

# Lyapunov-Based Long Short-Term Memory (Lb-LSTM) Neural Network-Based Control

Emily J. Griffis<sup>ID</sup>, Omkar Sudhir Patil<sup>ID</sup>, Zachary I. Bell<sup>ID</sup>, and Warren E. Dixon<sup>ID</sup>, *Fellow, IEEE*

**Abstract**—Recurrent neural networks (RNNs) are a dynamic mapping that can capture time-varying, accumulative effects in a sequence that static, feedforward neural networks (NNs) cannot. Long short-term memory (LSTM) NNs are a type of RNN that have gained recent popularity because the cell structure allows them to retain long-term information more efficiently than traditional RNNs. Existing results develop LSTM-based controllers to compensate for uncertainties in nonlinear systems. However, these results use discrete-time LSTMs with offline-trained weights. In this letter, a Lyapunov-based LSTM controller is developed for general Euler-Lagrange systems. Specifically, an Lb-LSTM is implemented in the control design to adaptively estimate uncertain model dynamics, where the weight estimates of the LSTM cell are updated using Lyapunov-based adaptation laws. This allows the LSTM cell to adapt to system uncertainties without requiring offline training. A Lyapunov-based stability analysis yields uniform ultimate boundedness (UUB) of the tracking errors and LSTM state and weight estimation errors. Simulations indicate the developed Lb-LSTM-based controller yielded significant improvement in tracking and function approximation performance when compared to several DNN examples.

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

## I. INTRODUCTION

ADAPTIVE neural network (NN)-based controllers have become increasingly popular in recent years due to their real-time function approximation capabilities [1], [2], [3]. While most adaptive control results only consider single-hidden layer NNs, recent developments focus on deep learning with feedback control [1], [2], [3], [4]. However, the developed

Manuscript received 17 March 2023; revised 26 May 2023; accepted 19 June 2023. Date of publication 30 June 2023; date of current version 21 July 2023. 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 M. Guay. (Corresponding author: Emily J. Griffis.)

Emily J. Griffis, Omkar Sudhir Patil, and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32603 USA (e-mail: emilygriffis00@ufl.edu; patilomkarsudhir@ufl.edu; wdixon@ufl.edu).

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

Digital Object Identifier 10.1109/LCSYS.2023.3291328

adaptation methods are restricted to feedforward NNs, which are static structures and therefore only have access to current state information. Previous results in [5], [6], [7] establish that the presence of a memory capable of accessing previous state information both reduces the required data set for training and leads to faster learning. Motivated by the improved performance of NNs with access to previous state information, results in [8] augment static NN-based controllers with an external memory and show faster learning and improved function approximation performance. Although the results in [8] augment the NN with a working memory, the NN is feedforward and the augmented memory is external to the NN.

Unlike feedforward NNs, recurrent NNs (RNNs) are a dynamic mapping. Thus, RNNs have an internal memory that can leverage dependencies in a sequence and increase approximation capabilities, thus improving performance [9]. This internal memory allows RNNs to capture time-varying, accumulative effects exhibited in some dynamical systems that feedforward NNs cannot (cf., [4], [9], [10], [11], [12], [13], [14], [15], [16], [17]). However, theoretical and empirical evidence has shown that the structure of traditional RNNs inhibits their ability to learn long-term time dependencies. One type of RNNs, long short-term memory (LSTM) NNs, have a better ability to learn long term dependencies, and therefore, have improved memory capability when compared to traditional RNNs.

LSTMs have gained recognition in machine learning applications such as computer vision, natural language processing, sound recognition, and handwriting recognition due to their improved memory capability [18], [19], [20]. Specifically, LSTMs regulate the flow of the gradient along long time sequences by adding an explicit memory through three gate units: the input, forget, and output gates [21]. Compared to traditional RNNs, the addition of an explicit memory to the LSTM cell improves function approximation performance by retaining relevant information across each time step and forgetting irrelevant information stored in the internal memory [20]. Results such as [4], [15], [16] develop LSTM-based controllers and implement offline optimization techniques to train the weights of the LSTM based on some loss function. While these offline optimization techniques have been successfully implemented in empirical studies, they often require large, sufficiently rich data sets for training, and when used offline, are not able to adjust to disturbances in real-time due to the lack of sustained learning. In contrast to offline learning techniques, real-time stability-driven methods consider data online in a closed-loop implementation and provide stability guarantees.

While previous results develop Lyapunov-derived adaptation laws for some of the weights of a simplified LSTM cell structure, they employ offline optimization techniques to train the remaining weights [15]. Thus, deriving Lyapunov-based adaptation laws for LSTMs remains an open problem.

In this letter, an adaptive LSTM controller is developed for general Euler-Lagrange systems, where the adaptation law is derived from Lyapunov-based methods (hence, we refer to the architecture as Lb-LSTM). Specifically, a continuous-time Lb-LSTM NN is constructed and implemented in the controller as a feedforward term to adaptively estimate uncertain model dynamics. Despite the technical challenges posed by the complexity of the LSTM cell structure, stability-driven adaptation laws adjust the Lb-LSTM weights in real-time and allow the developed architecture to adapt to system uncertainties without any offline training requirements. A Lyapunov-based stability analysis is performed to guarantee uniform ultimate boundedness (UUB) of the tracking errors and LSTM state and weight estimation errors. To demonstrate the performance of the adaptive Lb-LSTM controller, simulations were performed and compared to the adaptive DNN-based controller in [2] using three different DNN architectures. The simulation results indicate significant improvements in tracking and function approximation performance when compared to various feedforward DNN architectures.

### A. Notation and Preliminaries

An  $n \times n$ -dimension identity matrix is denoted by  $I_n \in \mathbb{R}^{n \times n}$ . The Hadamard (element-wise) and Kronecker products are denoted by  $\odot$  and  $\otimes$ , respectively. 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 vectorization operator is denoted by  $\text{vec}(\cdot)$ , i.e., given  $A \triangleq [a_{i,j}] \in \mathbb{R}^{n \times m}$ ,  $\text{vec}(A) \triangleq [a_{1,1}, \dots, a_{1,m}, \dots, a_{n,1}, \dots, a_{n,m}]^\top$ . The Hadamard product satisfies the following properties [22, Definition 9.3.1]. Given any  $a, b \in \mathbb{R}^n$ ,  $a \odot b = D_a b$  and therefore,  $\frac{\partial}{\partial b}(a \odot b) = D_a$ , where  $D_a \in \mathbb{R}^{n \times n}$  denotes a diagonal matrix with the vector  $a$  as its main diagonal. The vectorization operator satisfies the following properties [22, Proposition 7.1.9]. Given any  $A \in \mathbb{R}^{n \times m}$ ,  $B \in \mathbb{R}^{m \times p}$ , and  $C \in \mathbb{R}^{p \times r}$ ,  $\text{vec}(ABC) = (C^\top \otimes A)\text{vec}(B)$ , and therefore  $\frac{\partial}{\partial \text{vec}(B)}\text{vec}(ABC) = (C^\top \otimes A)$ .

## II. SYSTEM DYNAMICS AND CONTROL OBJECTIVE

Consider a general uncertain Euler-Lagrange system modeled as

$$M(q)\ddot{q} + V_m(q)\dot{q} + F(\dot{q}) + G(q) = \tau, \quad (1)$$

where  $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$  denote the generalized position, velocity, and acceleration, respectively, and  $M: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ ,  $V_m: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ ,  $G: \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $F: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ , and  $\tau: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$  denote unknown, continuous generalized inertial effects, generalized centripetal-Coriolis effects, generalized vector of potential forces, generalized dissipation effects, and the control input, respectively. The model dynamics in (1) satisfy the following property.

*Property 1:* The inertial and centripetal-Coriolis effects satisfy  $\xi^\top(M(q) - 2V_m(q, \dot{q}))\xi = 0$  for all  $q, \dot{q}, \xi \in \mathbb{R}^n$ .

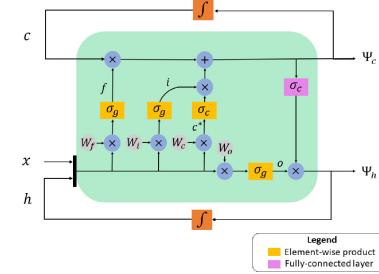


Fig. 1. LSTM model in (5), where the green box represents the LSTM cell.

The control objective is to design an adaptive Lb-LSTM controller to achieve UUB tracking of a desired trajectory  $q_d \in \mathbb{R}^n$ . To quantify the control objective, a tracking error  $e \in \mathbb{R}^n$  and an auxiliary tracking error  $r \in \mathbb{R}^n$  are defined as

$$e \triangleq q_d - q, \quad (2)$$

$$r \triangleq \dot{e} + \alpha e, \quad (3)$$

respectively, where  $\alpha \in \mathbb{R}_{>0}$  denotes a user-selected constant. The desired trajectory  $q_d$  is designed to be sufficiently smooth, i.e.,  $q_d, \dot{q}_d, \ddot{q}_d$  can be bounded as  $\|q_d\| \leq \bar{q}_d$ ,  $\|\dot{q}_d\| \leq \bar{\dot{q}}_d$ ,  $\|\ddot{q}_d\| \leq \bar{\ddot{q}}_d$ , where  $\bar{q}_d, \bar{\dot{q}}_d, \bar{\ddot{q}}_d \in \mathbb{R}_{>0}$  denote known constants and  $\dot{q}_d \in \mathbb{R}^n$  and  $\ddot{q}_d \in \mathbb{R}^n$  denote the first and second time-derivatives of  $q_d$ , respectively.

Taking the time-derivative of (3), multiplying by  $M(q)$ , and using (1) and (2) yields

$$M(q)\dot{r} = g(x) - \tau - V_m(q, \dot{q})r, \quad (4)$$

where  $x \triangleq [q^\top, \dot{q}^\top, q_d^\top, \dot{q}_d^\top, \ddot{q}_d^\top]^\top \in \mathbb{R}^{5n}$  denotes a concatenated vector and the function  $g(x) \in \mathbb{R}^n$  denotes the unknown system dynamics defined as  $g(x) \triangleq M(q)(\ddot{q}_d + \alpha\dot{e}) + V_m(q, \dot{q})(\dot{q}_d + \alpha e) + F(\dot{q}) + G(q)$ .

## III. CONTROL DEVELOPMENT

Long short-term memory (LSTM) NNs have grown in recent popularity due to their ability to leverage both long-term and short-term dependencies in time sequences for faster learning and improved performance. This motivates the development of an LSTM-based controller that can estimate and compensate for the unknown model dynamics in (4). Based on the design of continuous-time RNNs [23] and using Euler's method, an LSTM NN (see Fig. 1) can be modeled in continuous-time as [21]<sup>1,2</sup>

$$\begin{aligned} f(z, W_f) &= \sigma_g \circ W_f^\top z, & i(z, W_i) &= \sigma_g \circ W_i^\top z, \\ o(z, W_o) &= \sigma_g \circ W_o^\top z, & c^*(z, W_c) &= \sigma_c \circ W_c^\top z, \\ \dot{c} &= -b_c c + b_c \Psi_c(x, c, h, \theta), \\ \dot{h} &= -b_h h + b_h \Psi_h(x, c, h, \theta, W_o), \end{aligned} \quad (5)$$

<sup>1</sup>The LSTM cell architecture developed in [21] is in discrete-time and is converted to a continuous-time model in (5) to make it more appropriate for controlling a continuous-time system. The gains  $b_c$  and  $b_h$  in (5) are a result of constructing a continuous-time model and can be tuned accordingly to enhance the performance of the continuous-time LSTM.

<sup>2</sup>Like the hidden state  $h$ , the cell state  $c$  is passed from one time step to the next. The gate output  $c^*$  (typically referred to as  $\tilde{c}$  in literature) is not passed to the next time step and represents the output of one of the internal gates within the LSTM cell.

where  $b_c, b_h \in \mathbb{R}_{>0}$  denote user-selected constants and  $c \in \mathbb{R}^{l_2}$  and  $h \in \mathbb{R}^{l_2}$  denote the cell state and hidden state, respectively, where  $h(0) = c(0) = 0$  and  $l_2 \in \mathbb{R}_{>0}$  denotes the number of neurons. The concatenated state vector  $z \in \mathbb{R}^{l_1}$  is defined as  $z \triangleq [x^\top, h^\top, 1]^\top$ , where  $x \in \mathbb{R}^{5n}$  denotes the LSTM input and  $l_1 \triangleq 5n + l_2 + 1$ . The state  $z$  is augmented with a 1 to incorporate a bias term. The forget gate, input gate, cell gate, and output gate are denoted by  $f(z, W_f) \in \mathbb{R}^{l_2}$ ,  $i(z, W_i) \in \mathbb{R}^{l_2}$ ,  $c^*(z, W_c) \in \mathbb{R}^{l_2}$ , and  $o(z, W_o) \in \mathbb{R}^{l_2}$ , respectively. The sigmoid and tanh activation functions are denoted by  $\sigma_g: \mathbb{R}^{l_2} \rightarrow \mathbb{R}^{l_2}$  and  $\sigma_c: \mathbb{R}^{l_2} \rightarrow \mathbb{R}^{l_2}$ , respectively, and the weight matrices are denoted by  $W_f^\top, W_c^\top, W_i^\top, W_o^\top \in \mathbb{R}^{l_2 \times l_1}$ , where  $\theta \triangleq [W_c^\top, W_i^\top, W_f^\top]^\top \in \mathbb{R}^{l_2 \times 3l_1}$ . The functions  $\Psi_c(x, c, h, \theta) \in \mathbb{R}^{l_2}$  and  $\Psi_h(x, c, h, \theta, W_o) \in \mathbb{R}^{l_2}$  are defined as  $\Psi_c(x, c, h, \theta) \triangleq f(z, W_f) \odot c + i(z, W_i) \odot c^*(z, W_c)$  and  $\Psi_h(x, c, h, \theta, W_o) \triangleq o(z, W_o) \odot (\sigma_c \circ \Psi_c(x, c, h, \theta))$ , respectively. To ensure the output of the LSTM has the appropriate dimensions, a fully-connected layer is added to the LSTM cell. To add generality to the LSTM model, a feedforward component is added to the output of the LSTM. The resulting LSTM model allows for a direct transmission of the input information through the feedforward component while leveraging the internal memory capabilities of LSTMs. Thus, the output of the LSTM  $\Phi(x, c, h, \theta, W_o, W_h, W_{FF}) \in \mathbb{R}^n$  can be modeled as

$$\Phi = W_h^\top (\Psi_h(x, c, h, \theta, W_o) + \sigma \circ W_{FF}^\top x), \quad (6)$$

where  $\sigma: \mathbb{R}^{l_2} \rightarrow \mathbb{R}^{l_2}$  denotes a vector of smooth activation functions and  $W_h^\top \in \mathbb{R}^{n \times l_2}$  and  $W_{FF}^\top \in \mathbb{R}^{l_2 \times 5n}$  denote the output weight matrix and weight matrix of the feedforward NN component, respectively.

The universal function approximation property states that the function space of (5) is dense in  $\mathcal{C}(\mathcal{Z})$ , where  $\mathcal{C}(\mathcal{Z})$  denotes the space of continuous functions over the set  $\mathcal{Z} \subseteq \mathbb{R}^{l_1}$ , where  $z \in \mathcal{Z}$  [24, Th. 1.1].<sup>3</sup> Therefore, for any prescribed  $\bar{e} \in \mathbb{R}_{>0}$ , there exist ideal weight matrices  $W_c^\top, W_i^\top, W_f^\top, W_o^\top, W_h^\top$ , and  $W_{FF}^\top$  such that the system dynamics  $g(x)$  can be modeled using the LSTM architecture in (5) as

$$g(x) = \Phi(x, c, h, \theta, W_o, W_h, W_{FF}) + \varepsilon(x). \quad (7)$$

It is assumed that there exists a known constant  $\bar{W} \in \mathbb{R}_{>0}$  such that the ideal weights can be bounded as  $\|W_j\|_F \leq \bar{W}$  for all  $j \in \{c, i, f, o, h, FF\}$  [2].

To compensate for the unknown LSTM model dynamics in (5), auxiliary cell and hidden state estimation errors are introduced in this section. The auxiliary cell and hidden state estimation errors  $\tilde{c} \in \mathbb{R}^{l_2}$  and  $\tilde{h} \in \mathbb{R}^{l_2}$  are defined as

$$\tilde{c} \triangleq c - \hat{c} + \eta_c, \quad (8)$$

$$\tilde{h} \triangleq h - \hat{h} + \eta_h, \quad (9)$$

respectively, where  $\hat{c} \in \mathbb{R}^{l_2}$  and  $\hat{h} \in \mathbb{R}^{l_2}$  denote the estimated cell state and hidden state, respectively, and  $\eta_c \in \mathbb{R}^{l_2}$  and  $\eta_h \in \mathbb{R}^{l_2}$  are designed as

$$\dot{\eta}_c \triangleq -k_{1,c}\eta_c - K_{2,c}r, \quad (10)$$

$$\dot{\eta}_h \triangleq -k_{1,h}\eta_h - K_{2,h}r, \quad (11)$$

<sup>3</sup>Since the subspace of LSTMs in (6) involving the feedforward term  $W_h^\top \sigma \circ W_{FF}^\top x$  is dense in  $\mathcal{C}(\mathcal{Z})$ , the space of LSTMs is also dense.

where  $k_{1,c}, k_{1,h} \in \mathbb{R}_{>0}$  denote user-selected constants and  $K_{2,c}, K_{2,h} \in \mathbb{R}^{l_2 \times n}$  denote user-selected matrices.

The following lemma establishes boundedness properties of the cell state and hidden state of the LSTM model in (5), which is essential for the ensuing development.

*Lemma 1:* Consider the LSTM model in (5). The hidden state  $h$  and cell state  $c$  can be bounded as

$$\|h\| \leq \frac{b_h \sqrt{l_2}}{\sqrt{2(b_h - \frac{1}{2})}}, \|c\| \leq \frac{\sqrt{l_2}}{\sqrt{2(b_c - b_c \sqrt{l_2} - \frac{1}{2})}}.$$

*Proof:* Consider the hidden state dynamics in (5), where the input  $\Psi_h$  can be bounded as  $\|\Psi_h\| \leq \sqrt{l_2}$  by design of the sigmoid and tanh activation functions. Consider the candidate Lyapunov function  $\mathcal{V}_h: \mathbb{R}^{l_2} \rightarrow \mathbb{R}_{\geq 0}$  defined as  $\mathcal{V}_h(h) \triangleq \frac{1}{2}h^\top h$ . Taking the derivative, using (5), bounding, and applying the Gronwall inequality yields  $\mathcal{V}_h \leq \mathcal{V}_h(h(t_0)) \exp(-2(b_h - \frac{1}{2})(t - t_0)) + \frac{b_h^2 l_2}{4(b_h - \frac{1}{2})}$ . Therefore, provided  $b_h \geq \frac{1}{2}$ , initializing  $h$  as  $h(t_0) = 0$  yields  $\|h\| \leq \frac{b_h \sqrt{l_2}}{\sqrt{2(b_h - \frac{1}{2})}}$ . Similarly, to prove boundedness of the cell state

$c$ , consider the candidate Lyapunov function  $\mathcal{V}_c: \mathbb{R}^{l_2} \rightarrow \mathbb{R}_{\geq 0}$  defined as  $\mathcal{V}_c = \frac{1}{2}c^\top c$ . By design of the sigmoid and tanh activation functions,  $\|f\| \leq \sqrt{l_2}$  and  $\|i \odot c^*\| \leq \sqrt{l_2}$ . Taking the derivative of the candidate Lyapunov function, substituting the cell state dynamics in (5), bounding, and applying the Gronwall inequality yields  $\mathcal{V}_c(c(t)) \leq \mathcal{V}_c(c(t_0)) \exp(-2(b_c - b_c \sqrt{l_2} - \frac{1}{2})(t - t_0)) + \frac{l_2}{4(b_c - b_c \sqrt{l_2} - \frac{1}{2})}$ . Therefore, provided  $b_c \geq \frac{1}{2(1+\sqrt{l_2})}$ , initializing  $c$  as  $c(t_0) = 0$  yields  $\|c\| \leq \frac{\sqrt{l_2}}{\sqrt{2(b_c - b_c \sqrt{l_2} - \frac{1}{2})}}$ . ■

### A. Control Design

Let the adaptive estimates of the LSTM weights be denoted as  $\hat{\theta} \triangleq [\hat{W}_c^\top, \hat{W}_i^\top, \hat{W}_f^\top]^\top \in \mathbb{R}^{3l_1 \times l_2}$ ,  $\hat{W}_o^\top \in \mathbb{R}^{l_2 \times l_1}$ ,  $\hat{W}_h^\top \in \mathbb{R}^{n \times l_2}$ , and  $\hat{W}_{FF}^\top \in \mathbb{R}^{l_2 \times 5n}$ . Based on the adaptive weight estimates, an Lb-LSTM adaptive feedforward term  $\hat{\Phi} \triangleq \Phi(x, \hat{c}, \hat{h}, \hat{\theta}, \hat{W}_o, \hat{W}_h, \hat{W}_{FF})$  is constructed and the control input is designed as

$$\tau \triangleq \hat{\Phi} + k_r r - K_{2,c}\eta_c - K_{2,h}\eta_h + e, \quad (12)$$

where  $k_r, k_s \in \mathbb{R}_{>0}$  denote user-selected constants. Substituting the LSTM model in (7) and the control input in (12) into (4) yields the closed-loop error system

$$M(q)\dot{r} = \hat{\Phi} + j_e + \varepsilon(x) - V_m(q, \dot{q})(r) - k_r r + K_{2,c}\eta_c + K_{2,h}\eta_h - e, \quad (13)$$

where the function  $j_e(x, c, h, \theta, W_o, W_h) \in \mathbb{R}^n$  is defined as  $j_e \triangleq \Phi(x, c, h, \theta, W_o, W_h, W_{FF}) - \Phi(x, \hat{c}, \hat{h}, \theta, W_o, W_h, W_{FF})$ .

### B. Weight Adaptation Laws

Using the LSTM model in (5), the estimated cell state  $\hat{c}$  and estimated hidden state  $\hat{h}$  evolve according to

$$\dot{\hat{c}} = -b_c \hat{c} + b_c (f(\hat{z}, \hat{W}_f) \odot c + i(\hat{z}, \hat{W}_i) \odot c^*(\hat{z}, \hat{W}_c)), \quad (14)$$

$$\dot{\hat{h}} = -b_h \hat{h} + b_h (o(\hat{z}, \hat{W}_o) \odot \sigma_c \circ \Psi_c(x, \hat{c}, \hat{h}, \hat{\theta})), \quad (15)$$

respectively, where  $\hat{z} \triangleq [x^\top, \hat{h}^\top, 1]^\top : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n+l_2+1}$  denotes the augmented input of the LSTM estimate. To facilitate the subsequent stability analysis, let the shorthand notations  $\tilde{\Psi}_c \in \mathbb{R}^{l_2}$  and  $\tilde{\Psi}_h \in \mathbb{R}^{l_2}$  be defined as  $\tilde{\Psi}_c \triangleq \Psi_c(x, \hat{c}, \hat{h}, \theta) - \hat{\Psi}_c$ , and  $\tilde{\Psi}_h \triangleq \Psi_h(x, \hat{c}, \hat{h}, \theta, W_o) - \hat{\Psi}_h$ , respectively, where  $\hat{\Psi}_c \triangleq \Psi_c(x, \hat{c}, \hat{h}, \hat{\theta})$  and  $\hat{\Psi}_h \triangleq \Psi_h(x, \hat{c}, \hat{h}, \hat{\theta}, \hat{W}_o)$ . Taking the derivative on both sides of (8) and (9) and substituting in the LSTM model in (5) and the auxiliary error dynamics in (10) and (11) yields

$$\dot{\tilde{c}} = -b_c \tilde{c} + b_c \tilde{\Psi}_c + g_e + \dot{\eta}_c, \quad (16)$$

$$\dot{\tilde{h}} = -b_h \tilde{h} + b_h \tilde{\Psi}_h + f_e + \dot{\eta}_h, \quad (17)$$

where the functions  $f_e(x, \tilde{c}, \tilde{h}, \theta, W_o) \in \mathbb{R}^{l_2}$  and  $g_e(x, \tilde{c}, \tilde{h}, \theta) \in \mathbb{R}^{l_2}$  are defined as  $f_e \triangleq b_h \Psi_h(x, c, h, \theta, W_o) - b_h \Psi_h(x, \hat{c}, \hat{h}, \theta, W_o)$  and  $g_e \triangleq b_c \Psi_c(x, c, h, \theta) - b_c \Psi_c(x, \hat{c}, \hat{h}, \theta)$ , respectively. Furthermore, let  $\tilde{\Phi} \triangleq \Phi(x, \hat{c}, \hat{h}, \theta, W_o, W_h) - \hat{\Phi}$ . Based on the subsequent stability analysis, the weight adaptation laws are designed as<sup>4</sup>

$$\begin{aligned} \text{vec}(\dot{\tilde{\theta}}) &\triangleq \text{proj}_\theta \left( \Gamma_\theta \left( b_c \tilde{\Psi}_c^\top \eta_c + b_h \tilde{\Psi}_{h,\theta}^\top \eta_h + \hat{\Phi}_\theta^\top r - \gamma_\theta \text{vec}(\tilde{\theta}) \right) \right), \\ \text{vec}(\dot{\hat{W}}_o) &\triangleq \text{proj}_{W_1} \left( \Gamma_o \left( b_h \tilde{\Psi}_{h,W_o}^\top \eta_h + \hat{\Phi}_{W_o}^\top r - \gamma_o \text{vec}(\hat{W}_o) \right) \right), \\ \text{vec}(\dot{\hat{W}}_h) &\triangleq \text{proj}_{W_2} \left( \Gamma_h \left( \hat{\Phi}_{W_h}^\top r - \gamma_h \text{vec}(\hat{W}_h) \right) \right), \\ \text{vec}(\dot{\hat{W}}_{FF}) &\triangleq \text{proj}_{W_3} \left( \Gamma_{FF} \hat{\Phi}_{W_{FF}}^\top r - \gamma_{FF} \text{vec}(\hat{W}_{FF}) \right), \end{aligned} \quad (18)$$

where  $\gamma_\theta, \gamma_o, \gamma_h, \gamma_{FF} \in \mathbb{R}_{>0}$  denote user-selected constants,  $\Gamma_\theta \in \mathbb{R}^{3l_1l_2 \times 3l_1l_2}$ ,  $\Gamma_o \in \mathbb{R}^{l_1l_2 \times l_1l_2}$ ,  $\Gamma_h \in \mathbb{R}^{l_2n \times l_2n}$ , and  $\Gamma_{FF} \in \mathbb{R}^{5l_2n \times 5l_2n}$  denote user-selected positive-definite gain matrices, the short-hand notations  $\tilde{\Psi}_c$ ,  $\tilde{\Psi}_{h,\theta}$ ,  $\tilde{\Psi}_{h,W_o}$ ,  $\hat{\Phi}_\theta$ ,  $\hat{\Phi}_{W_o}$ ,  $\hat{\Phi}_{W_h}$ , and  $\hat{\Phi}_{W_{FF}}$  denote the Jacobians  $\tilde{\Psi}_c' \triangleq \frac{\partial \tilde{\Psi}_c}{\partial \text{vec}(\tilde{\theta})}$ ,  $\tilde{\Psi}_{h,\theta}' \triangleq \frac{\partial \tilde{\Psi}_h}{\partial \text{vec}(\tilde{\theta})}$ ,  $\tilde{\Psi}_{h,W_o}' \triangleq \frac{\partial \tilde{\Psi}_{h,W_o}}{\partial \text{vec}(\tilde{W}_o)}$ ,  $\hat{\Phi}_\theta' \triangleq \frac{\partial \hat{\Phi}}{\partial \text{vec}(\tilde{\theta})}$ ,  $\hat{\Phi}_{W_o}' \triangleq \frac{\partial \hat{\Phi}_{W_o}}{\partial \text{vec}(\tilde{W}_o)}$ ,  $\hat{\Phi}_{W_h}' \triangleq \frac{\partial \hat{\Phi}_{W_h}}{\partial \text{vec}(\tilde{W}_h)}$ , and  $\hat{\Phi}_{W_{FF}}' \triangleq \frac{\partial \hat{\Phi}_{W_{FF}}}{\partial \text{vec}(\tilde{W}_{FF})}$ , respectively, and  $\text{proj}(\cdot)$  denotes the projection operator defined in [25, Appendix E, Eq. E.4]. The projection operators  $\text{proj}_\theta(\cdot)$ ,  $\text{proj}_{W_1}(\cdot)$ ,  $\text{proj}_{W_2}(\cdot)$ , and  $\text{proj}_{W_3}(\cdot)$  in (18) are used to ensure  $\tilde{\theta}(t) \in \mathcal{B}_\theta \triangleq \{\zeta \in \mathbb{R}^{3l_1l_2} : \|\zeta\| \leq \sqrt{3\bar{W}}\}$ ,  $\hat{W}_o(t) \in \mathcal{B}_{W_1} \triangleq \{\zeta \in \mathbb{R}^{l_1l_2} : \|\zeta\| \leq \bar{W}\}$ ,  $\hat{W}_h(t) \in \mathcal{B}_{W_2} \triangleq \{\zeta \in \mathbb{R}^{l_2n} : \|\zeta\| \leq \bar{W}\}$ , and  $\hat{W}_{FF}(t) \in \mathcal{B}_{W_3} \triangleq \{\zeta \in \mathbb{R}^{5nl_2} : \|\zeta\| \leq \bar{W}\}$ , respectively.

The Jacobians  $\tilde{\Psi}_c'$ ,  $\tilde{\Psi}_{h,\theta}'$ , and  $\hat{\Phi}_\theta'$  can be represented as  $\tilde{\Psi}_c' \triangleq [\tilde{\Psi}_{c,W_c}', \tilde{\Psi}_{c,W_i}', \tilde{\Psi}_{c,W_f}']$ ,  $\tilde{\Psi}_{h,\theta}' \triangleq [\tilde{\Psi}_{h,W_c}', \tilde{\Psi}_{h,W_i}', \tilde{\Psi}_{h,W_f}']$ , and  $\hat{\Phi}_\theta' \triangleq [\hat{\Phi}_{W_c}', \hat{\Phi}_{W_i}', \hat{\Phi}_{W_f}']$ , respectively, where  $\tilde{\Psi}_{c,W_j}' \triangleq \frac{\partial \tilde{\Psi}_c}{\partial \text{vec}(\tilde{W}_j)}$ ,  $\tilde{\Psi}_{h,W_j}' \triangleq \frac{\partial \tilde{\Psi}_h}{\partial \text{vec}(\tilde{W}_j)}$ , and  $\hat{\Phi}_{W_j}' \triangleq \frac{\partial \hat{\Phi}}{\partial \text{vec}(\tilde{W}_j)}$  for all  $j \in \{c, i, f\}$ . Using (5), (14), (15), the chain rule, the properties of the Hadamard product, and the properties of vectorization,

<sup>4</sup>The terms  $\eta_c$  and  $\eta_h$  are introduced and implemented in the auxiliary cell and hidden state estimation errors  $\tilde{c}$  and  $\tilde{h}$  to allow the weight adaptation laws in (18) to adaptively compensate for the uncertainty in the internal dynamics inherent in the LSTM cell through the terms  $b_c \tilde{\Psi}_c^\top \eta_c$ ,  $b_h \tilde{\Psi}_{h,\theta}^\top \eta_h$ , and  $b_h \tilde{\Psi}_{h,W_o}^\top \eta_h$ .

the terms  $\tilde{\Psi}_{c,W_c}'$ ,  $\tilde{\Psi}_{c,W_i}'$ , and  $\tilde{\Psi}_{c,W_f}'$  can be expressed as

$$\begin{aligned} \tilde{\Psi}_{c,W_c}' &= \text{diag}(\sigma_g(\hat{W}_c^\top \tilde{z})) \sigma'_c(W_c^\top \tilde{z})(I_{l_2} \otimes \tilde{z}^\top), \\ \tilde{\Psi}_{c,W_i}' &= \text{diag}(\sigma_c(\hat{W}_c^\top \tilde{z})) \sigma'_c(W_i^\top \tilde{z})(I_{l_2} \otimes \tilde{z}^\top), \\ \tilde{\Psi}_{c,W_f}' &= \text{diag}(\hat{c}) \sigma'_g(\hat{W}_f^\top \tilde{z})(I_{l_2} \otimes \tilde{z}^\top), \end{aligned} \quad (19)$$

respectively. Similarly, using (19), the terms  $\tilde{\Psi}_{h,W_j}'$  and  $\tilde{\Psi}_{h,W_o}'$  can be expressed as

$$\begin{aligned} \tilde{\Psi}_{h,W_j}' &= \text{diag}(\sigma_g(\hat{W}_o^\top \tilde{z})) \sigma'_c(\hat{\Psi}_c) \tilde{\Psi}_{c,W_j}', \\ \tilde{\Psi}_{h,W_o}' &= \text{diag}(\sigma_c(\hat{\Psi}_c)) (\sigma'_g(\hat{W}_o^\top \tilde{z})) (I_{l_2} \otimes \tilde{z}^\top), \end{aligned} \quad (20)$$

for all  $j \in \{c, i, f\}$ , respectively. Using (6) and the chain rule, the Jacobians  $\tilde{\Phi}_{W_j}'$ ,  $\tilde{\Phi}_{W_o}'$ ,  $\tilde{\Phi}_{W_h}'$ , and  $\tilde{\Phi}_{W_{FF}}'$  can be expressed as  $\tilde{\Phi}_{W_j}' = \hat{W}_h^\top \tilde{\Psi}_{h,W_j}'$ ,  $\tilde{\Phi}_{W_o}' = \hat{W}_h^\top \tilde{\Psi}_{h,W_o}'$ ,  $\tilde{\Phi}_{W_h}' = I_{l_2} \otimes \tilde{\Psi}_h^\top$ , and  $\tilde{\Phi}_{W_{FF}}' = \hat{W}_h^\top \sigma'(\hat{W}_{FF} x)(I_{l_2} \otimes x^\top)$ , for all  $j \in \{c, i, f\}$ , respectively. NNs such as the LSTM model in (5) are nonlinear in terms of the weights. Moreover, the LSTM model has added complexity due to the three gate units present in the cell architecture. To address the resulting mathematical challenges, a first-order Taylor Series approximation-based error model of the LSTM in (5) and (6) is given by

$$\begin{aligned} \tilde{\Psi}_c &= \tilde{\Psi}_c' \text{vec}(\tilde{\theta}) + \mathcal{O}_c^2(\tilde{\theta}), \\ \tilde{\Psi}_h &= \tilde{\Psi}_{h,W_o}' \text{vec}(\tilde{W}_o) + \tilde{\Psi}_{h,\theta}' \text{vec}(\tilde{\theta}) + \mathcal{O}_h^2(\tilde{\theta}, \tilde{W}_o), \\ \tilde{\Phi} &= \tilde{\Phi}_{W_h}' \text{vec}(\tilde{W}_h) + \tilde{\Phi}_{W_{FF}}' \text{vec}(\tilde{W}_{FF}) + \tilde{\Phi}_{W_o}' \text{vec}(\tilde{W}_o) \\ &\quad + \tilde{\Phi}_\theta' \text{vec}(\tilde{\theta}) + \mathcal{O}_\Phi^2(\tilde{\theta}, \tilde{W}_o, \tilde{W}_h, \tilde{W}_{FF}), \end{aligned} \quad (21)$$

where  $\mathcal{O}_c^2(\tilde{\theta}) \in \mathbb{R}^{l_2}$ ,  $\mathcal{O}_h^2(\tilde{\theta}, \tilde{W}_o) \in \mathbb{R}^{l_2}$ , and  $\mathcal{O}_\Phi^2(\tilde{\theta}, \tilde{W}_o, \tilde{W}_h, \tilde{W}_{FF}) \in \mathbb{R}^n$  denotes the higher-order terms. Using Lemma 1, the higher-order terms can be bounded as  $\|\mathcal{O}_c^2(\tilde{\theta})\|, \|\mathcal{O}_h^2(\tilde{\theta}, \tilde{W}_o)\|, \|\mathcal{O}_\Phi^2(\tilde{\theta}, \tilde{W}_o, \tilde{W}_h, \tilde{W}_{FF})\| \leq \bar{O}$ , where  $\bar{O} \in \mathbb{R}_{>0}$  denotes a known constant.

#### IV. STABILITY ANALYSIS

To facilitate the subsequent stability analysis, let the concatenated state vector  $\zeta : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^\psi$  and constant  $\kappa \in \mathbb{R}_{>0}$  be defined as  $\zeta \triangleq [e^\top, r^\top, \eta_c^\top, \tilde{c}^\top, \eta_h^\top, \tilde{h}^\top, \text{vec}(\tilde{\theta})^\top, \text{vec}(\tilde{W}_h)^\top, \text{vec}(\tilde{W}_o)^\top, \text{vec}(\tilde{W}_{FF})^\top]^\top$  and  $\kappa \triangleq \min\{\frac{b_c}{2} - \frac{k_{1,c}}{2} - \frac{\|K_{2,c}\|_F}{2}, \frac{b_h}{2} - \frac{k_{1,h}}{2} - \frac{\|K_{2,h}\|_F}{2}, \frac{k_r}{2} - \frac{\|K_{2,c}\|_F}{2} - \frac{\|K_{2,h}\|_F}{2}, \frac{k_{1,c}}{4}, \frac{k_{1,h}}{4}, \gamma_\theta, \gamma_h, \gamma_{FF}, \gamma_o, \alpha\}$ , respectively, where  $\psi \triangleq 2n + 4l_2 + 4l_1l_2 + 6nl_2$ . Additionally, let the auxiliary function  $\tilde{N} : \mathbb{R}^\psi \rightarrow \mathbb{R}$  be defined as  $\tilde{N} \triangleq r^\top j_e + \tilde{c}^\top g_e + \tilde{h}^\top f_e + b_c \tilde{\Psi}_c^\top \text{vec}(\tilde{\theta})(c - \hat{c}) + (b_h \tilde{\Psi}_{h,W_o}' \text{vec}(\tilde{W}_o) + b_h \tilde{\Psi}_{h,\theta}' \text{vec}(\tilde{\theta}))^\top (h - \hat{h})$ , where  $\tilde{N}$  represents a group of terms that appear in the subsequent stability analysis. Applying the mean value theorem-based inequality [26, Appendix A] on the terms  $r^\top j_e$ ,  $\tilde{c}^\top g_e$ , and  $\tilde{h}^\top f_e$ , and bounding  $\|r\|$ ,  $\|\tilde{c}\|$ ,  $\|\tilde{h}\|$ , and  $\|z\|$  terms with  $\|\zeta\|$ , the auxiliary function  $\tilde{N}$  can be bounded as  $\|\tilde{N}\| \leq \rho(\|\zeta\|) \|\zeta\|^2$ , where  $\rho(\cdot)$  denotes an invertible, strictly non-increasing function. Let the open and connected sets  $\mathcal{D} \subset \mathbb{R}^\psi$  and  $\Upsilon \subseteq \mathcal{Z}$  be defined as  $\mathcal{D} \triangleq \{\zeta \in \mathbb{R}^\psi : \|\zeta\| < \sqrt{\frac{\beta_1}{\beta_2}} \rho^{-1}(\kappa - \lambda)\}$

and  $\Upsilon = \{\zeta \in \mathcal{Z} : \|\zeta\| < \bar{z}\}$ , respectively, where  $\lambda \in \mathbb{R}_{>0}$  denotes a user-selected constant,  $\delta \triangleq \frac{(\bar{O}^2 + \bar{\varepsilon})^2}{2k_r} + \frac{(2b_c \bar{O}^2)^2}{k_{1,c}} + \frac{(2b_h \bar{O}^2)^2}{k_{1,h}} + \frac{(b_c \bar{O}^2)^2}{2b_c} + \frac{(b_h \bar{O}^2)^2}{2b_h} + 6\gamma_\theta \bar{W} + 2(\gamma_h + \gamma_o + \gamma_{FF}) \bar{W}^2$ , and  $\bar{z} \triangleq (2 + \alpha)\omega + 2\bar{q}_d + 2\bar{q}_d + \bar{q}_d + \frac{\sqrt{I_2}}{\sqrt{2(b_h - \frac{1}{2})}} + 1$ . The

developed adaptive LSTM-based architecture in (12) and (18) is shown to be uniformly ultimately bounded (UUB) in the following theorem.

*Theorem 1:* Consider the model dynamics in (1) with Property 1. The Lb-LSTM controller in (12) and the weight adaptation laws in (18) ensure the states  $\zeta$  are UUB in the

sense that  $\|\zeta\| \leq \sqrt{\frac{\beta_2}{\beta_1}} \|\zeta(t_0)\|^2 e^{-\frac{\lambda}{\beta_1}(t-t_0)} + \frac{\delta}{\lambda} \left(1 - e^{-\frac{\lambda}{\beta_1}(t-t_0)}\right)$  provided the sufficient gain conditions  $\kappa \geq \lambda + \rho \left( \sqrt{\frac{\beta_2}{\beta_1}} (\|y(t_0)\| + 2\sqrt{I_2} + 6\bar{W} + 6\bar{W}^2) \right)$ ,  $b_h \geq \frac{1}{2}$ ,  $b_c \geq \frac{1}{2(1+\sqrt{I_2})}$  are satisfied, where  $y \triangleq [e^\top, r^\top, \eta_c^\top, \eta_h^\top]^\top$ ,  $\beta_1 \triangleq \lambda_{\min}\{1, \Gamma_\theta^{-1}, \Gamma_o^{-1}, \Gamma_{FF}^{-1}, \Gamma_h^{-1}\}$  and  $\beta_2 \triangleq \lambda_{\max}\{1, \Gamma_\theta^{-1}, \Gamma_o^{-1}, \Gamma_{FF}^{-1}, \Gamma_h^{-1}\}$ .

*Proof:* Consider the Lyapunov candidate function  $\mathcal{V}_L : \mathbb{R}^\psi \rightarrow \mathbb{R}_{\geq 0}$

$$\begin{aligned} \mathcal{V}_L(\zeta) &\triangleq \frac{1}{2} \eta_c^\top \eta_c + \frac{1}{2} \eta_h^\top \eta_h + \frac{1}{2} \tilde{c}^\top \tilde{c} + \frac{1}{2} \tilde{h}^\top \tilde{h} + \frac{1}{2} e^\top e + \frac{1}{2} r^\top M r \\ &\quad + \frac{1}{2} \text{vec}(\tilde{W}_h)^\top \Gamma_h^{-1} \text{vec}(\tilde{W}_h) + \frac{1}{2} \text{vec}(\tilde{W}_o)^\top \Gamma_o^{-1} \text{vec}(\tilde{W}_o) \\ &\quad + \frac{1}{2} \text{vec}(\tilde{\theta})^\top \Gamma_\theta^{-1} \text{vec}(\tilde{\theta}) + \frac{1}{2} \text{vec}(\tilde{W}_{FF})^\top \Gamma_h^{-1} \text{vec}(\tilde{W}_{FF}), \end{aligned} \quad (22)$$

which can be bounded as  $\beta_1 \|\zeta\|^2 \leq \mathcal{V}_L(\zeta) \leq \beta_2 \|\zeta\|^2$ . Substituting (2), (3), (13), (16), and (17) into the time derivative of  $\mathcal{V}_L$  and canceling cross-terms yields

$$\begin{aligned} \dot{\mathcal{V}}_L &= -\alpha e^\top e - k_r r^\top r - k_{1,c} \eta_c^\top \eta_c - k_{1,h} \eta_h^\top \eta_h + \tilde{h}^\top (f_e + \dot{\eta}_h) \\ &\quad - \tilde{h}^\top (b_h \tilde{h} - b_h \tilde{\Psi}_h) - \tilde{c}^\top (b_c \tilde{c} - b_c \tilde{\Psi}_c - g_e - \dot{\eta}_c) \\ &\quad + r^\top (\tilde{\Phi} + j_e + \varepsilon(x_d)) - \text{vec}(\tilde{W}_o)^\top \Gamma_o^{-1} \text{vec}(\hat{W}_o) \\ &\quad - \text{vec}(\tilde{W}_h)^\top \Gamma_h^{-1} \text{vec}(\hat{W}_h) - \text{vec}(\tilde{\theta})^\top \Gamma_\theta^{-1} \text{vec}(\dot{\tilde{\theta}}) \\ &\quad - \text{vec}(\tilde{W}_{FF})^\top \Gamma_{FF}^{-1} \text{vec}(\hat{W}_{FF}). \end{aligned} \quad (23)$$

Using [25, Lemma E.1.IV],  $-\tilde{V}^\top \Gamma^{-1} \text{proj}(\kappa) \leq -\tilde{V}^\top \Gamma^{-1} \kappa$ , where the estimation error  $\tilde{V} \in \mathbb{R}^m$  is defined as  $\tilde{V} \triangleq V - \hat{V}$  for some  $V, \hat{V} \in \mathbb{R}^m$  such that  $\|V\| \leq \bar{V}$  and  $\text{proj}(\cdot)$  ensures  $\hat{V}(t) \in \mathcal{B}_V \triangleq \{\zeta \in \mathbb{R}^m : \|\zeta\| \leq \bar{V}\}$ , where  $\bar{V} \in \mathbb{R}_{>0}$  denotes a known constant. Therefore, substituting in (10) and (11), the weight adaptation laws in (18), the first order Taylor series approximation in (21), and the definition of  $\tilde{N}$ , and using the facts that  $\tilde{\Psi}_c^\top \tilde{c} = \tilde{\Psi}_c^\top (c - \hat{c} + \eta_c)$  and  $\tilde{\Psi}_h^\top \tilde{h} = \tilde{\Psi}_h^\top (h - \hat{h} + \eta_h)$ , yields

$$\begin{aligned} \dot{\mathcal{V}}_L &\leq -\alpha e^\top e - k_r r^\top r - b_c \tilde{c}^\top \tilde{c} - b_h \tilde{h}^\top \tilde{h} - k_{1,c} \eta_c^\top \eta_c - k_{1,h} \eta_h^\top \eta_h \\ &\quad + r^\top (\mathcal{O}^2(\tilde{\theta}, \tilde{W}_o, \tilde{W}_h, \tilde{W}_{FF}) + \varepsilon(x)) + b_h \eta_h \mathcal{O}^2(\tilde{\theta}, \tilde{W}_o) \\ &\quad - \tilde{c}^\top (k_{1,c} \eta_c + K_{2,c} r) - \tilde{h}^\top (k_{1,h} \eta_h + K_{2,h} r) + b_c \eta_c \mathcal{O}^2(\tilde{\theta}) \\ &\quad + b_c \mathcal{O}^2(\tilde{\theta})^\top (c - \hat{c}) + b_h \mathcal{O}^2(\tilde{\theta}, \tilde{W}_o)^\top (h - \hat{h}) + \tilde{N} \end{aligned}$$

$$\begin{aligned} &- \text{vec}(\tilde{\theta})^\top (-\gamma_\theta \text{vec}(\hat{\theta})) - \text{vec}(\tilde{W}_o)^\top (-\gamma_o \text{vec}(\hat{W}_o)) \\ &+ \gamma_h \text{vec}(\tilde{W}_h)^\top \text{vec}(\hat{W}_h) + \gamma_{FF} \text{vec}(\tilde{W}_{FF})^\top \text{vec}(\hat{W}_{FF}). \end{aligned} \quad (24)$$

Using Young's inequality and the facts that  $\|\tilde{N}\| \leq \rho(\|\zeta\|) \|\zeta\|^2$ ,  $\hat{\theta} = \theta - \tilde{\theta}$ ,  $\tilde{W}_o = W_o - \tilde{W}_o$ ,  $\tilde{W}_{FF} = W_{FF} - \tilde{W}_{FF}$ , and  $\tilde{W}_h = W_h - \tilde{W}_h$ , (24) can be bounded as  $\dot{\mathcal{V}}_L \leq -(\kappa - \rho(\|\zeta\|)) \|\zeta\|^2 + \delta$ . From (22),  $\|\zeta\| \leq \sqrt{\frac{\mathcal{V}_L}{\beta_1}}$ , and therefore  $\dot{\mathcal{V}}_L$  can be bounded as  $\dot{\mathcal{V}}_L \leq -(\kappa - \rho(\sqrt{\frac{\mathcal{V}_L}{\beta_1}})) \frac{\mathcal{V}_L}{\beta_1} + \delta$ . Selecting  $\kappa$  according to Theorem 1 ensures  $\|\zeta(t_0)\|$  is bounded as  $\|\zeta(t_0)\| < \sqrt{\frac{\beta_1}{\beta_2}} \rho^{-1}(\kappa - \lambda)$ . Thus, when all trajectories are initialized in  $\mathcal{D}$ ,  $\dot{\mathcal{V}}_L$  can be further bounded as  $\dot{\mathcal{V}}_L \leq -\frac{\lambda}{\beta_1} \mathcal{V}_L + \delta$ , which implies  $\mathcal{V}_L(t) \leq \mathcal{V}_L(t_0) e^{-\frac{\lambda}{\beta_1}(t-t_0)} + \frac{\delta \beta_1}{\lambda} \left(1 - e^{-\frac{\lambda}{\beta_1}(t-t_0)}\right)$ . Then, [27, Definition 4.6] can be invoked to conclude that  $\zeta$  is UUB such that  $\|\zeta\| \leq \mu \triangleq \sqrt{\frac{\beta_2}{\beta_1}} \|\zeta(t_0)\|^2 e^{-\frac{\lambda}{\beta_1}(t-t_0)} + \frac{\delta}{\lambda} \left(1 - e^{-\frac{\lambda}{\beta_1}(t-t_0)}\right)$ . To show  $z \in \mathcal{Z}$ , and therefore the universal function approximation property holds, let  $\xi \triangleq [e^\top, r^\top]^\top$  and let  $\omega = \rho^{-1}(\kappa - \lambda)$ . Thus, if  $\|\zeta(t_0)\| \leq \omega \sqrt{\frac{\beta_1}{\beta_2}}$ , then  $\|\xi(t)\| \leq \omega$ , and therefore  $\|e(t)\| \leq \omega$  and  $\|r(t)\| \leq \omega$ . Hence, using (2), (3), and Lemma 1,  $\|z\|$  can be bounded as  $\|z\| \leq (2 + \alpha)\omega + 2\bar{q}_d + 2\bar{q}_d + \bar{q}_d + \frac{\sqrt{I_2}}{\sqrt{2(b_h - \frac{1}{2})}} + 1$  provided the sufficient gain conditions  $b_h \geq \frac{1}{2}$ ,  $b_c \geq \frac{1}{2(1+\sqrt{I_2})}$  are met for Lemma 1 to hold. Therefore, if  $\zeta(t_0) \in \mathcal{D}$ , then  $z \in \Upsilon \subseteq \mathcal{Z}$ . Since  $\zeta \in L_\infty$ ,  $q, \dot{q} \in \mathcal{L}_\infty$ . That and the fact that  $\hat{c}, \hat{h}, \hat{\theta}, \tilde{W}_o, \tilde{W}_h, \tilde{W}_{FF} \in \mathcal{L}_\infty$  by design imply  $\tau \in \mathcal{L}_\infty$ . ■

## V. SIMULATIONS

To demonstrate the performance and efficacy of the developed Lb-LSTM control design, simulations were performed on the two-link robot manipulator model in [26, Eq. (80)]. To demonstrate the advantages of using the Lb-LSTM architecture instead of a feedforward DNN architecture in the adaptive controller, the results are compared with the DNN-based adaptive controller developed in [2] as the baseline. The baseline adaptive DNN-based controller in [2] is  $\tau \triangleq \hat{\Phi}_{DNN} + k_r r + e$ , where the DNN estimate  $\hat{\Phi}_{DNN}$  was updated according to the weight adaptation laws defined in [2, Eqs. (7)–(8)]. The LSTM model in (5) was used with tanh activation functions for the feedforward term and  $l_2 = 12$  neurons and was compared to 3 baseline fully-connected DNN architectures, DNN1, DNN2, and DNN3, with 1, 2, and 5 hidden layers each, respectively, with tanh activation functions. DNN1 and DNN2 had 12 neurons in each layer and DNN3 had 14 neurons. The weights of all NNs were randomly initialized with a uniform distribution with values ranging between -1 and 1. The gains were selected as  $\alpha = 15$ ,  $k_r = 50$ ,  $k_{1,c} = 5$ ,  $k_{1,h} = 5$ ,  $K_{2,c} = 0.1 \cdot [I_2 \ 0_{2 \times 10}]^\top$ ,  $K_{2,h} = 0.1 \cdot [I_2 \ 0_{2 \times 10}]^\top$ ,  $b_c = 5$ ,  $b_h = 1$ ,  $\Gamma_\theta = 40 \cdot I_{3l_1 l_2}$ ,  $\Gamma_o = 40 \cdot I_{l_1 l_2}$ ,  $\Gamma_h = 40 \cdot I_{l_2 n}$ ,  $\Gamma_{FF} = 40 \cdot I_{5l_2 n}$ , and  $\gamma_\theta = \gamma_o = \gamma_h = \gamma_{FF} = 0.01$  for the adaptive LSTM controller. For the baseline controllers, the gains were selected as  $\alpha = 15$ ,  $k_r = 50$ , and  $\Gamma_j = 24 \cdot I_{L_j L_{j+1}}$   $\forall j \in \{0, \dots, k\}$ . For a fair comparison, the same robust control gains were used for

**TABLE I**  
PERFORMANCE COMPARISON RESULTS

NN Architecture	$\ e\ [\text{deg}]$	$\ g(x) - \hat{\Phi}\ $	$\ \tau\ [\text{N}\cdot\text{m}]$
DNN1	0.6374	36.4609	33.3563
DNN2	0.6360	22.2069	19.5852
DNN3	0.5302	11.8436	9.2579
LSTM	0.3970	3.7480	6.1471

each controllers. The NN gains and parameters (e.g., the learning gains and activation functions) were empirically adjusted to achieve the best performance for each network. For all simulations, the desired trajectory  $q_d(t) \triangleq [q_{d,1}, q_{d,2}]^\top \in \mathbb{R}^2$  was selected as  $q_d \triangleq \begin{bmatrix} \frac{\pi}{3} \sin(\frac{\pi}{4}t) \\ \frac{\pi}{2} \sin(\frac{\pi}{2}t) \end{bmatrix} \in \mathbb{R}^2$  [rad], and the simulations were performed for 25 s with the initial conditions  $q(0) = [1.0472, -0.5236]^\top$  [rad] and  $\dot{q}(0) = [0, 0]^\top$  [rad/s].

Although all four adaptive NN architectures compensated for the uncertainty in the dynamics and achieved tracking, the LSTM provided improved tracking performance with a significant improvement in both function approximation performance and control effort, when compared to the feedforward NN architectures (see Table I). Moreover, the LSTM provided twofold and fourfold faster tracking and function approximation error convergence, respectively, compared to DNN3 with better transient behavior. When compared to the adaptive controller DNN3, the LSTM-based controller and developed weight adaptation law resulted in 25.1343% and 68.3541% improvement in the tracking error and function approximation error, respectively, while requiring 33.6015% reduced control effort, as shown in Table I.

## VI. CONCLUSION

An adaptive LSTM-based controller was developed for general uncertain Euler-Lagrange systems. Leveraging the dynamic structure and internal memory inherent in LSTMs, the developed Lb-LSTM architecture is able to leverage time dependencies in the system dynamics and capture time-varying accumulative effects in the system dynamics that static, feed-forward NNs cannot. Unlike traditional RNNs, the cell structure of LSTMs allows the LSTM to retain relevant information across longer time sequences. Stability-driven weight adaptation laws are developed for the Lb-LSTM weights in real-time, eliminating the need for offline pre-training. A Lyapunov-based stability analysis is performed and guarantees UUB for the tracking errors, LSTM estimation errors, and weight estimation errors. To validate the developed adaptive LSTM-based controller, simulations were performed to compare the developed method to the adaptive DNN-based controller in [2] and yielded twofold and fourfold faster tracking error and function approximation error convergence, respectively, when compared to a baseline DNN architecture of a similar size.

## 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

- 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.
- 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.
- 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.
- E. Inanc, Y. Gurses, A. Habboush, Y. Yildiz, and A. M. Annaswamy, “Neural network adaptive control with long short-term memory,” 2023, *arXiv:2301.02316*.
- A. Graves, G. Wayne, and I. Danihelka, “Neural turing machines,” 2014, *arXiv:1410.5401*.
- 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.
- E. Parisotto and R. Salakhutdinov, “Neural map: Structured memory for deep reinforcement learning,” in *Proc. Int. Conf. Learn. Represent.*, 2018.
- 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.
- K. Funahashi and Y. Nakamura, “Approximation of dynamical systems by continuous time recurrent neural networks,” *Neural Netw.*, vol. 6, no. 6, pp. 801–806, 1993.
- F.-J. Lin, H.-J. Shieh, P.-H. Shieh, and P.-H. Shen, “An adaptive recurrent-neural-network motion controller for X-Y table in CNC machine,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 2, pp. 286–299, Apr. 2006.
- S.-B. Chen et al., “Recurrent neural network-based robust nonsingular sliding mode control with input saturation for a non-holonomic spherical robot,” *IEEE Access*, vol. 8, pp. 188441–188453, 2020.
- A. El-Nagar, A. Zaki, F. Soliman, and M. El-Bardini, “Hybrid deep learning diagonal recurrent neural network controller for nonlinear systems,” *Neural. Comput. Appl.*, vol. 34, pp. 22367–22386, Aug. 2022.
- R.-J. Wai and F.-J. Lin, “Adaptive recurrent-neural-network control for linear induction motor,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 4, pp. 1176–1192, Oct. 2001.
- Y. Chu, J. Fei, and S. Hou, “Adaptive global sliding-mode control for dynamic systems using double hidden layer recurrent neural network structure,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1297–1309, Apr. 2020.
- 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, pp. 825–836, Apr. 2023.
- 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. Problem Eng.*, vol. 2021, Mar. 2021, Art. no. 8882312.
- E. Griffis, O. Patil, W. Makumi, and W. E. Dixon, “Deep recurrent neural network-based observer for uncertain nonlinear systems,” in *Proc. IFAC World Congr.*, 2023.
- J. S. Alex Graves, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, 2005.
- V. Carbune et al., “Fast multi-language LSTM-based Online handwriting recognition,” *Int. J. Doc. Anal. Recognit.*, vol. 23, no. 2, pp. 89–102, 2020.
- Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, pp. 1735–80, Dec. 1997.
- D. S. Bernstein, *Matrix Mathematics*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
- 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.
- P. Kidger and T. Lyons, “Universal approximation with deep narrow networks,” in *Proc. Conf. Learn. Theory*, 2020, pp. 2306–2327.
- M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*. New York, NY, USA: Wiley, 1995.
- M. de Queiroz, J. Hu, D. Dawson, T. Burg, and S. Donepudi, “Adaptive position/force control of robot manipulators without velocity measurements: Theory and experimentation,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 27-B, no. 5, pp. 796–809, Oct. 1997.
- H. K. Khalil, *Nonlinear Systems*, 3rd ed. Hoboken, NJ, USA: Prentice Hall, 2002.