theta) + \mathcal{O}^2(\hat{x},\tilde{\theta}) + \varepsilon(x)$.

B. Control Design

An OFB controller is designed using the developed adaptive Lb-DRNN architecture and observed state estimates $\hat{x}$. The Lb-DRNN weights are adjusted online using subsequently designed stability-driven weight adaptation laws that allow the developed OFB control design to estimate the unknown states and system dynamics in real time. The control input is designed as

$$u \triangleq g(x_1)^+ \left[-(-b\hat{x}_2 + \Phi(\hat{x},\hat{\theta})) - \beta_2 \text{sgn}(e_{\text{tr}}) + \ddot{x}_{d,1} - (k_r + \alpha)(\dot{\hat{e}}_1 + \alpha\hat{e}_1) - \alpha^2 e_1 - \nu \right] \tag{17}$$

where $\hat{e}_1 \triangleq \hat{x}_1 - x_{d,1}$ and $\beta_2 \in \mathbb{R}_0$ denotes a user-defined control gain. Taking the time-derivative of r, substituting (6)–(8) and (17), adding and subtracting $\Phi(\hat{x},\theta)$, and using the Taylor series-based approximation in (15) yield the closed-loop error system

$$\dot{r} = N_1 + \widehat{\Phi}'\tilde{\theta} - (k_r + \alpha)\left(\dot{\hat{e}}_1 + \alpha\hat{e}_1\right) + \dot{\eta} - \beta_2 \text{sgn}(e_{\text{tr}}) + \alpha(r - \eta) + \nu. \tag{18}$$

C. Adaptive Weight Update Laws

In this section, a Lyapunov stability-driven weight adaptation law is developed for the DRNN architecture. The weight adaptation law allows the developed OFB controller to adaptively compensate for the uncertain model dynamics of the system while ensuring stability guarantees. The DRNN weight adaptation law is designed as

$$\dot{\hat{\theta}} \triangleq \Gamma\widehat{\Phi}'^{\top}(e_{\text{es}} + e_{\text{tr}}) \tag{19}$$

where $\Gamma \in \mathbb{R}^{\sum_{j=0}^{k} L_j L_{j+1} \times \sum_{j=0}^{k} L_j L_{j+1}}$ and $\widehat{\Phi}' \in \mathbb{R}^{2n \times \sum_{j=0}^{k} L_j L_{j+1}}$ denote a user-selected positive-definite gain matrix and a shorthand notation denoting the Jacobian $\widehat{\Phi}' \triangleq [\widehat{\Phi}'_0, \ldots, \widehat{\Phi}'_k]$, where the shorthand notation $\widehat{\Phi}'_j$ is defined as $\widehat{\Phi}'_j \triangleq \frac{\partial \Phi_j(\hat{x},\hat{\theta})}{\partial \hat{\theta}}$, for all $j \in \{0,\ldots,k\}$. Using the chain rule, the DRNN model in (5), and the properties of the vectorization operator in (1) and (2), the terms $\widehat{\Phi}'_0$ and $\widehat{\Phi}'_j$ can be expressed as $\widehat{\Phi}'_0 \triangleq (\prod_{l=1}^{\curvearrowleft k} \widehat{W}_l^{\top}\hat{\phi}'_l)(I_{L_1} \otimes \hat{x}_a^{\top})$ and $\widehat{\Phi}'_j \triangleq (\prod_{l=j+1}^{\curvearrowleft k} \widehat{W}_l^{\top}\hat{\phi}'_l)(I_{L_{j+1}} \otimes \hat{\phi}_j^{\top}) \forall j \in \{1,\ldots,k\}$, respectively, where $\hat{x}_a \in \mathbb{R}^{2n+1}$ denotes the augmented RNN input $\hat{x}_a \triangleq [\hat{x}^{\top}\, 1]^{\top}$, and the shorthand notations $\hat{\phi}_j$ and $\hat{\phi}'_j$ are defined as $\hat{\phi}_j \triangleq \phi_j(\Phi_{j-1}(\hat{x},\hat{\theta}))$ and $\hat{\phi}'_j \triangleq \phi'_j(\Phi_{j-1}(\hat{x},\hat{\theta}))$ for all $j \in \{1,\ldots,k\}$, respectively, where $\phi'_j \triangleq \frac{\partial}{\partial y}\phi_j(y) \forall y \in \mathbb{R}^{l_2}$.

IV. Stability Analysis

Let the function $\mathcal{V}_L : \mathbb{R}^{\psi} \to \mathbb{R}_{\geq 0}$ be defined as

$$\mathcal{V}_L(\zeta) \triangleq \frac{\gamma}{2}\tilde{x}_1^{\top}\tilde{x}_1 + \frac{1}{2}\xi^{\top}\xi + \frac{\gamma}{2}e_1^{\top}e_1 + \frac{1}{2}r^{\top}r + \frac{\gamma}{2}\eta^{\top}\eta + \frac{\gamma}{2}\nu^{\top}\nu + P + \frac{\alpha}{2}\tilde{\theta}^{\top}\Gamma^{-1}\tilde{\theta} \tag{20}$$

where the concatenated state vector $\zeta \in \mathbb{R}^{\psi}$ is defined as $\zeta \triangleq [z^{\top}\ \sqrt{P}\ \tilde{\theta}^{\top}]^{\top}$, $z \triangleq [\tilde{x}_1^{\top}\ \xi^{\top}\ e_1^{\top}\ r^{\top}\ \eta^{\top}\ \nu^{\top}]^{\top}$, $\psi \triangleq 6n + 1 + \sum_{j=0}^{k} L_j L_{j+1}$, and $\alpha, \gamma \in \mathbb{R}_{>0}$ and $\Gamma \in \mathbb{R}^{\sum_{j=0}^{k} L_j L_{j+1} \times \sum_{j=0}^{k} L_j L_{j+1}}$ denote user-selected constants defined in (8), (13), and (19), respectively. The term $P : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denotes a subsequently designed positive P-function used to account for mismatches that appear in the function $\mathcal{V}_L$ due to the weight adaptation law in (19) being expressed in terms of the known errors e_{es} and e_{tr} rather than the unknown auxiliary estimation and tracking errors ξ and r. The function $\mathcal{V}_L$ in (20) satisfies

$$\lambda_1 \|\zeta\|^2 \leq \mathcal{V}_L \leq \lambda_2 \|\zeta\|^2 \tag{21}$$

where the auxiliary constants λ_1 and λ_2 are defined as $\lambda_1 \triangleq \frac{1}{2}\min\{1, \gamma, \alpha\lambda_{\min}\{\Gamma^{-1}\}\}$ and $\lambda_2 \triangleq \frac{1}{2}\max\{1, \gamma, \alpha\lambda_{\max}\{\Gamma^{-1}\}\}$, respectively. The closed-loop error system has a discontinuous right-hand side due to the signum function, and therefore, a Filippov regularization is applied on the system to regularize the discontinuities of the right-hand side by reformulating it as a differential inclusion [31]. As a result, it does not have solutions defined in the classical sense of continuously differentiable solutions. However, for systems with discontinuous right-hand sides given by Lebesgue measurable and locally essentially bounded functions, absolutely continuous Filippov solutions are known to exist and can be examined by applying a Filippov regularization [32]. Therefore, let $\partial\mathcal{V}_L$ denote the Clarke gradient of $\mathcal{V}_L$ defined in [32, p. 39]. Since $\zeta \mapsto \mathcal{V}_L(\zeta)$ is continuously differentiable, $\partial\mathcal{V}_L(\zeta) = \{\nabla\mathcal{V}_L(\zeta)\}$, where ∇ denotes the standard gradient operator. Based on the chain rule in [33, Thm. 2.2], it can be verified that $t \to \mathcal{V}_L(\zeta(t))$ satisfies the differential inclusion $\dot{\mathcal{V}}_L \overset{a.a.t.}{\in} \bigcap_{\mu \in \partial\mathcal{V}_L(\varsigma)} \mu^{\top}(\psi,t)K[h](\varsigma,t)$ for $\varsigma \triangleq [z^{\top}\ P\ \tilde{\theta}^{\top}]^{\top}$. Taking the derivative of $\mathcal{V}_L(\zeta)$, substituting in (9), (11), and (16), and canceling cross-terms yield

$$\dot{\mathcal{V}}_L \overset{a.a.t}{\in} -\alpha\gamma\tilde{x}_1^{\top}\tilde{x}_1 + \gamma e_1^{\top}\dot{e}_1 + r^{\top}\dot{r} - \alpha\gamma\nu^{\top}\nu - k_r\xi^{\top}\xi + \xi^{\top}\left(\widehat{\Phi}'\tilde{\theta} - \beta_1 K[\text{sgn}](e_{\text{es}}) + e_1 + N_1\right) - \gamma\eta^{\top}(\alpha\eta + e_1) + \dot{P} - \alpha\tilde{\theta}^{\top}\Gamma^{-1}\dot{\hat{\theta}}.$$

Substituting in (8) and (11), the closed-loop tracking error system in (18), and the weight adaptation law in (19) yields

$$\dot{\mathcal{V}}_L \overset{a.a.t}{\in} -\alpha\gamma\tilde{x}_1^{\top}\tilde{x}_1 - \alpha\gamma\nu^{\top}\nu - \alpha\gamma e_1^{\top}e_1 - k_r r^{\top}r + \dot{P} + \xi^{\top}\left(\widehat{\Phi}'\tilde{\theta} - \beta_1 K[\text{sgn}](e_{\text{es}}) + e_1 + N_1\right) - k_r\xi^{\top}\xi - \gamma\eta^{\top}\alpha\eta - \alpha\tilde{\theta}^{\top}\widehat{\Phi}'^{\top}(e_{\text{es}} + e_{\text{tr}}) + r^{\top}\left(N_1 + \widehat{\Phi}'\tilde{\theta} - \beta_2 K[\text{sgn}](e_{\text{tr}}) + \tilde{x}_1\right). \tag{22}$$

Let the sets $\mathcal{S} \subset \mathbb{R}^{\psi}$ and $\mathcal{D} \subset \mathbb{R}^{\psi}$ be defined as $\mathcal{S} \triangleq \{\sigma \in \mathcal{D} : \|\sigma\| \leq \sqrt{\frac{\lambda_1}{\lambda_2}}\omega\}$ and $\mathcal{D} \triangleq \{\sigma \in \mathbb{R}^{\psi} : \|\sigma\| < \omega\}$, respectively, for some bounding constant $\omega \in \mathbb{R}_{>0}$. The following theorem establishes asymptotic

tracking and estimation error convergence for the developed adaptive DRNN observer and overall OFB control design.

Theorem 1: Consider the system in (3) and let Assumptions 1–4 hold. The adaptive Lb-DRNN observer in (12), controller in (17), and Lb-DRNN weight adaptation law in (19) ensure asymptotic estimation and tracking error convergence in the sense that $\|e_{\text{tr}}\| \to 0$, $\|e_{\text{es}}\| \to 0$, $\|x_2 - \hat{x}_2\| \to 0$, and $\|x_2 - \dot{x}_{d,1}\| \to 0$ as $t \to \infty$, provided $\zeta(0) \in \mathcal{S}$, and the following sufficient gain conditions are satisfied:

$$\beta_1, \beta_2 > \xi_1 + \xi_2 + \frac{1}{\alpha - \lambda_P}(\alpha\xi_2 + \xi_3)$$
$$\alpha \geq \frac{1}{2\gamma k_r}. \tag{23}$$

Proof: Consider the candidate Lyapunov function in (20). The P-function is designed as

$$\begin{aligned} P(t) \triangleq \mathrm{e}^{-\lambda_P t} * \Big(& \alpha \left(e_{\text{es}} + e_{\text{tr}}\right)^\top N_2 \\ & + (\alpha - \lambda_P)\left(\beta_1 \|e_{\text{es}}\|_1 + \beta_2 \|e_{\text{tr}}\|_1\right) \\ & - (\alpha - \lambda_P)\left(e_{\text{es}} + e_{\text{tr}}\right)^\top (N_1 + N_2) \\ & + \left(e_{\text{es}} + e_{\text{tr}}\right)^\top \left(\dot{N}_1 + \dot{N}_2\right)\Big) + \beta_1 \|e_{\text{es}}\|_1 \\ & + \beta_2 \|e_{\text{tr}}\|_1 - \left(e_{\text{es}} + e_{\text{tr}}\right)^\top (N_1 + N_2) \end{aligned} \tag{24}$$

where $\lambda_P \in \mathbb{R}_{>0}$ denotes a constant and $N_2 \in \mathbb{R}^n$ denotes an auxiliary term defined as $N_2 \triangleq \widehat{\Phi}'\tilde{\theta}$. Using Assumption 4 and the continuous differentiability of the auxiliary terms N_1 and N_2, N_1 and N_2 and the time-derivative of $N_1 + N_2$ can be bounded as $\|N_1\| \leq \xi_1$, $\|N_2\| \leq \xi_2$, and $\|\dot{N}_1 + \dot{N}_2\| \leq \xi_3$, respectively, when $\zeta \in \mathcal{D}$, for known constants $\xi_1,\ \xi_2,\ \xi_3 \in \mathbb{R}_{>0}$. Therefore, using similar arguments as in [34, Lemma 4], it can be shown that P remains positive for all time $t \in \mathbb{R}_{\geq 0}$ provided the sufficient gain conditions in (23) are satisfied. The P-function in (24) is a Filippov solution to

$$\begin{aligned} \dot{P} \in & -\lambda_P P - \xi^\top \left(N_1 - \beta_1 K\left[\text{sgn}\right](e_{\text{es}})\right) \\ & - r^\top \left(N_1 - \beta_2 K\left[\text{sgn}\right](e_{\text{tr}})\right) - \left(\dot{e}_{\text{es}} + \dot{e}_{\text{tr}}\right)^\top N_2. \end{aligned} \tag{25}$$

The mappings $t \to \xi^\top K[\text{sgn}](e_{\text{es}})$ and $t \to r^\top K[\text{sgn}](e_{\text{tr}})$ are set-valued only for the set of time instants $T_1 = \{t \in [0,\infty) | \exists i \in \{1,2,\ldots,n\} \text{s.t. } e_{\text{es},i}(t) = 0 \wedge \xi_i(t) \neq 0\}$ and $T_2 = \{t \in [0,\infty) | \exists i \in \{1,2,\ldots,n\} \text{s.t. } e_{\text{tr},i}(t) = 0 \wedge r_i(t) \neq 0\}$, respectively. According to [34, Lemma 1], the sets T_1 and T_2 have Lebesgue measure zero. Therefore, using (10), (22), (25), and the definition of N_2 and canceling like terms yield

$$\begin{aligned} \dot{\mathcal{V}}_L \overset{a.a.t}{\in} & -\alpha\gamma\tilde{x}_1^\top\tilde{x}_1 - \alpha\gamma\nu^\top\nu - \alpha\gamma e_1^\top e_1 - \alpha\gamma\eta^\top\eta \\ & - k_r r^\top r - k_r\xi^\top\xi - \lambda_P P + \xi^\top e_1 + r^\top\tilde{x}_1 \end{aligned} \tag{26}$$

when $\zeta \in \mathcal{D}$. Using Young's inequality, (26) can be further bounded as

$$\dot{\mathcal{V}}_L \leq -\lambda_3 \|z\|^2 - \lambda_P P \quad \forall \zeta \in \mathcal{D} \tag{27}$$

where $\lambda_3 \triangleq \min\{\alpha\gamma - \frac{1}{2k_r}, \alpha\gamma, \frac{k_r}{2}\}$. Therefore, provided the sufficient gain condition in (23) is satisfied, $\dot{\mathcal{V}}_L \leq 0, \forall \zeta \in \mathcal{D}$.

To show $\zeta(t) \in \mathcal{D}$ for all $t \geq 0$, the fact that $\mathcal{V}_L$ is nonincreasing can be used to show $\|\zeta(t)\| \leq \sqrt{\frac{\mathcal{V}_L(t)}{\lambda_1}} \leq \sqrt{\frac{\mathcal{V}_L(0)}{\lambda_1}} \leq \sqrt{\frac{\lambda_2}{\lambda_1}}\|\zeta(0)\|$. Therefore, if $\|\zeta(0)\| \leq \sqrt{\frac{\lambda_1}{\lambda_2}}\omega$, then $\|\zeta\| < \omega$ for all $t \geq 0$. Thus, the states ζ should be initialized such that $\zeta(0) \in \mathcal{S}$ in order to guarantee that $\zeta(t) \in \mathcal{D}$ for all $t \in [0,\infty)$. To show $x \in \mathcal{X}$ so that the universal function approximation property holds, let the set $\Upsilon \subseteq \mathcal{X}$ be defined as $\Upsilon \triangleq \{\varsigma \in \mathcal{X} : \|\varsigma\| < \overline{x_d} + \overline{\dot{x}_d} + (3 + 2\alpha + \alpha^2)\omega\}$. Thus, if $\|\zeta(0)\| \in \mathcal{S}$, then $\|\zeta(t)\| \leq \omega$, and therefore, $\|e_1(t)\| \leq \omega$, $\|\eta(t)\| \leq \omega$, and $\|r(t)\| \leq \omega$. Hence, using (7) and (8), $\|x\|$ can be bounded as $\|x\| \leq \overline{x_d} + \overline{\dot{x}_d} + (3 + 2\alpha + \alpha^2)\omega$. Therefore, if $\zeta(0) \in \mathcal{S}$, then $x \in \Upsilon \subseteq \mathcal{X}$, and thus, the universal function approximation property holds.

Since $\zeta \in \mathcal{L}_\infty$, $x_1, \hat{x}_1, e_{\text{tr}}, e_{\text{es}} \in \mathcal{L}_\infty$. Since $e_{\text{tr}}, e_{\text{es}}, \widehat{\Phi}' \in \mathcal{L}_\infty$ and Φ is a smooth function, $\dot{\hat{\theta}} \in \mathcal{L}_\infty$. Using (20), (21), and (27), the LaSalle–Yoshizawa corollary in [35, Corollary 1] can be invoked to show $\|\tilde{x}_1\| \to 0$, $\|e_1\| \to 0$, $\|\nu\| \to 0$, $\|\eta\| \to 0$, $\|\xi\| \to 0$, and $\|r\| \to 0$ as $t \to \infty$. From (7) and (8), it can be further shown that $\|e_{\text{tr}}\| \to 0$, $\|e_{\text{es}}\| \to 0$, $\|x_2 - \hat{x}_2\| \to 0$, and $\|x_2 - \dot{x}_{d,1}\| \to 0$ as $t \to \infty$. ■

V. Simulations

To demonstrate the performance and efficacy of the developed adaptive DRNN-based OFB controller, comparative simulations were performed with a shallow RNN (henceforth referred to as "SRNN") [16] and a central difference observer (henceforth referred to as "CD") as baselines for comparison. The simulations were performed on an UUV system modeled as [36]

$$\begin{aligned} \dot{x}_1 &= \dot{\eta} \\ \dot{x}_2 &= -\overline{M}^{-1}(\eta)\left(\overline{C}(\eta,\dot{\eta},\nu)\dot{\eta} + \overline{D}(\eta,\nu)\dot{\eta}\right) + \overline{M}^{-1}(\eta)\tau_n \end{aligned} \tag{28}$$

where $x_1 = \eta \in \mathbb{R}^6$ denotes a vector of position and orientation with coordinates in the Earth-fixed frame, $x_2 = \dot{\eta} \in \mathbb{R}^6$ 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, and control input in the Earth-fixed frame can be represented by $\overline{M} : \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$, $\overline{C} : \mathbb{R}^6 \times \mathbb{R}^6 \times \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$, $\overline{D} : \mathbb{R}^6 \times \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$, and $\tau_n : \mathbb{R}_{\geq 0} \to \mathbb{R}^6$, respectively. The velocities in the body-fixed frame can be related to the velocities in the Earth-fixed frame using the relation

$$\dot{\eta} = J(\eta)\nu \tag{29}$$

where $J : \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$ is a Jacobian transformation matrix relating the two frames [36, eq. (2)]. Using the kinematic transformation in (29), the Earth–fixed dynamics in (28) can be expressed using body-fixed dynamics as $\overline{M} = J^{-\top}MJ^{-1}$,$\overline{C} = J^{-\top}[C(\nu) - MJ^{-1}\dot{J}]J^{-1}$, $\overline{D} = J^{-\top}D(\nu)J^{-1}$, and $\tau_n = J^{-\top}\tau_b$, where $M \in \mathbb{R}^{6\times 6}$, $C : \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$, $D : \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$, and $\tau_b : \mathbb{R}_{\geq 0} \to \mathbb{R}^6$ denote the inertial effects, centripetal-Coriolis effects, hydrodynamic damping 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 [37, eq. (2.246)]

$$M = \text{diag}\{m_1, m_2, m_3, m_4, m_5, m_6\}$$
$$\begin{aligned} D = \text{diag}\{ & d_{11} + d_{12}|\nu(1)|, d_{21} + d_{22}|\nu(2)|, d_{31} + d_{32}|\nu(3)|, \\ & d_{41} + d_{42}|\nu(4)|, d_{51} + d_{52}|\nu(5)|, d_{61} + d_{62}|\nu(6)|\} \end{aligned}$$

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

where the following numerical values of mass, inertia, and damping parameters were used (see Table I).

TABLE I
UUV SYSTEM PARAMETERS [37, EQ. (2.247)]

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

TABLE II
SIMULATION PERFORMANCE RESULTS FOR 50 RUNS

Architecture		$\|e_1\|$	$\|\tilde{x}_2\|$	$\|\tau_b\|$
SRNN	lin	0.135±0.051[m]	0.075±0.156[m/s]	106±17.5[N]
	ang	0.087±0.051[rad]	0.062±0.097[rad/s]	29.2±22.0[N/m]
CD	lin	0.092±0.012[m]	0.101±0.00[m/s]	195±0.4[N]
	ang	0.009±0.001[rad]	0.101±0.00[rad/s]	50.9±0.4[N/m]
DRNN	lin	0.092±0.017[m]	0.018±0.004[m/s]	95.6±0.8[N]
	ang	0.012 ±0.003[rad]	0.018±0.003[rad/s]	16.7±1.6[N/m]

The desired trajectory was selected as a helical trajectory defined as $x_1 = [5\cos(0.1t)\text{m}, 5\sin(0.1t)\text{m}, 0.1t\ \text{m}, 0\ \text{rad}, 0\ \text{rad}, -0.05t\ \text{rad}]^\top$. The control gains were selected as $b = 1$, $k_r = 2$, $\alpha = 5$, $\beta_1 = 0.001$, $\beta_2 = 0.001$, and $\Gamma = 0.5 \cdot I_{592}$. The simulation was run for 150 s with a step size of 0.001 s and initial conditions $x_1 = [4\ \text{m}, 0.5\ \text{m}, 0\ \text{m}, 0\ \text{rad}, 0.2\ \text{rad}, 0\ \text{rad}]^\top$ and $x_2 = [0\ \text{m}, 0\ \text{m}, 0\ \text{m}, 0\ \text{rad}, 0\ \text{rad}, 0\ \text{rad}]^\top$.

The DRNN was composed of $k = 8$ layers with eight neurons in each layer and hyperbolic tangent activation functions. The SRNN baseline used the same controller and observer design, but with $k = 2$ layers and $L = 17$ neurons to ensure that approximately the same number of individual weights was used for both RNNs. The second comparison simulation used a central difference observer with the controller in (17) without the feedforward DNN, i.e., $u = g(x_1)^+[\ddot{x}_{d,1} - (k_r + \alpha)(\hat{\dot{e}}_1 + \alpha\hat{e}_1) - \alpha^2 e_1]$. For a fair comparison, the same robust control gains were used for all three simulations. The SRNN and DRNN gains and parameters (e.g., the learning gains, activation functions, and depth and width of the network) were empirically adjusted to achieve the best performance for both networks. To better emulate real-world systems and demonstrate robustness, noise was added to the position measurements from a uniform distribution of $U(-0.001, 0.001)$, and the simulations were performed for 50 iterations where the initial condition of the unknown state was randomly selected from a uniform distribution of $U(-0.5, 0.5)$.[7] The simulations were performed on a Dell Inspiron 7373 laptop on Windows 10 with 16.0 GB RAM and a 64-bit operating system. To further emulate a real-world system, the control input was saturated at 200 N or 200 N/m. To emulate a worst-case scenario, the RNN weight estimates were initialized from a uniform distribution of $U(-0.5, 0.5)$. However, for a system where pretraining is feasible, the weights can be initialized using traditional machine learning-based offline training methods.

The average and standard deviation performance results of the 50 iterations for the three methods are presented in Table II, and comparative plots of an example representative run are shown in Figs. 2–4. The average run times for the three OFB controllers were 46.17, 45.94, and 45.39 s for the CD observer, SRNN, and DRNN-based OFB controllers, respectively. As evident in Fig. 2, the RNN-based OFB controllers yield significant improvements in the estimation error

[7]To demonstrate the worst-case scenario of no prior data, the weights were randomly initialized. For further optimal tuning, Monte Carlo-based approaches can be used.

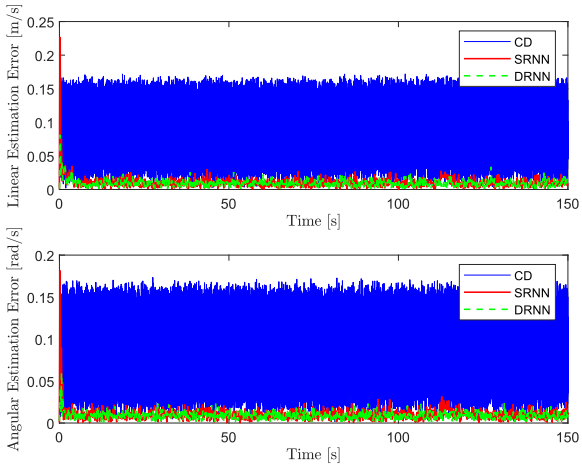

Fig. 2. Plots of the norm of the linear (top) and angular (bottom) estimation errors over time for one run of the DRNN OFB controller compared to the SRNN OFB controller and CD observer.

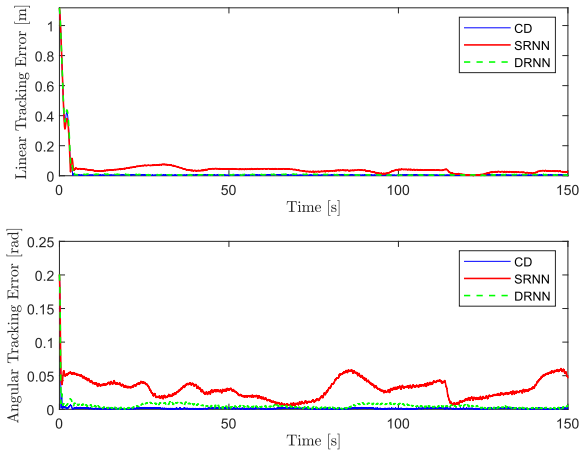

Fig. 3. Plots of the norm of the linear (top) and angular (bottom) tracking errors over time for one run of the DRNN OFB controller compared to the SRNN OFB controller and CD observer.

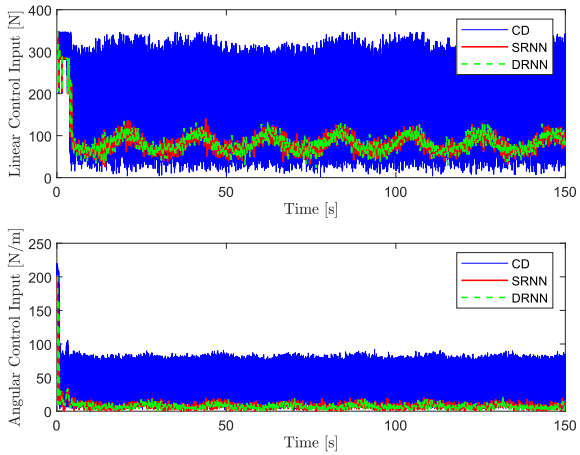

Fig. 4. Plots of the norm of the linear (top) and angular (bottom) control input over time for one run of the DRNN OFB controller compared to the SRNN OFB controller and CD observer.

performance when compared to the central difference observer. Specifically, the DRNN-based OFB controller on average yielded an 82.17% and 82.17% improvement in linear and angular estimation errors, respectively, when compared to the CD observer. The CD observer was significantly less robust to measurement noise and therefore yielded chattering, oscillatory behavior in the estimation error, which degraded steady-state performance. This oscillatory behavior in the estimation errors led to similar behavior in the control inputs in Fig. 4, and also created chattering behavior in the tracking error (as shown in Fig. 3). Thus, the DRNN-based observer design was significantly more robust to measurement noise with significantly less control effort (50.97% and 67.9% reduction in linear and angular control effort, respectively).

While the SRNN-based OFB controller yielded noticeably better estimation error performance when compared to the CD observer, the DRNN-based OFB controller improved the normalized estimation error by an average of 76.00% and 70.97% for the linear and angular estimation errors, respectively across the 50 iterations. Moreover, the DRNN-based OFB controller yielded significant improvements in the tracking error performance. As shown in Fig. 3, both OFB controllers yielded similar transient tracking error performance and settled after approximately 5 s. However, the DRNN architecture significantly improved steady-state behavior and the tracking error for the DRNN converged to a considerably smaller value than that of the SRNN. Specifically, the DRNN architecture improved the linear and angular tracking error by 31.85% and 86.21%, respectively, when compared to the shallow architecture with a comparable control effort.

VI. Conclusion

An adaptive Lb-DRNN-based OFB controller is developed for a class of uncertain nonlinear systems. Motivated by the dynamic nature and internal memory structure of RNNs, an Lb-DRNN observer is developed to adaptively estimate the unknown states of the system, which are implemented in the developed control design. Stability-driven weight adaptation laws adjust the weights of the Lb-DRNN architecture online using Lyapunov-based techniques. A Lyapunov-based stability analysis ensures asymptotic tracking and estimation error convergence using the developed observer and OFB controller. Validation simulation experiments on a UUV system yielded a 31.85% and 86.21% improvement in linear and angular tracking error, respectively, when compared to the shallow architecture with a comparable control effort.

Potential challenges associated with this work include practical concerns, such as computational resources, noise sensitivity, and guidelines, for sizing the network architecture. While the computational cost is higher when compared to the CD approach, the CD approach resulted in significantly worse performance and more noise sensitivity. Moreover, previous theoretical and empirical evidence indicates that the structure of the traditional RNN architecture used in the developed OFB controller inhibits its ability to learn long-term time dependencies. Therefore, motivation exists to investigate adaptive OFB controllers using other RNN architecture, such as long short-term memory (LSTM) networks. Thus, potential future research could include extending the developed LSTM controller and observer design in [6] and [38] to develop an OFB controller. Extending the developed OFB control design to adaptive LSTMs would bring additional challenges because the developed adaptation laws for the LSTM are more complex due to the more nonlinear structure of LSTMs compared to traditional DRNNs. Developing an adaptive LSTM-based OFB controller would involve incorporating the twofold control objective of the OFB control design into the online training of the LSTM by integrating both the tracking and state estimation error into the weight adaptation laws. Since the estimation error is unknown, a dynamic filter similar to the one in (9) could be used to construct auxiliary errors that would be implementable in the weight adaptation law. Thus, the developments in this article could pave the way for constructing an LSTM-based OFB controller.

Acknowledgment

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

References

[1] 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*, 2022, pp. 3487–3492.
[2] 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.
[3] R. Sun, M. Greene, D. Le, Z. Bell, G. Chowdhary, and W. E. Dixon, "Lyapunov-based real-time and iterative adjustment of deep neural networks," *IEEE Control Syst. Lett.*, vol. 6, pp. 193–198, 2022.
[4] F. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural net robot controller: Structure and stability proofs," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 388–399, Mar. 1996.
[5] 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*, 2022, pp. 3475–3480.
[6] E. Griffis, O. Patil, Z. Bell, and W. E. Dixon, "Lyapunov-based long short-term memory (Lb-LSTM) neural network-based control," *IEEE Control Syst. Lett.*, vol. 7, pp. 2976–2981, 2024.
[7] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 4, pp. 589–600, Jul. 1998.
[8] N. Hovakimyan, F. Nardi, A. Calise, and N. Kim, "Adaptive output feedback control of uncertain nonlinear systems using single-hidden-layer neural networks," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1420–1431, Nov. 2002.
[9] A. J. Calise, N. Hovakimyan, and M. Idan, "Adaptive output feedback control of nonlinear systems using neural networks," *Automatica*, vol. 37, no. 8, pp. 1201–1211, 2001.
[10] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," 2014, arXiv:1410.5401.
[11] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proc. Int. Conf. Mach. Learn.*, ser. ICML'16. JMLR.org, 2016, pp. 1842–1850.
[12] E. Parisotto and R. Salakhutdinov, "Neural map: Structured memory for deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent*., 2018.
[13] D. Muthirayan and P. P. Khargonekar, "Memory augmented neural network adaptive controllers: Performance and stability," *IEEE Trans. Autom. Control*, vol. 68, no. 2, pp. 825–838, Feb. 2023.
[14] A. Nikolakopoulou, M. S. Hong, and R. D. Braatz, "Dynamic state feedback controller and observer design for dynamic artificial neural network models," *Automatica*, vol. 146, 2022, Art. no. 110622.
[15] E. Griffis, O. Patil, W. Makumi, and W. E. Dixon, "Deep recurrent neural network-based observer for uncertain nonlinear systems," in *IFAC World Congr*, vol. 56, no. 2, 2023, pp. 6851–6856.
[16] H. Dinh, S. Bhasin, R. Kamalapurkar, and W. E. Dixon, "Dynamic neural network-based output feedback tracking control for uncertain nonlinear systems," *ASME J. Dyn. Syst. Meas. Control*, vol. 139, no. 7, pp. 074502–1–074502–7, Jul. 2017.
[17] H. T. Dinh, R. Kamalapurkar, S. Bhasin, and W. E. Dixon, "Dynamic neural network-based robust observers for uncertain nonlinear systems," *Neural Netw.*, vol. 60, pp. 44–52, Dec. 2014.
[18] P. Ioannou and J. Sun, *Robust Adaptive Control*. Englewood Cliffs, NJ, USA: Prentice Hall, 1996.
[19] Y. H. Kim and F. L. Lewis, "Neural network output feedback control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 301–309, Apr. 1999.
[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[21] D. Rolnick and M. Tegmark, "The power of deeper networks for expressing natural functions," in *Proc. Int. Conf. Learn. Represent*, 2018.

[22] D. S. Bernstein, *Matrix Mathematics*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
[23] B. E. Paden and S. S. Sastry, "A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators," *IEEE Trans. Circuits Syst.*, vol. AC-34, no. 1, pp. 73–82, Jan. 1987.
[24] E. Kosmatopoulos, M. Polycarpou, M. Christodoulou, and P. Ioannou, "High-order neural network structures for identification of dynamical systems," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 422–431, Mar. 1995.
[25] R. Pascanu, C. Gulcehre, and K. Cho, "How to construct deep recurrent neural networks," in *Proc. Int. Conf. Learn. Rep*., 2014.
[26] P. Kidger and T. Lyons, "Universal approximation with deep narrow networks," in *Proc. Conf. Learn. Theory*, 2020, pp. 2306–2327.
[27] V. Stepanyan and A. Kurdila, "Asymptotic tracking of uncertain systems with continuous control using adaptive bounding," *IEEE Trans. Neur. Netw.*, vol. 20, no. 8, pp. 1320–1329, Aug. 2009.
[28] B. Xian, M. S. de Queiroz, D. M. Dawson, and M. McIntyre, "A discontinuous output feedback controller and velocity observer for nonlinear mechanical systems," *Automatica*, vol. 40, no. 4, pp. 695–700, 2004.
[29] H. T. Dinh, S. Bhasin, D. Kim, and W. E. Dixon, "Dynamic neural network-based global output feedback tracking control for uncertain second-order nonlinear systems," in *Proc. Am Control Conf.*, Montréal, ON, Canada, Jun. 2012, pp. 6418–6423.
[30] D. Luenberger, "Observing the state of a linear system," *IEEE Trans. Mil. Electron.*, vol. ME-8, no. 2, pp. 74–80, Apr. 1964.
[31] A. F. Filippov, *Differential Equations With Discontinuous Right-Hand Side*. Norwell, MA, USA: Kluwer Academic Publishers, 1988.
[32] F. H. Clarke, *Optimization and Nonsmooth Analysis*. Philadelphia, PA, USA: SIAM, 1990.
[33] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 39 no. 9, pp. 1910–1914, Sep. 1994.
[34] O. Patil, A. Isaly, B. Xian, and W. E. Dixon, "Exponential stability with RISE controllers," *IEEE Control Syst. Lett.*, vol. 6, pp. 1592–1597, 2022.
[35] N. Fischer, R. Kamalapurkar, and W. E. Dixon, "LaSalle-Yoshizawa corollaries for nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2333–2338, Sep. 2013.
[36] 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, no. 4, pp. 845–852, Aug. 2014.
[37] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Boston, MA, USA: Birkhauser, 2003.
[38] E. Griffis, O. Patil, R. Hart, and W. E. Dixon, "Lyapunov-based long short-term memory (Lb-LSTM) neural network-based adaptive observer," *IEEE Control Syst. Lett.*, vol. 8, pp. 97–102, 2024.