Lyapunov-Based Long Short-Term Memory (Lb-LSTM) Neural Network-Based Adaptive Observer

Emily J. Griffis, *Graduate Student Member, IEEE*, Omkar Sudhir Patil, Rebecca G. Hart, *Graduate Student Member, IEEE*, and Warren E. Dixon, *Fellow, IEEE*

***Abstract*—Long short-term memory (LSTM) neural networks excel at capturing short- and long-term dependencies, making them powerful tools for system identification and state estimation. Their unique design improves memory capabilities by retaining important information and discarding irrelevant data over time. However, due to mathematical challenges involved in developing adaptive control methods for LSTMs, their training is predominantly limited to offline methods. This letter develops a Lyapunov-based (Lb-) LSTM observer for state estimation in nonlinear systems. The Lb-LSTM weights adapt in real-time using Lyapunov-based stability-driven adaptation laws. A nonsmooth Lyapunov-based stability analysis ensures state estimation error convergence and stability of the overall Lb-LSTM architecture. To validate the developed observer design, simulations were performed to estimate the unknown angular velocity states of a two-link robot manipulator. The developed method yielded a 41.13% improvement in the root mean square estimation error when compared to an adaptive RNN observer.**

***Index Terms*—Long short-term memory, neural networks, Lyapunov methods, adaptive control, nonlinear control systems.**

I. Introduction

A KEY element of human learning and cognition is the ability to remember important information. In an effort to leverage the pivotal impact of memory, prior work in [1] proposed a control method to incorporate external memory within a feedforward neural network (NN)- based controller. This inclusion of memory into static NN structures has been shown to accelerate learning and improve function approximation capabilities. Although a working memory can improve the overall performance of NNs, they remain feedforward and static, and therefore exhibit no dynamic behavior or internal memory capabilities.

In contrast, recurrent neural networks (RNNs) exhibit dynamic behavior due to an internal memory inherent to the architecture's design. This enables RNNs to capture time-evolving and cumulative effects in dynamic systems, which make them capable of learning the internal dynamics of the system [2], [3], [4]. However, both theoretical and empirical findings indicate that although conventional RNN structures can learn general dynamic behavior, they struggle to learn long-term time dependencies [5]. This limitation is attributed to the vanishing gradient problem, which in RNNs, causes the weights associated with long-term state information to become increasingly small as they are backpropagated through time.

Aiming to enhance the memory capabilities of RNNs, long short-term memory (LSTM) NNs are designed using an explicit memory mechanism and additional gate units. These features allow LSTMs to excel in learning both short- and long-term time-dependencies and demonstrate superior memory capabilities and performance compared to their traditional RNN counterparts. Thus, LSTMs are well-suited for estimating dynamical systems where an accurate representation of accumulative effects is advantageous. Moreover, the additional gating mechanisms augment learning of the internal dynamics and state estimation [6]. However, the implementation of LSTMs is largely dependent either on offline training methods [7], [8], [9], [10], [11], or training methods that only update a portion of the weights online [12], whereas adaptive control results are mostly restricted to simpler NN architectures [2], [3], [13], [14], [15], [16]. Although offline training techniques have well-established empirical success, they often require extensive and sufficiently rich datasets. Furthermore, they struggle to adapt to disturbances in real-time due to their lack of continual learning. Compared to offline learning approaches, online stability-driven methods consider real-time data in a closed-loop framework and provide stability guarantees. Our previous work in [6] develops Lyapunov-based stability-driven weight adaptation laws for all weights of the LSTM cell. However, this result only considers a tracking control problem and assumes full-state feedback. Thus, an adaptive LSTM architecture for real-time state estimation has

Manuscript received 13 September 2023; revised 21 November 2023; accepted 14 December 2023. Date of publication 29 December 2023; date of current version 22 January 2024. This work was supported in part by the Air Force Research Laboratory (AFRL) under Grant FA8651-21-F-1027 and Grant FA8651-21-F-1025; in part by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-19-1-0169; and in part by the Office of Naval Research under Grant N00014-21-1-2481. Recommended by Senior Editor A. P. Aguiar. *(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; rebecca.hart@ufl.edu; wdixon@ufl.edu).

Digital Object Identifier 10.1109/LCSYS.2023.3348706

not been considered. Developing an adaptive LSTM-based observer is technically challenging since the observer error is unknown and cannot be used in the adaptation laws.

In this letter, an adaptive Lyapunov-based (Lb-) LSTM architecture is designed and implemented in an observer to estimate unmeasurable states in a class of nonlinear systems. Specifically, the developed observer leverages the dynamic structure of LSTMs to produce an adaptive estimate of the unknown system states. Since the unknown observer error is not available for online learning, a dynamic filter is designed to construct an auxiliary error that is implementable in the weight adaptation law. Despite the challenges posed by the complex structure of the LSTM cell, a continuous-time representation of the LSTM architecture is constructed, and a Lyapunov stability-driven adaptation law is developed for all weights of the LSTM. To do so, Jacobians are computed of the LSTM cell dynamics with respect to the weights. Compared to typical offline LSTM methods (which can be used to provide an initial condition for the observer), the developed method provides lifelong, continued learning using an analytical adaptation law, thereby providing significantly more robustness against changes in parameters or reference signal. Thus, the developed Lb-LSTM observer is able to learn the system dynamics in real-time and adapt to model uncertainties without any offline training requirements. A nonsmooth Lyapunov-based stability analysis is performed that guarantees asymptotic convergence of the estimation errors and stability of the Lb-LSTM architecture. To validate the developed observer design, simulations were performed to estimate the angular velocity states of a two-link robot manipulator. The Lb-LSTM observer yielded a 41.13% improvement in the estimation error when compared to the adaptive shallow RNN observer in [2].

II. Problem Formulation

Notation and Preliminaries: An identity matrix of size n is denoted by $I_n \in \mathbb{R}^{n\times n}$. The Kronecker product is denoted $\otimes$. The Hadamard (element-wise) product is denoted by $\odot$ and satisfies the following properties [17, Definition 9.3.1]. Given any $b, c \in \mathbb{R}^n$, $b \odot c = D_b c$ and therefore, $\frac{\partial}{\partial c}(b \odot c) = D_b$, where $D_b \in \mathbb{R}^{n\times n}$ denotes a diagonal matrix with the vector b as its main diagonal. The function composition operator is denoted by $\circ$, i.e., given appropriate functions $f(\cdot)$ and $g(\cdot)$, $f \circ g(x) = f(g(x))$. The notation $\overset{a.a.t.}{(\cdot)}$ denotes the relation $(\cdot)$ holds for almost all time (a.a.t.). Let $h : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ denote a Lebesgue measurable and locally essentially bounded function. 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 $\dot{y} \overset{\text{a.a.t.}}{\in} \mathrm{K}[h](y, t)$, where $K[\,\cdot\,]$ denotes Filippov's differential inclusion [18]. Given some functions f and g and some $w \in \mathbb{R}$, the notation $f(w) = \mathcal{O}^m(g(w))$ means that $\|f(w)\| \leq M\|g(w)\|^m$ for all $w \geq w_0$, where $M \in \mathbb{R}_{>0}$ and $w_0 \in \mathbb{R}$ denote constants. The vectorization operator is denoted by $\mathrm{vec}(\cdot)$, i.e., given $A \triangleq [a_{i,j}] \in \mathbb{R}^{n\times m}$, $\mathrm{vec}(A) \triangleq [a_{1,1}, \ldots, a_{1,m}, \ldots, a_{n,1}, \ldots, a_{n,m}]^\top$. The p-norm is denoted by $\|\cdot\|_p$, where the subscript is suppressed when $p = 2$, and the Frobenius norm is denoted by $\|\cdot\|_F \triangleq \|\mathrm{vec}(\cdot)\|$. The vectorization operator satisfies the following properties [17, Proposition 7.1.9]. Given any $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{m\times p}$, and $C \in \mathbb{R}^{p\times r}$, $\mathrm{vec}(ABC) = (C^\top \otimes A)\mathrm{vec}(B)$, and therefore $\frac{\partial}{\partial \mathrm{vec}(B)}\mathrm{vec}(ABC) = C^\top \otimes A$.

A. System Dynamics

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

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= g(x, u), \end{aligned} \tag{1}$$

where $x \triangleq [x_1^\top \; x_2^\top]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^{2n}$ and $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ denote the generalized state and control input of the system, respectively, and $g : \mathbb{R}^{2n} \times \mathbb{R}^m \to \mathbb{R}^n$ denotes an unknown function. The following assumptions facilitate the subsequent observer development.

Assumption 1: The unknown function g is continuously differentiable.

Assumption 2: The system is assumed to be bounded-input bounded-output stable. Furthermore, the control input is assumed to be sufficiently smooth such that $\|u\| \leq \overline{u}$ and $\|\dot{u}\| \leq \overline{\dot{u}}$, where $\overline{u}, \overline{\dot{u}} \in \mathbb{R}_{>0}$ denote known constants. Therefore, the state can be bounded as $\|x\| \leq \overline{x}$, and there is a known, compact set $\mathcal{Z} \subseteq \mathbb{R}^{2n} \times \mathbb{R}^m$ such that $z \in \mathcal{Z}$, where $z \triangleq [x^\top \; u^\top]^\top$ and $\overline{x} \in \mathbb{R}_{>0}$ denotes a known constant.

Assumption 3: The system dynamics in (1) are observable.

Assumption 4: The state x_1 is assumed to be known.

III. Observer Development

Since only the first state x_1 is available for state feedback, the objective is to design an adaptive Lyapunov-based (Lb-) LSTM observer to estimate the unknown system dynamics. Let $\hat{x} \triangleq [\hat{x}_1^\top \; \hat{x}_2^\top]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^{2n}$ denote the observer state estimate. To quantify the objective of the observer, an estimation error $\tilde{x}_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and an auxiliary estimation error $r : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ are defined as

$$\tilde{x}_1 \triangleq x_1 - \hat{x}_1, \tag{2}$$

$$r \triangleq \dot{\tilde{x}}_1 + \alpha\tilde{x}_1 + \eta, \tag{3}$$

respectively, where $\alpha \in \mathbb{R}_{>0}$ denotes a user-selected constant and $\eta : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ denotes the output of a dynamic filter designed to compensate for the lack of availability of r since x_2 is unknown. Based on the subsequent stability analysis, the dynamic filter is designed as [2]

$$\begin{aligned} \eta &\triangleq p - (\alpha + k_r)\tilde{x}_1, \\ \dot{p} &\triangleq -(k_r + 2\alpha)p - \nu + \Big((\alpha + k_r)^2 + 1\Big)\tilde{x}_1, \\ \dot{\nu} &\triangleq p - \alpha\nu - (\alpha + k_r)\tilde{x}_1, \end{aligned} \tag{4}$$

where $k_r \in \mathbb{R}_{>0}$, $p : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, and $\nu : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ denote a user-defined constant, an internal filter variable, and auxiliary filter output, respectively. The filter variables p and ν are initialized such that $p(0) = (\alpha + k_r)\tilde{x}_1(0)$ and $\nu(0) = 0$. The developed dynamic filter in (4) uses $\tilde{x}_1$ as an input, yielding the filter outputs ν and η. The internal variable p of the filter is utilized to generate the output η, circumventing the need for the unmeasurable derivative of the estimation error $\dot{\tilde{x}}_1$.

[1]The development in this letter is restricted to second-order systems for the ease of illustration, but can be extended for n^{th} order systems using the observer development in [3].

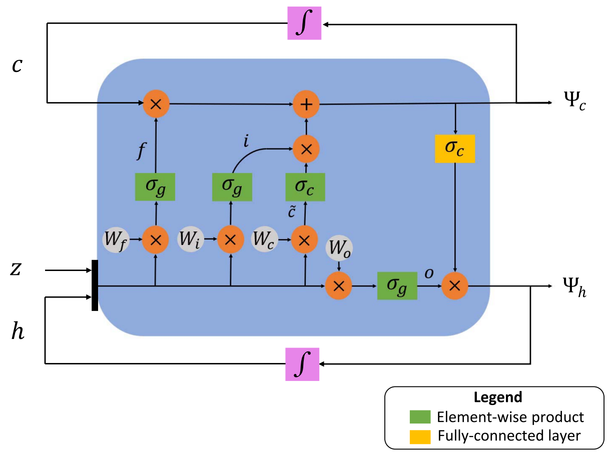

Fig. 1. LSTM model in (7), where the box represents the LSTM cell [6].

From (3) and (4), the dynamic filter can be related to the unmeasurable auxiliary estimation error r as

$$r = \dot{e} + \alpha e, \tag{5}$$

where $e : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is an auxiliary error defined as

$$e \triangleq \tilde{x}_1 + \nu. \tag{6}$$

A. Adaptive Long Short-Term Memory (LSTM) Architecture

Previous findings demonstrate that incorporating a memory capable of accessing prior state information in a NN architecture both reduces data requirements for training and accelerates learning [1], [19], [20], [21]. Unlike feedforward NNs, RNNs possess an internal memory enabling them to capture sequential dependencies and enhance function approximation performance. Traditional RNNs struggle with learning long-term dependencies, but LSTMs, equipped with three additional gate units (the input, forget, and output gates), excel at learning and retaining such dependencies. This makes them well-suited for estimating dynamic system states, where long-term memory and accurate representation of accumulative effects are crucial for making informed predictions. Thus, integrating an LSTM model into the observer design can improve predictive accuracy and enable robust modeling. Based on the design of continuous-time RNNs in works such as [22], and using Euler's method, an LSTM NN (see Fig. 1) can be modeled in continuous-time as[2]

$$\begin{aligned} f\big(\zeta, W_f\big) &= \sigma_g \circ W_f^\top \zeta, \\ i(\zeta, W_i) &= \sigma_g \circ W_i^\top \zeta, \\ c^*(\zeta, W_c) &= \sigma_c \circ W_c^\top \zeta, \\ o(\zeta, W_o) &= \sigma_g \circ W_o^\top \zeta, \\ \dot{c} &= -b_c c + b_c \Psi_c(\zeta, c, \theta), \\ \dot{h} &= -b_h h + b_h \Psi_h(\zeta, c, \theta), \end{aligned} \tag{7}$$

[2]The LSTM cell architecture introduced in [5] operates in discrete-time and is transformed into a continuous-time model as depicted in (7). This conversion aims to adapt the LSTM cell to make it more appropriate for controlling continuous-time systems. The parameters b_c and b_h in (7) emerge from constructing the continuous-time model and can be adjusted to optimize the performance of the continuous-time LSTM cell.

where $b_c, b_h \in \mathbb{R}_{>0}$ denote user-selected constants. The concatenated state vector $\zeta \in \mathbb{R}^{l_1}$ is augmented with a 1 to incorporate a bias term and is defined as $\zeta \triangleq [z^\top, h^\top, 1]^\top$, where $z \in \mathbb{R}^{2n+m}$ denotes the LSTM input, $h \in \mathbb{R}^{l_2}$ denotes the hidden state, $l_1 \triangleq 2n + m + l_2 + 1$, and $l_2 \in \mathbb{R}_{>0}$ denotes the user-selected number of neurons in the weight matrices. The cell state is denoted by $c \in \mathbb{R}^{l_2}$, where $c(0) = h(0) = 0$, and the sigmoid and tanh activation functions are denoted by $\sigma_g : \mathbb{R}^{l_2} \to \mathbb{R}^{l_2}$ and $\sigma_c : \mathbb{R}^{l_2} \to \mathbb{R}^{l_2}$, respectively. The forget gate, input gate, cell gate, and output gate of the LSTM are denoted by $f(\zeta, W_f) \in \mathbb{R}^{l_2}$, $i(\zeta, W_i) \in \mathbb{R}^{l_2}$, $c^*(\zeta, W_c) \in \mathbb{R}^{l_2}$, and $o(\zeta, W_o) \in \mathbb{R}^{l_2}$, respectively, and the weight matrices are denoted by $W_c^\top, W_i^\top, W_f^\top, W_o^\top \in \mathbb{R}^{l_2 \times l_1}$, and $W_h^\top \in \mathbb{R}^{n \times l_2}$, where $\theta \triangleq [\mathrm{vec}(W_c)^\top, \mathrm{vec}(W_i)^\top, \mathrm{vec}(W_f)^\top, \mathrm{vec}(W_o)^\top, \mathrm{vec}(W_h)^\top]^\top \in \mathbb{R}^{4l_2 l_1 + l_2 n}$. The functions $\Psi_c(\zeta, c, \theta) \in \mathbb{R}^{l_2}$ and $\Psi_h(\zeta, c, \theta) \in \mathbb{R}^{l_2}$ in the cell and hidden state dynamics are defined as

$$\begin{aligned} &\Psi_c(\zeta, c, \theta) \triangleq f\big(\zeta, W_f\big) \odot c + i(\zeta, W_i) \odot c^*(\zeta, W_c), \\ &\Psi_h(\zeta, c, \theta) \triangleq o(\zeta, W_o) \odot (\sigma_c \circ \Psi_c(\zeta, c, \theta)), \end{aligned}$$

respectively. To ensure the output of the LSTM has the appropriate dimensions, a fully-connected layer is added to the LSTM cell using the output weight matrix W_h. Thus, the output of the LSTM $\Phi(\zeta, c, \theta) \in \mathbb{R}^n$ can be modeled as

$$\Phi(\zeta, c, \theta) = W_h^\top \Psi_h(\zeta, c, \theta). \tag{8}$$

Let $\mathcal{C}(\mathcal{Z})$ denote the space of continuous functions over the set $\mathcal{Z}$. The universal function approximation property states that the function space of (7) is dense in $\mathcal{C}(\mathcal{Z})$ [23, Th. 1.1], and therefore, for any prescribed $\overline{\varepsilon} \in \mathbb{R}_{>0}$, there exist ideal weight matrices θ such that $\|g(x, u) - \Phi(\zeta, c, \theta)\| \leq \overline{\varepsilon}$. Hence, the system dynamics $g(x, u)$ can be modeled using the LSTM architecture in (7) as $g(x, u) = \Phi(\zeta, c, \theta) + \varepsilon(z)$, where $\varepsilon : \mathbb{R}^{2n+m} \to \mathbb{R}^n$ denotes a function reconstruction error that can be bounded as $\|\varepsilon\|_{z \in \mathcal{Z}} \leq \overline{\varepsilon}$, where $\overline{\varepsilon} \in \mathbb{R}_{>0}$ denotes a bounding constant. The ideal weights are assumed to be bounded as $\|W_j\|_F \leq \overline{W}$ for all $j \in \{c, i, f, o, h\}$, where $\overline{W} \in \mathbb{R}_{>0}$ denotes a known constant [6]. Therefore, taking the time-derivative of (3) and using (1) and (2) yields

$$\dot{r} = \Phi(\zeta, c, \theta) + \varepsilon(z) - \dot{\hat{x}}_2 + \alpha \dot{\tilde{x}}_1 + \dot{\eta}, \tag{9}$$

where $\dot{\eta}$ can be determined by taking the time derivative of η and using (3) and (4) to yield

$$\dot{\eta} = -(\alpha + k_r) r - \alpha \eta + \tilde{x}_1 - \nu. \tag{10}$$

B. Observer Design

While LSTMs have improved memory capabilities compared to other NN architectures, their application has not been explored for real-time state estimation. Offline approaches remain static and do not allow updates of the NN weights irrespective of system performance, resulting in a lack of robustness to uncertainty in the dynamics. In contrast, adaptive NN-based observers dynamically update the weights online through stability-driven methods. Motivated by the adaptability to changing conditions and improved robustness of adaptive NN architectures, a Lb-LSTM using the shorthand notation $\widehat{\Phi} \triangleq \Phi(\hat{\zeta}, \hat{c}, \hat{\theta})$ is constructed and an observer is designed as

$$\begin{aligned} \dot{\hat{x}}_1 &\triangleq \hat{x}_2, \\ \dot{\hat{x}}_2 &= \widehat{\Phi} + k_s \mathrm{sgn}(e) + \chi, \end{aligned} \tag{11}$$

where $k_s \in \mathbb{R}_{>0}$ denotes a user-selected constant, $\hat{\zeta} \triangleq [\hat{z}^\top, \hat{h}^\top, 1]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^{l_1}$, $\hat{z} \triangleq [\hat{x}^\top \ u^\top]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^{2n+m}$ denotes the input of the LSTM estimate, $\hat{\theta} \triangleq [\mathrm{vec}(\widehat{W}_c)^\top, \mathrm{vec}(\widehat{W}_i)^\top, \mathrm{vec}(\widehat{W}_f)^\top, \mathrm{vec}(\widehat{W}_o)^\top, \mathrm{vec}(\widehat{W}_h)^\top]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^{4l_2l_1+l_2n}$ denotes the adaptive weight estimates, and $\chi \in \mathbb{R}^n$ denotes an auxiliary term defined as $\chi \triangleq -(3\alpha + k_r)\eta + (\alpha^2 + 2)\tilde{x}_1 - \nu$. Thus, using the LSTM model in (7) and the adaptive weight estimates $\hat{\theta}$, the estimated cell state $\hat{c}$ and estimated hidden state $\hat{h}$ evolve according to

$$\dot{\hat{c}} = -b_c\hat{c} + b_c\Big(f\Big(\hat{\zeta}, \widehat{W}_f\Big) \odot \hat{c} + i\Big(\hat{\zeta}, \widehat{W}_i\Big) \odot c^*\Big(\hat{\zeta}, \widehat{W}_c\Big)\Big), \tag{12}$$

$$\dot{\hat{h}} = -b_h\hat{h} + b_h\Big(o\Big(\hat{\zeta}, \widehat{W}_o\Big) \odot \sigma_c \circ \Psi_c\Big(\hat{\zeta}, \hat{c}, \hat{\theta}\Big)\Big), \tag{13}$$

respectively. Substituting the observer in (11) into (9) and adding and subtracting $\Phi(\hat{\zeta}, \hat{c}, \theta)$ yields

$$\begin{aligned} \dot{\tilde{x}}_1 &= \dot{x}_1 - \dot{\hat{x}}_1, \\ \dot{r} &= \widetilde{\Phi} + \varepsilon(z) - k_s \mathrm{sgn}(e) - \chi + \alpha\dot{\tilde{x}}_1 + \dot{\eta} + N_1, \end{aligned} \tag{14}$$

where $\widetilde{\Phi} \triangleq \Phi(\hat{\zeta}, \hat{c}, \theta) - \widehat{\Phi}$ and $N_1 \triangleq \Phi(\zeta, c, \theta) - \Phi(\hat{\zeta}, \hat{c}, \theta)$. Using the bounds on the tanh and sigmoid activation functions, the auxiliary function N_1 can be bounded as $\|N_1\| \leq C_1$, where $C_1 \triangleq 2\overline{W}\sqrt{l_2}$.

C. Weight Adaptation Laws

Based on the subsequent stability analysis, the weight adaptation law is designed as

$$\dot{\hat{\theta}} \triangleq \Gamma\widehat{\Phi}'^\top e, \tag{15}$$

where $\Gamma \in \mathbb{R}^{(4l_1l_2+l_2n)\times(4l_1l_2+l_2n)}$ denotes a user-selected positive-definite adaptation gain matrix and the short-hand notation $\widehat{\Phi}'$ denotes the Jacobian $\widehat{\Phi}' \triangleq \frac{\partial\widehat{\Phi}}{\partial\hat{\theta}}$.

The Jacobian $\widehat{\Phi}'$ can be represented as $\widehat{\Phi}' \triangleq [\widehat{\Phi}'_{W_c}, \widehat{\Phi}'_{W_i}, \widehat{\Phi}'_{W_f}, \widehat{\Phi}'_{W_o}, \widehat{\Phi}'_{W_h}]$, where $\widehat{\Phi}'_{W_j} \triangleq \frac{\partial\widehat{\Phi}}{\partial \mathrm{vec}(\widehat{W}_j)}$ for all $j \in \{c, i, f, o, h\}$. Using (8) and the chain rule, the Jacobians $\widehat{\Phi}'_{W_j}$ and $\widehat{\Phi}'_{W_h}$ can be expressed as $\widehat{\Phi}'_{W_j} = \widehat{W}_h^\top\widehat{\Psi}'_{h,W_j}$ and $\widehat{\Phi}'_{W_h} = I_n \otimes \Psi_h^\top(\hat{\zeta}, \hat{c}, \hat{\theta})$, for all $j \in \{c, i, f, o\}$, respectively, where $\widehat{\Psi}'_{h,W_j} \triangleq \frac{\partial\Psi_h(\hat{\zeta},\hat{c},\hat{\theta})}{\partial \mathrm{vec}(\widehat{W}_j)}$ $\forall j \in \{c, i, f, o\}$. Using (7), (12), and (13), the properties of the Hadamard product, the properties of the vectorization operator, and the chain rule, the terms $\widehat{\Psi}'_{h,W_j}$ and $\widehat{\Psi}'_{h,W_o}$ can be expressed as

$$\begin{aligned} \widehat{\Psi}'_{h,W_j} &= \mathrm{diag}\Big(\sigma_g\Big(\widehat{W}_o^\top\hat{\zeta}\Big)\Big)\sigma_c'\Big(\Psi_c\Big(\hat{\zeta}, \hat{c}, \hat{\theta}\Big)\Big)\widehat{\Psi}'_{c,W_j}, \\ \widehat{\Psi}'_{h,W_o} &= \mathrm{diag}\big(\sigma_c(\widehat{\Psi}_c)\big)\Big(\sigma_g'\Big(\widehat{W}_o^\top\hat{\zeta}\Big)\Big)\Big(I_{l_2} \otimes \hat{\zeta}^\top\Big), \end{aligned}$$

for all $j \in \{c, i, f\}$, respectively, where $\widehat{\Psi}'_{c,W_j} \triangleq \frac{\partial\Psi_c(\hat{\zeta},\hat{c},\hat{\theta})}{\partial \mathrm{vec}(\widehat{W}_j)}$ $\forall j \in \{c, i, f\}$. Likewise, using (7), (12), and (13), the terms $\widehat{\Psi}'_{c,W_c}$, $\widehat{\Psi}'_{c,W_i}$, and $\widehat{\Psi}'_{c,W_f}$ can be expressed as

$$\begin{aligned} \widehat{\Psi}'_{c,W_c} &= \mathrm{diag}\Big(\sigma_g\Big(\widehat{W}_i^\top\hat{\zeta}\Big)\Big)\sigma_c'\Big(\widehat{W}_c^\top\hat{\zeta}\Big)\Big(I_{l_2} \otimes \hat{\zeta}^\top\Big), \\ \widehat{\Psi}'_{c,W_i} &= \mathrm{diag}\Big(\sigma_c\Big(\widehat{W}_c^\top\hat{\zeta}\Big)\Big)\sigma_g'\Big(\widehat{W}_i^\top\hat{\zeta}\Big)\Big(I_{l_2} \otimes \hat{\zeta}^\top\Big), \\ \widehat{\Psi}'_{c,W_f} &= \mathrm{diag}\big(\hat{c}\big)\sigma_g'\Big(\widehat{W}_f^\top\hat{\zeta}\Big)\Big(I_{l_2} \otimes \hat{\zeta}^\top\Big), \end{aligned}$$

respectively, where $\sigma_j'(y) \triangleq \frac{\partial}{\partial z}\sigma_j(z)|_{z=y}$, $\forall j \in \{c, g\}$, $y \in \mathbb{R}^{l_2}$.

IV. Stability Analysis

NNs like the LSTM architecture introduced in (7) are nonlinear with respect to their weights. Furthermore, the LSTM model introduces increased complexity due to the inclusion of three gate units within its cell structure. To address the mathematical issues arising due to the nonlinear parameterization, a first-order Taylor Series approximation of the LSTM in (7) and (8) is constructed, given by $\widetilde{\Phi} = \widehat{\Phi}'\tilde{\theta} + \mathcal{O}^2(\tilde{\theta})$, where $\mathcal{O}^2(\tilde{\theta}) \in \mathbb{R}^n$ denotes the higher-order terms. Thus, substituting this into (14) yields the closed-loop error system

$$\begin{aligned} \dot{\tilde{x}}_1 &= \dot{x}_1 - \dot{\hat{x}}_1, \\ \dot{r} &= \widehat{\Phi}'\tilde{\theta} + N_2 - k_s\mathrm{sgn}(e) - \chi + \alpha\dot{\tilde{x}}_1 + \dot{\eta}, \end{aligned} \tag{16}$$

where $N_2 \triangleq N_1 + \mathcal{O}^2(\tilde{\theta}) + \varepsilon(z)$.

To facilitate the stability analysis, let a candidate Lyapunov function $\mathcal{V}_L : \mathbb{R}^\psi \to \mathbb{R}_{\geq 0}$ be defined as

$$\mathcal{V}_L(\xi) \triangleq \frac{1}{2}\eta^\top\eta + \frac{1}{2}\nu^\top\nu + \frac{1}{2}\tilde{x}_1^\top\tilde{x}_1 + \frac{1}{2}r^\top r + P + \frac{\alpha}{2}\tilde{\theta}^\top\Gamma^{-1}\tilde{\theta}, \tag{17}$$

where the concatenated state vector $\xi : \mathbb{R}_{\geq 0} \to \mathbb{R}^\psi$ is defined as $\xi \triangleq [\tilde{x}_1^\top, r^\top, \eta^\top, \nu^\top, \tilde{\theta}^\top, \sqrt{P}]^\top$, $\psi \triangleq 4n + 4l_1l_2 + l_2n + 1$, and $P : \mathbb{R}_{\geq 0} \to \mathbb{R}$ denotes a subsequently designed P-function. The candidate Lyapunov function in (17) can be bounded as $\beta_1\|\xi\|^2 \leq \mathcal{V}_L(\xi) \leq \beta_2\|\xi\|^2$, where $\beta_1 \triangleq \min\{\frac{1}{2}, \frac{\alpha}{2}\lambda_{\min}\{\Gamma\}\}$ and $\beta_2 \triangleq \max\{1, \frac{\alpha}{2}\lambda_{\max}\{\Gamma\}\}$. Let the open and connected sets $\mathcal{D} \subset \mathbb{R}^\psi$ and $\mathcal{S} \subset \mathbb{R}^\psi$ be defined as $\mathcal{D} \triangleq \{\varsigma \in \mathbb{R}^\psi : \|\varsigma\| < \sqrt{\frac{\beta_1}{\beta_2}}\omega\}$ and $\mathcal{S} = \{\varsigma \in \mathbb{R}^\psi : \|\varsigma\| < \omega\}$, respectively, where $\omega \in \mathbb{R}_{>0}$ denotes a bounding constant. The universal function approximation property only holds on the compact domain $\mathcal{Z}$. Therefore, the following stability analysis must guarantee $z(t) \in \mathcal{Z}$ for all $t \geq 0$ which is achieved by a stability result that constrains ξ to a compact domain, specifically that $\xi(t) \in \mathcal{S}$ for all $t \geq 0$ by initializing $\xi(0) \in \mathcal{D}$.

Taking the time-derivative of $\mathcal{V}_L$ using the chain rule for nonsmooth systems in [24, Th. 2.2], substituting in the closed-loop dynamics in (16), and canceling the coupling terms yields

$$\begin{aligned} \dot{\mathcal{V}}_L &\overset{a.a.t.}{\in} r^\top\Big(\widehat{\Phi}'\tilde{\theta} + N_2 - k_sK\big[\mathrm{sgn}\big](e) - \chi + \alpha\dot{\tilde{x}}_1 + \dot{\eta}\Big) \\ &\quad + \tilde{x}_1^\top\dot{\tilde{x}}_1 + \eta^\top\dot{\eta} + \nu^\top\dot{\nu} + \dot{P} - \alpha\tilde{\theta}^\top\Gamma^{-1}\dot{\hat{\theta}}. \end{aligned} \tag{18}$$

Substituting (6) and the weight adaptation law in (15) into (18) yields

$$\begin{aligned} \dot{\mathcal{V}}_L &\overset{a.a.t.}{\in} r^\top\Big(N_2 - k_sK\big[\mathrm{sgn}\big](e) - \chi + \alpha\dot{\tilde{x}}_1 + \dot{\eta} + N_1\Big) \\ &\quad + \tilde{x}_1^\top\dot{\tilde{x}}_1 + \eta^\top\dot{\eta} + \nu^\top\dot{\nu} + \dot{P} + \dot{e}^\top\widehat{\Phi}'\tilde{\theta}. \end{aligned} \tag{19}$$

Using the design of the dynamic filter in (4)-(6) and canceling like terms yields

$$\begin{aligned} \dot{\mathcal{V}}_L &\overset{a.a.t.}{\in} r^\top\big(N_2 - k_sK\big[\mathrm{sgn}\big](e) - k_r r\big) - (\alpha + k_r)\tilde{x}_1^\top\tilde{x}_1 \\ &\quad - \alpha\eta^\top\eta + \nu^\top(\alpha\tilde{x}_1 - \alpha\nu) + \dot{e}^\top\widehat{\Phi}'\tilde{\theta} + \dot{P}. \end{aligned} \tag{20}$$

Convergence of the estimation errors using the developed adaptive LSTM architecture and overall observer design is guaranteed in the following theorem.

To facilitate the subsequent stability analysis, let $N_3 \triangleq \widehat{\Phi}'\tilde{\theta}$. Using Assumptions 1 and 2, Lemma 1 in [6], and the facts that $N_2 + N_3 = g(x, u) - \Phi(\hat{\zeta}, \hat{c}, \hat{\theta})$ and the LSTM Φ is continuously differentiable by design, the bounds

$$\|N_2\| \leq \kappa_1, \|N_3\| \leq \kappa_2, \left\|\dot{N}_2 + \dot{N}_3\right\| \leq \kappa_3, \tag{21}$$

hold when $\xi \in \mathcal{S}$, where $\kappa_1, \kappa_2, \kappa_3 \in \mathbb{R}_{>0}$ are known positive bounding constants.

Theorem 1: Consider the system in (1). Let Assumptions 1-4 hold. The Lb-LSTM observer in (11) and the weight adaptation law in (15) ensure asymptotic estimation error convergence in the sense that $\|x_2 - \hat{x}_2\| \to 0$ as $t \to \infty$, provided $\xi(0) \in \mathcal{D}$, the following gain condition is satisfied.

$$k_s \geq \kappa_1 + \kappa_2 + \frac{1}{\alpha - 1}(\alpha\kappa_2 + \kappa_3),$$
$$\alpha > 1. \tag{22}$$

Proof: Consider the Lyapunov candidate function in (17). The P-function in (17) is designed as

$$P(t) \triangleq \mathrm{e}^{-t} * \left((\alpha - 1)\left(k_s\|e\|_1 - e^\top(N_2 + N_3)\right)\right) + k_s\|e\|_1 + \mathrm{e}^{-t} * \left(\alpha e^\top N_3 + e^\top(\dot{N}_2 + \dot{N}_3)\right) - e^\top(N_2 + N_3). \tag{23}$$

Using [25, Lemma 4] and (21), it can be shown that $P(t) \geq 0$ for all $t \geq 0$, provided the sufficient gain conditions in (22) are satisfied.

Therefore, substituting the time-derivative of (23) into (20) and using Young's inequality, (20) can be further bounded as

$$\dot{\mathcal{V}}_L \overset{a.a.t.}{\leq} -\lambda\|y\|^2,$$

when $\xi \in \mathcal{S}$, where $\lambda \triangleq \min\{k_r, \frac{\alpha}{2} + k_r, \alpha, \frac{\alpha}{2}, 1\}$ and $y \triangleq [r^\top, \tilde{x}_1^\top, \eta^\top, \nu^\top, \sqrt{P}]^\top$ denotes a concatenated state vector. To show $\xi \in \mathcal{S}$ for all $t \geq 0$, using the fact that $\dot{\mathcal{V}}_L(\xi(t)) \overset{a.a.t.}{\leq} 0$ and (17) implies $\xi(t)$ can be bounded as $\|\xi(t)\| \leq \sqrt{\frac{\beta_2}{\beta_1}}\|\xi(0)\|$ when $\xi \in \mathcal{S}$. Thus, if $\|\xi(0)\| \leq \omega\sqrt{\frac{\beta_1}{\beta_2}}$, then $\|\xi(t)\| \leq \omega$ for all $t \geq 0$. Therefore, if the states ξ are initialized such that $\xi(0) \in \mathcal{D}$, then $\xi \in \mathcal{S}$ for all $t \geq 0$. Since $\xi \in \mathcal{S}$ when $\xi(0) \in \mathcal{D}$, the bounds in (21) hold. To show $z \in \mathcal{Z}$ and the universal function approximation property holds, let the open and connected set $\Upsilon \subseteq \mathcal{Z}$ be defined as $\Upsilon = \{\varsigma \in \mathcal{Z} : \|\varsigma\| < \bar{x} + (3+\alpha)\omega + \bar{u}\}$. Using the fact that $\|\xi(t)\| \leq \omega$ for all $t \geq 0$, it can be shown that $\|\tilde{x}_1(t)\| \leq \omega$, $\|\eta(t)\| \leq \omega$, and $\|r(t)\| \leq \omega$ for all $t \geq 0$. Hence, using (2) and (3), $\hat{z}$ can be bounded as $\|\hat{z}\| \leq \bar{x} + (3 + \alpha)\omega + \bar{u}$. Therefore, if $\xi(0) \in \mathcal{D}$, then $\hat{z} \in \Upsilon \subseteq \mathcal{Z}$. Using (17) and the fact that $\dot{\mathcal{V}}_L \overset{a.a.t.}{\leq} 0$ implies $\tilde{x}_1, \nu, \eta, r, P, \tilde{\theta} \in \mathcal{L}_\infty$. Therefore, the observer $\hat{x} \in \mathcal{L}_\infty$ and $\hat{\theta} \in \mathcal{L}_\infty$. Since $\hat{x}, \hat{\theta} \in \mathcal{L}_\infty$ and the function Φ is continuously differentiable, $\dot{\hat{\theta}} \in \mathcal{L}_\infty$. The extension of LaSalle-Yoshizawa corollary in [26, Corollary 1] can be invoked to show $\|\tilde{x}_1\| \to 0$, $\|\nu\| \to 0$, $\|\eta\| \to 0$, and $\|r\| \to 0$ as $t \to \infty$. Therefore, using (3) and (4), it can be further shown that $\|x_2 - \hat{x}_2\| \to 0$ as $t \to \infty$. ∎

V. Simulation Results

Comparative simulations were performed to demonstrate the performance of the developed Lb-LSTM observer, where the results were compared to the adaptive shallow RNN observer in [2]. Simulations were performed to estimate the unknown angular velocity states of a two-link robot manipulator, which is modeled as [27, eq. (80)]

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = M^{-1}(x_1)((-V(x_1, x_2) - F_d)x_2 + u) - F_s(x_2), \tag{24}$$

where $x_1 \triangleq [x_{11}\ \ x_{12}]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^2$ and $x_2 \triangleq [x_{21}\ x_2]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^2$ denote the angular position and velocity of the two links, respectively, and $F_d \in \mathbb{R}^{2\times2}$, $F_s : \mathbb{R}^2 \to \mathbb{R}^{2\times2}$, $M : \mathbb{R}^2 \to \mathbb{R}^{2\times2}$, and $V : \mathbb{R}^4 \to \mathbb{R}^{2\times2}$ denote the dynamic friction, static friction, inertia matrix and the centripetal-Coriolis matrix, respectively, as defined in [27]. A proportional derivative (PD) controller was selected as $u = 15(x_1 - x_{1d}) + 5(\hat{x}_2 - x_{2d})$ to track the desired position trajectory $x_{d,1} = [\frac{\pi}{6}\sin(\frac{\pi}{2}t)\ \frac{\pi}{6}\sin(\frac{\pi}{2}t)]^\top$. Each simulation was performed for 50 seconds with a step size of 0.001 seconds, and noise was added to the joint angle measurements from a uniform distribution $U(-0.5, 0.5)$ [deg]. The observer and dynamic filter gains in (3), (4), and (11) were selected as $k_r = 20$, $k_s = 0.05$, $\alpha = 60$, and $b_c = b_h = 10$. For a fair comparison, the same robust gains and dynamic filter was used for both observers, and the comparative observer was constructed by replacing the LSTM estimate in (11) with the adaptive shallow RNN estimate developed in [2]. The LSTM and RNN estimates were composed of $l_2 = 12$ neurons each with $l_1 = 19$ for the LSTM. The LSTM and RNN weights were randomly selected from a uniform distribution $U(-2, 2)$, with learning gains of $\Gamma = 20 \cdot I_{4l_1l_2+l_2n}$ for the LSTM and $\Gamma_{W_f} = 20 \cdot I_{24}$ and $\Gamma_{V_{f1}} = 20 \cdot I_{96}$ for the shallow RNN. The performance results of the two simulations are shown in Table I and Figures 2 and 3.

The developed Lb-LSTM observer yielded a 41.13% improvement in the root mean square estimation error. While the estimation errors settled for both observers after approximately 1 s, the Lb-LSTM observer yielded a significant

TABLE I
Performance Comparison

Architecture	$\|x_2 - \hat{x}_2\|$ [deg/s]	$\|\tilde{x}_1\|$ [deg]
RNN	0.3856	0.1065
LSTM	0.2270	0.0595
Percent Improvement	41.13%	44.09%

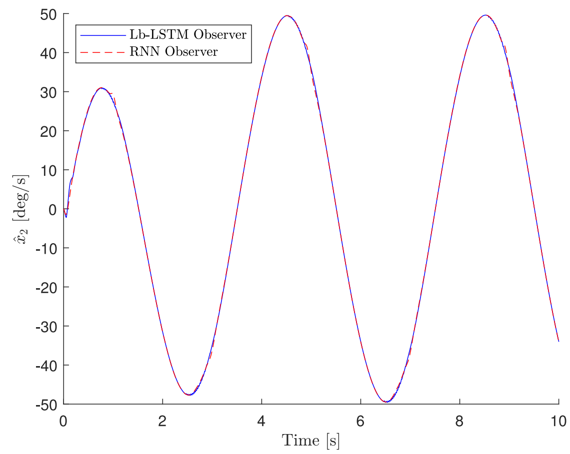

Fig. 2. Plot of the estimate $\hat{x}_2$ of the first robot link over time for the developed Lb-LSTM observer compared to the adaptive shallow RNN observer in [2]. For visual clarity, only the first 10 s of the simulation are shown.

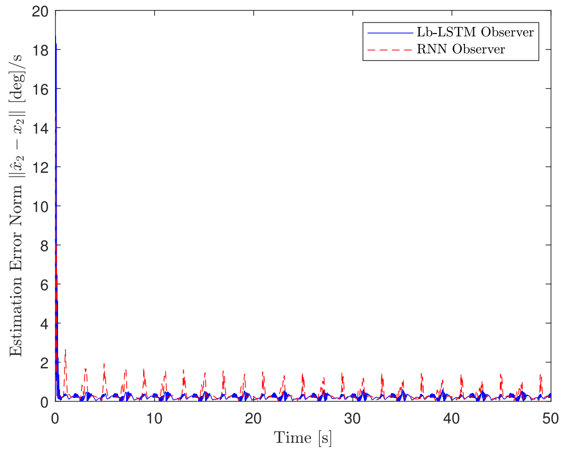

Fig. 3. Plot of the estimation error norm $\|\hat{x}_2 - x_2\|$ over time for the developed Lb-LSTM observer compared to the adaptive shallow RNN observer in [2].

improvement in the steady state performance. As evident from Figure 3, the adaptive shallow RNN observer produced small, frequent spikes in the estimation error, which contributed to a higher root mean square estimation error. Ultimately, the developed Lb-LSTM architecture and adaptive observer design resulted in significant improvements in estimation accuracy of the unknown state x_2 when compared to the baseline adaptive shallow RNN observer.

VI. Conclusion

In this letter, an Lb-LSTM observer is designed for nonlinear system state estimation. The developed Lb-LSTM architecture adapts in real-time through Lyapunov stability-driven adaptation laws. A nonsmooth Lyapunov-based stability analysis is performed to guarantee convergence of the state estimation error and stability of the overall Lb-LSTM observer design. Comparative simulations are provided to estimate the unknown angular velocity states of a two-link robot manipulator. The simulation results show the developed method yielded a 41.13% improvement in the root mean square estimation error when compared to the adaptive shallow RNN observer in [2]. Future work will incorporate the developed adaptive Lb-LSTM observer into an output-feedback control architecture. Additional efforts could implement the developed adaptive LSTM observer for open-loop future state prediction or for intermittent state feedback problems.

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] 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.

[2] 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.

[3] E. J. Griffis, O. S. Patil, W. A. Makumi, and W. E. Dixon, "Deep recurrent neural network-based observer for uncertain nonlinear systems," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 6851–6856, 2023.

[4] K.-I.Funahashi and Y. Nakamura, "Approximation of dynamic systems by continuous-time recurrent neural networks," *Neural Netw.*, vol. 6, no. 6, pp. 801–806, 1993.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Dec. 1997.

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

[7] X. Chang, L. Rong, K. Chen, and W. Fu, "LSTM-based output-constrained adaptive fault-tolerant control for fixed-wing UAV with high dynamic disturbances and actuator faults," *Math. Probl. Eng.*, vol. 2021, Mar. 2021, Art. no. 8882312.

[8] E. Inanc, Y. Gurses, A. Habboush, and Y. Yildiz, "Long short-term memory for improved transients in neural network adaptive control," in *Proc. Amer. Control Conf. (ACC)*, 2023, pp. 3614–3619.

[9] M. Y. Uzun, E. Inanc, A. Habboush, and Y. Yildiz, "Human-in-the-loop performance evaluation of an adaptive control framework with long short-term memory augmentation," in *Proc. Eur. Control Conf.*, 2023, pp. 1–6.

[10] N. Hirose and R. Tajima, "Modeling of rolling friction by recurrent neural network using LSTM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 6471–6478.

[11] S. Wen, Y. Wang, Y. Tang, Y. Xu, P. Li, and T. Zhao, "Real-time identification of power fluctuations based on LSTM recurrent neural network: A case study on singapore power system," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5266–5275, Sep. 2019.

[12] J. Wen, C. Zhao, and Z. Shi, "LSTM-based adaptive robust nonlinear controller design of a single-axis hydraulic shaking table," *IET Control Theory Appl.*, vol. 17, no. 7, pp. 825–836, 2023.

[13] 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 61st Conf. Decis. Control*, 2022, pp. 3487–3492.

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

[15] D. M. Le, O. S. Patil, C. F. Nino, and W. E. Dixon, "Accelerated gradient approach for neural network-based adaptive control of nonlinear systems," in *Proc. 61st IEEE Conf. Decis. Control*, 2022, pp. 3475–3480.

[16] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 703–715, May 1995.

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

[18] 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. 34, no. 1, pp. 73–82, Jan. 1987.

[19] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," 2014, *arXiv:1410.5401*.

[20] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1842–1850.

[21] E. Parisotto and R. Salakhutdinov, "Neural map: Structured memory for deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.

[22] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural network structures for identification of dynamical systems," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 422–431, Mar. 1995.

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

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

[25] O. S. Patil, A. Isaly, B. Xian, and W. E. Dixon, "Exponential stability with RISE controllers," *IEEE Control Syst. Lett.*, vol. 6, pp. 1592–1597, 2022.

[26] 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.

[27] M. S. De Queiroz, J. Hu, D. M. Dawson, T. Burg, and S. R. Donepudi, "Adaptive position/force control of robot manipulators without velocity measurements: Theory and experimentation," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 27, no. 5, pp. 796–809, Oct. 1997.