

Lyapunov-Based Physics-Informed Long Short-Term Memory (LSTM) Neural Network-Based Adaptive Control

Rebecca G. Hart¹, Graduate Student Member, IEEE, Emily J. Griffis², Graduate Student Member, IEEE, Omkar Sudhir Patil³, and Warren E. Dixon⁴, Fellow, IEEE

Abstract—Deep neural networks (DNNs) and long short-term memory networks (LSTMs) have grown in recent popularity due to their function approximation performance when compared to traditional NN architectures. However, the predictions that may result from these networks often do not align with physical principles. This letter introduces the first physics-informed LSTM (PI-LSTM) controller composed of DNNs and LSTMs, where the weight adaptation laws are designed from a Lyapunov-based analysis. The developed PI-LSTM combines DNNs and LSTMs for the purpose of function approximation and memory while respecting the underlying system physics. Simulations were performed to demonstrate feasibility and resulted in a root mean square tracking error of 0.0185 rad and a 33.76% improvement over the baseline method.

Index Terms—Long short-term memory, physics-informed learning, nonlinear control systems.

I. INTRODUCTION

IN RECENT years, with the improvement of computing power, there has been a surge in deep learning and its applicability in a wide range of control applications. Deep neural network (DNN)-based control has become increasingly popular due to the improved function approximation capabilities of DNNs when compared to shallow neural networks (NNs) [1], [2], [3], [4]. In particular, DNNs have been shown to be exponentially more efficient regarding the number of neurons required to achieve similar function approximation performance [5].

Manuscript received 15 September 2023; revised 22 November 2023; accepted 11 December 2023. Date of publication 26 December 2023; date of current version 22 January 2024. This work was supported in part by the National Science Foundation Graduate Research Fellowship under Grant DGE-1842473; in part by the Air Force Research Lab (AFRL) under Grant FA8651-21-F-1027; in part by the Office of Naval Research under Grant N00014-21-1-2481; and in part by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-19-1-0169. Recommended by Senior Editor A. P. Aguiar. (Corresponding author: Rebecca G. Hart.)

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

Digital Object Identifier 10.1109/LCSYS.2023.3347485

In general, NN architectures are popular due to their ability to approximate functions over a compact space for applications which have no model knowledge and rely solely on output data. However, in some applications, partial knowledge of the dynamics may be available. As a result, physics-informed learning emerged as a strategy which aims to incorporate established physical principles into the learning architectures [6], [7], [8]. This strategy allows physics-informed neural networks (PINNs) to make more realistic and more accurate predictions when compared to their traditional black-box NN counterparts. Results such as [6], [9], [10], [11] develop PINNs suitable for modeling and control of dynamical systems. However, these works implement offline training methods to update the PINN architectures and do not adapt the weights of the PINN online using real-time data. Offline training requires state-derivative information and representative training sets. While such methods could be used as an initial condition (but are not required if training data is not available), real-time weight adaptation can learn in dynamic environments and offers stability guarantees while incorporating streams of new data.

Results in [2], [12], [13], [14] develop adaptive control techniques using Lyapunov-based weight adaptation laws which allow for online learning with DNNs. Recently, Lyapunov-based techniques were also used to develop an adaptive PINN architecture using the known structure of Euler-Lagrange model dynamics [14]. The PINN architecture in [14] is composed of multiple feedforward DNNs which estimate the unknown inertia, centripetal-Coriolis, potential, and dissipation effects. However, feedforward DNNs are unable to capture temporal relationships and history-dependent behaviors. Many systems involve complex physics which can be characterized using history-dependent inertia, centripetal-Coriolis, potential, and dissipation effects. Some examples include systems composed of smart materials, or experience fluid-structure interaction, human-machine interaction, electromagnetic effects, and etc. Thus, there is a need to augment the PINN architecture to account for these phenomena.

Motivated by the desire to capture time-dependent relationships, long short-term memory (LSTM) NNs have been developed and have gained popularity due to their ability to

retain relevant information across multiple time steps [15]. Specifically, the LSTM cell includes an explicit memory which can be used to model internal dynamics and complex temporal behaviors. When integrated with control systems, LSTMs have been shown to improve function approximation and reduce control effort [12], [16], [17]. The result in [12] develops Lyapunov-based stability-driven weight adaptation laws for all weights of the LSTM cell; however, this result does not incorporate a physics-informed approach.

This letter provides the first result on an adaptive PI-LSTM-based control architecture which merges the function approximation capabilities of DNNs and memory retention attributes of LSTMs to model and compensate for unknown system dynamics while using physical insight. Unlike the black-box DNN architectures in previous results [2], [12], the developed PI-LSTM controller incorporates physical insight into the architecture. The PI-LSTM weights are updated in real-time using adaptation laws derived from a Lyapunov-based stability analysis. The Lyapunov-based PI-LSTM demonstrates a significant advancement in physics-informed learning for control which offers accurate modeling combined with real-time adaptability while maintaining physical plausibility. A Lyapunov-based stability analysis is performed to ensure asymptotic tracking error convergence and boundedness of the developed PI-LSTM controller. Simulations were performed on a two-link robot manipulator and yielded an root mean square (RMS) tracking error of 0.0185 rad and achieved a 33.76% improvement over the baseline method presented in [14].

II. PROBLEM FORMULATION

Notation and Preliminaries: Given $A \triangleq [a_{j,i}] \in \mathbb{R}^{n \times m}$, $\text{vec}(A) \triangleq [a_{1,1}, \dots, a_{1,m}, \dots, a_{n,m}]^\top$. Given any $A \in \mathbb{R}^{n \times m}$, $B \in \mathbb{R}^{m \times p}$, and $C \in \mathbb{R}^{p \times r}$, $\text{vec}(ABC) = (C^\top \otimes A)\text{vec}(B)$ a.a.t.

The notation (\cdot) denotes the relation (\cdot) holds for almost all time (a.a.t.). The notation $K[\cdot]$ denotes Filippov's differential inclusion applied on $[\cdot]$ [18]. The right-to-left matrix product

operator is represented by $\overset{\leftarrow}{\prod}$, i.e., $\overset{\leftarrow}{\prod}_{p=1}^m A_p = A_m \dots A_2 A_1$ and

$\overset{\leftarrow}{\prod}_{p=a}^m A_p = 1$ if $a > m$. The Kronecker and Hadamard (element-wise) products are defined by \otimes and \odot , respectively, given any $a, b \in \mathbb{R}^n$, $a \odot b = D_{ab}$, 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. Function composition is defined by \circ where $(f \circ g)(x) \triangleq f(g(x))$. The identity matrix of size $n \times n$ is denoted by I_n . 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.

A. Model Dynamics and Control Objective

Consider an uncertain Euler-Lagrange system modeled as

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

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ denote the generalized position, velocity, and acceleration, respectively. The generalized inertia matrix,

generalized centripetal-Coriolis effects, generalized potential forces, generalized dissipation effects, the time-varying disturbances, and the control input are denoted by $M \in \mathbb{R}^{n \times n}$, $V \in \mathbb{R}^{n \times n}$, $G \in \mathbb{R}^n$, $F \in \mathbb{R}^n$, $\tau_d \in \mathbb{R}^n$, and $\tau \in \mathbb{R}^n$, respectively. The system disturbances are assumed to be bounded as $\|\tau_d(t)\| \leq \bar{d}$, where $\bar{d} \in \mathbb{R}_{>0}$ denotes a known constant. The system in (1) satisfies following properties described in [19, Sec. 2.3].

Property 1: The inertia matrix $M(q)$, satisfies $m_1\|\zeta\|^2 \leq \zeta^\top M(q)\zeta \leq m_2\|\zeta\|^2$ for all $\zeta, q \in \mathbb{R}^n$, where $m_1, m_2 \in \mathbb{R}_{>0}$ denote known constants.

Property 2: The time-derivative of the inertia matrix and centripetal-Coriolis matrix satisfy the skew-symmetry relation, $\zeta^\top (\dot{M}(q) - 2V(q, \dot{q}))\zeta = 0$, for all $q, \dot{q}, \zeta \in \mathbb{R}^n$.

The control objective is to design a PI-LSTM controller to asymptotically track a user-defined, time-varying desired trajectory, $q_d \in \mathbb{R}^n$, which is designed to be sufficiently smooth such that $q_d(t), \dot{q}_d(t), \ddot{q}_d(t) \in \mathcal{Q}$, for all $t \in \mathbb{R}_{\geq 0}$, where $\mathcal{Q} \subseteq \mathbb{R}^n$ denotes a known compact set. To quantify the objective, the tracking error $e \in \mathbb{R}^n$ and auxiliary tracking error $r \in \mathbb{R}^n$ are defined as

$$e \triangleq q - q_d, r \triangleq \dot{e} + \alpha e, \quad (2)$$

respectively, where $\alpha \in \mathbb{R}_{>0}$ denotes a user-selected constant control gain. Quantitatively, the objective is to ensure $\|e(t)\| \rightarrow 0$ and $\|r(t)\| \rightarrow 0$ as $t \rightarrow \infty$. Using (1)-(2), the open-loop dynamics for r can be determined as

$$\begin{aligned} M(q)\dot{r} &= \tau - M(q)(\ddot{q}_d - \alpha\dot{e}) - V(q, \dot{q})(\dot{q}_d - \alpha e) \\ &\quad - G(q) - F(\dot{q}) - \tau_d - V(q, \dot{q})r. \end{aligned} \quad (3)$$

III. CONTROL LAW DEVELOPMENT

A. Adaptive PI-LSTM Architecture

Recent trends show an increased implementation of PINN architectures in control design, because they use known physical properties of systems to enhance the model accuracy and generalization [6]. Leveraging the memory capabilities and improved transient performance of LSTMs, the developed method combines LSTM cells which capture dynamic time dependencies with the function approximation power of feedforward DNN terms. Motivated by the known structure of Euler-Lagrange dynamics, the developed PI-LSTM structure is developed to individually approximate the contribution of the unknown terms $M(q)$, $V(q, \dot{q})$, $G(q)$, and $F(\dot{q})$ based on the model structure (e.g., functional dependencies, vectors that multiply by each uncertainty) thereby integrating the physics into the architecture. This differs from black box approaches that approximate the entirety of the dynamics.

B. Deep Neural Network (DNN) Model

A family of feedforward DNNs $\Phi_i(x_i, \theta_i) \in \mathbb{R}^{L_{k+1,i}}$ can be modeled as [2]

$$\Phi_i(x_i, \theta_i) \triangleq \left(v_{k_i}^\top \phi_{k,i} \circ \dots \circ v_{1_i}^\top \phi_{1,i} \right) \left(v_{0_i}^\top x_{a,i} \right), \quad (4)$$

for $i \in \{M, V, F, G\}$, where all of the weights within Φ_i are represented by $\theta_i \triangleq [\text{vec}(v_{0_i})^\top, \dots, \text{vec}(v_{k_i})^\top]^\top \in$

$\mathbb{R}^{\sum_{j=0}^{k_i} L_{j,i} L_{j+1,i}}$, and $v_{j,i} \in \mathbb{R}^{L_{j,i} \times L_{j+1,i}}$ denotes the matrix of weights and biases in the j^{th} hidden layer, $L_{j,i} \in \mathbb{N}$ denotes the number of nodes within the j^{th} hidden layer for all $j \in \{0, \dots, k_i\}$, $k_i \in \mathbb{N}$ denotes the number of hidden layers, and $L_{0,i} \triangleq m_i + 1$, where m_i is the size of the input to the DNN. The vector of smooth¹ activation functions at the j^{th} layer is denoted by $\phi_{j,i} \in \mathbb{R}^{L_{j,i}}$ and is defined as $\phi_{j,i} \triangleq [\varsigma_{j,i,1} \cdots \varsigma_{j,i,L_{j-1}} \ 1]^\top$, where $\varsigma_{j,i,y} \in \mathbb{R}$ denotes the activation function at the y^{th} node of the j^{th} layer for all $j \in \{1, \dots, k_i\}$ and $i \in \{M, V, F, G\}$. The augmented input $x_{a,i} \in \mathbb{R}^{m_i+1}$ is defined as $x_{a,i} \triangleq [x_i^\top \ 1]^\top$, where $x_i \in \Omega_i$ denotes the input to the DNN, and $\Omega_i \subset \mathbb{R}^{m_i}$ denotes a compact set for all $i \in \mathcal{I}$. To incorporate a bias term into the DNN model in (4), the input x_i and the activation functions $\phi_{j,i}$ are augmented with a 1 for all $j \in \{1, \dots, k_i\}$ and $i \in \{M, V, F, G\}$.

To facilitate the development of the weight adaptation laws, the DNN model in (4) can also be represented recursively using shorthand notation $\Phi_{j,i}$ as [2]

$$\Phi_{j,i} \triangleq \begin{cases} v_{L_{j,i},i}^\top \phi_{j,i}(\Phi_{j-1,i}), & j \in \{1, \dots, k_i\}, \\ v_{0,i}^\top x_{a,i} & j = 0, \end{cases} \quad (5)$$

where $\Phi_i(x_i, \theta_i) = \Phi_{k_i,i}$, for all $i \in \{M, V, F, G\}$.

C. Long Short-Term Memory (LSTM) Model

The incorporation of LSTM cells within the PINN architecture enables NN estimates to leverage previous state information to capture any dynamical behavior that the feed-forward DNN cannot, improving prediction accuracy.

An LSTM cell for $i \in \{M, V, F, G\}$ can be modeled in continuous time as [12]

$$\begin{aligned} f(z_i, W_{f,i}) &= \sigma_g \circ W_{f,i}^\top z_i, & p(z_i, W_{p,i}) &= \sigma_g \circ W_{p,i}^\top z_i, \\ o(z_i, W_{o,i}) &= \sigma_g \circ W_{o,i}^\top z_i, & \tilde{c}(z_i, W_{c,i}) &= \sigma_c \circ W_{c,i}^\top z_i, \\ \dot{c}_i &= -b_{c,i} c_i + b_{c,i} \Psi_c(z_i, c_i, \vartheta_i), \\ \dot{h}_i &= -b_{h,i} h_i + b_{h,i} \Psi_h(z_i, c_i, \vartheta_i), \end{aligned} \quad (6)$$

where user-selected constants are given by $b_{c,i}, b_{h,i} \in \mathbb{R}_{>0}$, the cell and hidden states are denoted by $c_i \in \mathbb{R}^{l_{2,i}}$ and $h_i \in \mathbb{R}^{l_{2,i}}$ respectively where $c_i(0) = h_i(0) = 0$ for all $i \in \{M, V, F, G\}$, and $l_{2,i} \in \mathbb{R}_{>0}$ denotes the number of neurons within the LSTM. The concatenated state vector $z_i \in \mathbb{R}^{l_{1,i}}$ is defined as $z_i \triangleq [x_i^\top, h_i^\top, 1]^\top$, where $l_{1,i} \triangleq m_i + l_{2,i} + 1$. The state z_i is augmented with 1 to incorporate a bias term. The forget gate, input gate, cell gate, and output gate are represented by $f(z_i, W_{f,i}) \in \mathbb{R}^{l_{2,i}}$, $p(z_i, W_{p,i}) \in \mathbb{R}^{l_{2,i}}$, $\tilde{c}(z_i, W_{c,i}) \in \mathbb{R}^{l_{2,i}}$, and $o(z_i, W_{o,i}) \in \mathbb{R}^{l_{2,i}}$, respectively, and the vector sigmoid and tanh activation functions are denoted by $\sigma_g \in \mathbb{R}^{l_{2,i}}$ and $\sigma_c \in \mathbb{R}^{l_{2,i}}$, respectively. The weight matrices are given by $W_{f,i}^\top, W_{c,i}^\top, W_{p,i}^\top, W_{o,i}^\top \in \mathbb{R}^{l_{2,i} \times l_{1,i}}$, and $W_{h,i}^\top \in \mathbb{R}^{l_{3,i} \times l_{2,i}}$, where the size of the output is defined as $l_{3,i} \triangleq n$ for $i \in \{F, G\}$ and $l_{3,i} \triangleq n^2$ for $i \in \{M, V\}$, and the collection of adaptive weight estimates is given by $\vartheta_i \triangleq [\text{vec}(W_{c,i})^\top, \text{vec}(W_{p,i})^\top, \text{vec}(W_{f,i})^\top, \text{vec}(W_{o,i})^\top, \text{vec}(W_{h,i})^\top]^\top \in \mathbb{R}^{4l_{2,i} + l_{2,i} l_{3,i}}$ for $i \in \{M, V, F, G\}$. The functions $\Psi_{c,i}(z_i, c_i, \vartheta_i) \in \mathbb{R}^{l_{2,i}}$ and

$\Psi_{h,i}(z_i, c_i, \vartheta_i) \in \mathbb{R}^{l_{2,i}}$ in the cell and hidden state dynamics are defined as

$$\begin{aligned} \Psi_c(z_i, c_i, \vartheta_i) &\triangleq f(z_i, W_{f,i}) \odot c_i + p(z_i, W_{p,i}) \odot \tilde{c}(z_i, W_{c,i}), \\ \Psi_h(z_i, c_i, \vartheta_i) &\triangleq o(z_i, W_{o,i}) \odot (\sigma_c \circ \Psi_c(z_i, c_i, \vartheta_i)), \end{aligned}$$

respectively. To ensure the output of the LSTM is of the appropriate dimension, a fully-connected layer with weight matrix W_h is added to the LSTM cell. Thus, the output of the i^{th} LSTM $\Xi_i(z_i, c_i, \vartheta_i) \in \mathbb{R}^{l_{3,i}}$ can be modeled as

$$\Xi_i(z_i, c_i, \vartheta_i) = W_{h,i}^\top \Psi_h(z_i, c_i, \vartheta_i), \quad (7)$$

for $i \in \{M, V, F, G\}$.

D. Adaptive PI-LSTM Control Strategy

Using the developed PI-LSTM architecture, an adaptive estimate of the dynamics is developed and implemented in the subsequently designed controller. The universal function approximation property states that the space of DNNs given by (5) is dense in $\mathcal{C}(\Omega_i)$, where $\mathcal{C}(\Omega_i)$ denotes a space of continuous functions over Ω_i [20, Th. 3.2]. Therefore, for any given $f_i \in \mathcal{C}(\Omega_i)$ and prescribed $\bar{\varepsilon}_i \in \mathbb{R}_{>0}$, there exist some $k_i, L_{j,i} \in \mathbb{N}$, and corresponding ideal weights and biases $\theta_{j,i} \in \mathbb{R}^{L_{j,i} \times L_{j+1,i}}$, $\forall j \in \{0, \dots, k_i\}$, such that $\sup_{x_i \in \Omega_i} \|f_i(x_i) - \chi_i(x_i, \theta_i)\| \leq \bar{\varepsilon}_i$, for all $i \in \mathcal{I}$. Based on this property, the unknown terms $M(q)$, $V(q, \dot{q})$, $G(q)$, and $F(\dot{q})$ can be modeled as

$$\text{vec}(M(q)) = \chi_M(x_M, \theta_M, c_M, h_M, \vartheta_M) + \varepsilon_M(x_M), \quad (8)$$

$$\text{vec}(V(q, \dot{q})) = \chi_V(x_V, \theta_V, c_V, h_V, \vartheta_V) + \varepsilon_V(x_V), \quad (9)$$

$$G(q) = \chi_G(x_G, \theta_G, c_G, h_G, \vartheta_G) + \varepsilon_G(x_G), \quad (10)$$

$$F(\dot{q}) = \chi_F(x_F, \theta_F, c_F, h_F, \vartheta_F) + \varepsilon_F(x_F), \quad (11)$$

where the function χ_i defined as $\chi_i(x_i, \theta_i, c_i, h_i, \vartheta_i) \triangleq \Phi_i(x_i, \theta_i) + \Xi_i(x_i, c_i, h_i, \vartheta_i)$ represents the combined DNN-LSTM estimate, where $\chi_i \in \mathbb{R}^{l_{3,i}}$ for $i \in \{M, V, F, G\}$. The unknown function approximation errors are denoted as $\varepsilon_i \in \mathbb{R}^{l_{3,i}}$ for $i \in \{M, V, F, G\}$.² The inputs for each DNN-LSTM are denoted as $x_M \triangleq q$, $x_V \triangleq [q^\top, \dot{q}^\top]^\top$, $x_G \triangleq q$, and $x_F \triangleq \dot{q}$. The following assumption facilitates the subsequent development.

Assumption 1: There exists known constants $\bar{\theta}_i \in \mathbb{R}_{>0}$, $\bar{\vartheta}_i \in \mathbb{R}_{>0}$ such that the unknown ideal weights can be bounded as $\|\theta_i\| \leq \bar{\theta}_i$, $\|\vartheta_i\| \leq \bar{\vartheta}_i$ for all $i \in \{M, V, F, G\}$ [21, Assumption 1].

To ensure an appropriate output dimension of the PI-LSTM estimate, the vectorization operator is applied to $M(q)$ and $V(q, \dot{q})$. Using properties of the vectorization operator and using (8)-(11) yields

$$\begin{aligned} M(q)\dot{r} &= \tau - \tau_d - \left((\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n \right) (\chi_M + \varepsilon_M(x_M)) \\ &\quad - V(q, \dot{q})r - \left((\dot{q}_d - \alpha e)^\top \otimes I_n \right) (\chi_V + \varepsilon_V(x_V)) \\ &\quad - (\chi_G + \varepsilon_G(x_G)) - (\chi_F + \varepsilon_F(x_F)). \end{aligned} \quad (12)$$

²The output of the DNN and the LSTM are of the same dimension such that $l_{3,i} = L_{k+1,i}$. Therefore the dimension of χ_i can be equivalently expressed as $\chi_i \in \mathbb{R}^{L_{k+1,i}}$ for all $i \in \{M, V, F, G\}$.

¹To consider nonsmooth activation functions, the switched systems analysis in [2] can be used with the subsequent control development.

Based on the subsequent stability analysis, an adaptive PI-LSTM control input³ is designed as

$$\begin{aligned} \tau(t) = & \left((\dot{q}_d - \alpha e)^\top \otimes I_n \right) \widehat{\chi}_V + \widehat{\chi}_G + \widehat{\chi}_F - k_1 r - e \\ & + \left((\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n \right) \widehat{\chi}_M - \text{sgn}(r) \left(k_2 \right. \\ & \left. + k_3 \|(\dot{q}_d - \alpha e)^\top \otimes I_n\| + k_4 \|(\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\| \right), \end{aligned} \quad (13)$$

where $k_1, k_2, k_3, k_4 \in \mathbb{R}_{>0}$ are user-defined control gains, and $\widehat{\chi}_i \triangleq \chi_i(x_i, \hat{\theta}_i, \hat{c}_i, \hat{h}_i, \hat{v}_i) \in \mathbb{R}^{l_{3,i}}$ denotes the combined DNN and LSTM estimate for all $i \in \{M, V, G, F\}$. Let the adaptive DNN and LSTM weight estimates be denoted as $\hat{\theta}_i \triangleq [\text{vec}(\hat{v}_{0,i})^\top, \dots, \text{vec}(\hat{v}_{k_i,i})^\top]^\top \in \mathbb{R}^{\sum_{j=0}^{k_i} L_{j,i} L_{j+1,i}}$ and $\hat{v}_i \triangleq [\text{vec}(\widehat{W}_{c,i})^\top, \text{vec}(\widehat{W}_{p,i})^\top, \text{vec}(\widehat{W}_{f,i})^\top, \text{vec}(\widehat{W}_{o,i})^\top, \text{vec}(\widehat{W}_{h,i})^\top]^\top \in \mathbb{R}^{4l_{2,i}l_{1,i}+l_{2,i}l_{3,i}}$ for all $i \in \{M, V, G, F\}$. Using the LSTM model in (6) and the adaptive weight estimates \hat{v}_i , the estimates of the cell state \hat{c}_i and hidden state \hat{h}_i dynamics are

$$\begin{aligned} \dot{\hat{c}}_i = & -b_{c,i} \hat{c}_i + b_{c,i} (f(\hat{z}_i, \widehat{W}_{f,i}) \odot c_i \\ & + p(\hat{z}_i, \widehat{W}_{p,i}) \odot \tilde{c}_i(\hat{z}_i, \widehat{W}_{c,i})), \end{aligned} \quad (14)$$

$$\dot{\hat{h}}_i = -b_{h,i} \hat{h}_i + b_{h,i} \left(o(\hat{z}_i, \widehat{W}_{o,i}) \odot \sigma_{c,i} \circ \Psi_{c,i}(\hat{z}_i, \hat{c}_i, \hat{v}_i) \right), \quad (15)$$

respectively, for all $i \in \{M, V, G, F\}$, where $\hat{z}_i \triangleq [x_i^\top, \hat{h}_i^\top, 1]^\top$ denotes the augmented input of the LSTM.

Substituting (13) into (12) yields

$$\begin{aligned} M(q)\dot{r} = & - \left((\dot{q}_d - \alpha e)^\top \otimes I_n \right) (\chi_V - \widehat{\chi}_V + \varepsilon_V) \\ & - \left((\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n \right) (\chi_M - \widehat{\chi}_M + \varepsilon_M) - e \\ & - \chi_G + \widehat{\chi}_G - \chi_F + \widehat{\chi}_F - k_1 r - \tau_d - V(q, \dot{q}) \\ & - \text{sgn}(r) \left(k_2 + k_3 \|(\dot{q}_d - \alpha e)^\top \otimes I_n\| \right) \\ & - k_4 \text{sgn}(r) \|(\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\| - \varepsilon_G - \varepsilon_F. \end{aligned} \quad (16)$$

IV. PI-LSTM WEIGHT ADAPTATION LAWS

The development of Lyapunov-based adaptation laws for the PI-LSTM architecture allows for continuous real-time adaptation. Based on the subsequent stability analysis, the weight adaptation laws are designed as

$$\dot{\widehat{\chi}}_M = -\Gamma_M \widehat{\chi}_M^\top \left((\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n \right)^\top r, \quad (17)$$

$$\dot{\widehat{\chi}}_V = -\Gamma_V \widehat{\chi}_V^\top \left((\dot{q}_d - \alpha e)^\top \otimes I_n \right)^\top r, \quad (18)$$

$$\dot{\widehat{\chi}}_F = -\Gamma_F \widehat{\chi}_F^\top r, \quad (19)$$

$$\dot{\widehat{\chi}}_G = -\Gamma_G \widehat{\chi}_G^\top r, \quad (20)$$

where $\widehat{\chi}_i \triangleq [\hat{v}_i^\top, \hat{\theta}_i^\top]^\top$, and $\Gamma_i \in \mathbb{R}^{l_{4,i} \times l_{4,i}}$ is a positive-definite adaptation gain matrix, where $l_{4,i} \triangleq 4l_{2,i}l_{1,i} + l_{2,i}l_{3,i} + \sum_{j=0}^{k_i} L_{j,i}L_{j+1,i}$ for $i \in \{M, V, G, F\}$ and $j \in \{0, \dots, k_i\}$. The shorthand notation $\widehat{\chi}'_i$ denotes the Jacobian $\widehat{\chi}'_i \triangleq \frac{\partial \chi_i(x_i, \hat{\theta}_i, \hat{c}_i, \hat{h}_i, \hat{v}_i)}{\partial \widehat{\chi}_i}$. Therefore, the Jacobian can be expressed as

$$\widehat{\chi}'_i \triangleq [\widehat{\Xi}'_i, \widehat{\Phi}'_i], \text{ where } \widehat{\Xi}'_i \triangleq \frac{\partial \Xi_i(x_i, \hat{c}_i, \hat{h}_i, \hat{v}_i)}{\partial \hat{v}_i} \text{ and } \widehat{\Phi}'_i \triangleq \frac{\partial \Phi_i(x_i, \hat{\theta}_i)}{\partial \hat{\theta}_i}$$

³The sgn function in (13) is motivated to obtain an asymptotic result given the function approximation error and added disturbance terms.

denote the Jacobians of the LSTM and feedforward DNN for $i \in \{M, V, G, F\}$, respectively.

A. Jacobians of the LSTM Architecture

The Jacobian of the LSTMs $\widehat{\Xi}'_i$ can be represented as $\widehat{\Xi}'_i \triangleq [\widehat{\Xi}'_{W_{c,i}}, \widehat{\Xi}'_{W_{p,i}}, \widehat{\Xi}'_{W_{f,i}}, \widehat{\Xi}'_{W_{o,i}}, \widehat{\Xi}'_{W_{h,i}}]$, where the Jacobian of each weight estimate can be expressed as $\widehat{\Xi}'_{W_{j,i}} \triangleq \frac{\partial \Xi_i(x_i, \hat{c}_i, \hat{h}_i, \hat{v}_i)}{\partial \text{vec}(W_{j,i})}$ for $j \in \{c, p, f, o, h\}$ and $i \in \{M, V, G, F\}$. Based on (7) and the chain rule, $\widehat{\Xi}'_{W_{h,i}}$ and $\widehat{\Xi}'_{W_{j,i}}$ can be written using $\widehat{\Xi}'_{W_{h,i}} = I_n \otimes \Psi_h^\top(\hat{z}_i, \hat{c}_i, \hat{v}_i)$, $\widehat{\Xi}'_{W_{j,i}} = \widehat{W}_{h,i}^\top \widehat{\Psi}'_{h,W_{j,i}}$, and $\widehat{\Psi}'_{h,W_{j,i}} \triangleq \frac{\partial \Psi_h(\hat{z}_i, \hat{c}_i, \hat{v}_i)}{\partial \text{vec}(W_{j,i})}$ for all $j \in \{c, p, f, o\}$ and $i \in \{M, V, G, F\}$. Using (6), (14)-(15), the chain rule, properties of the Hadamard product, and the vectorization operator, the terms $\widehat{\Psi}'_{h,W_j}$ and $\widehat{\Psi}'_{h,W_o}$ can be written as [12]

$$\widehat{\Psi}'_{h,W_{j,i}} = \text{diag} \left(\sigma_{g,i} \left(\widehat{W}_{o,i}^\top \hat{z}_i \right) \right) \sigma'_{c,i} \left(\Psi_c(\hat{z}_i, \hat{c}_i, \hat{v}_i) \right) \widehat{\Psi}'_{c,W_{j,i}},$$

$$\widehat{\Psi}'_{h,W_{o,i}} = \text{diag} \left(\sigma_{c,i} \left(\Psi_c(\hat{z}_i, \hat{c}_i, \hat{v}_i) \right) \right) \left(\sigma'_{g,i} \left(\widehat{W}_{o,i}^\top \hat{z}_i \right) \right) \left(I_{l_{2,i}} \otimes \hat{z}_i^\top \right),$$

respectively, for all $j \in \{c, p, f\}$ and $i \in \{M, V, G, F\}$, where $\widehat{\Psi}'_{c,W_{j,i}} \triangleq \frac{\partial \Psi_{c,i}(\hat{z}_i, \hat{c}_i, \hat{v}_i)}{\partial \text{vec}(W_{j,i})}$, and the gradient of the sigmoid and tanh activation functions are expressed using the shorthand notation $\sigma'_{g,i} \in \mathbb{R}^{l_{2,i} \times l_{2,i}}$ and $\sigma'_{c,i} \in \mathbb{R}^{l_{2,i} \times l_{2,i}}$, respectively. Similarly, the terms $\widehat{\Psi}'_{c,W_{c,i}}$, $\widehat{\Psi}'_{c,W_{p,i}}$, and $\widehat{\Psi}'_{c,W_{f,i}}$ can be expressed as

$$\widehat{\Psi}'_{c,W_{c,i}} = \text{diag} \left(\sigma_{g,i} \left(\widehat{W}_{p,i}^\top \hat{z}_i \right) \right) \sigma'_{c,i} \left(W_{c,i}^\top \hat{z}_i \right) \left(I_{l_{2,i}} \otimes \hat{z}_i^\top \right),$$

$$\widehat{\Psi}'_{c,W_{p,i}} = \text{diag} \left(\sigma_{c,i} \left(\widehat{W}_{c,i}^\top \hat{z}_i \right) \right) \sigma'_{g,i} \left(W_{p,i}^\top \hat{z}_i \right) \left(I_{l_{2,i}} \otimes \hat{z}_i^\top \right),$$

$$\widehat{\Psi}'_{c,W_{f,i}} = \text{diag}(\hat{c}_i) \sigma'_{g,i} \left(\widehat{W}_{f,i}^\top \hat{z}_i \right) \left(I_{l_{2,i}} \otimes \hat{z}_i^\top \right),$$

respectively, for all $i \in \{M, V, F, G\}$.

B. Jacobians of the Feedforward DNN Architecture

The Jacobians of the feedforward DNNs $\widehat{\Phi}'_{j,i} \triangleq \Phi_{j,i}(x_i, \hat{v}_{0,i}, \dots, \hat{v}_{j,i})$ and their respective Jacobians $\widehat{\Phi}'_i$ can be represented $\widehat{\Phi}'_i \triangleq [\widehat{\Phi}'_{0,i}, \dots, \widehat{\Phi}'_{j,i}]$, where the shorthand notation $\widehat{\Phi}'_{j,i}$ is defined as $\widehat{\Phi}'_{j,i} \triangleq \frac{\partial \Phi_{j,i}(x_i, \hat{\theta}_i)}{\partial \hat{\theta}_i}$, for all $j \in \{0, \dots, k_i\}$ and $i \in \{M, V, G, F\}$. Using (5), the chain rule, and properties of the vectorization operator, the terms $\widehat{\Phi}'_{0,i}$ and $\widehat{\Phi}'_{j,i}$ can be expressed as [14]

$$\widehat{\Phi}'_{0,i} \triangleq \left(\prod_{l=1}^{k_i} \hat{v}_{l,i}^\top \hat{\phi}'_{l,i} \right) \left(I_{L_{1,i}} \otimes x_{a,i}^\top \right),$$

$$\widehat{\Phi}'_{j,i} \triangleq \left(\prod_{l=j+1}^{k_i} \hat{v}_{l,i}^\top \hat{\phi}'_{l,i} \right) \left(I_{L_{j+1,i}} \otimes \hat{\phi}_{j,i}^\top \right),$$

for all $j \in \{1, \dots, k_i\}$ and $i \in \{M, V, G, F\}$, respectively, where the activation function at the j^{th} layer and its Jacobian are expressed using the shorthand notations $\hat{\phi}_{j,i} \triangleq \phi_{j,i}(\Phi_{j-1,i}(x_i, \hat{\theta}_i))$ and $\hat{\phi}'_{j,i} \triangleq \phi'_{j,i}(\Phi_{j-1,i}(x_i, \hat{\theta}_i))$, respectively, where $\phi'_{j,i} : \mathbb{R}^{L_{j,i}} \rightarrow \mathbb{R}^{L_{j,i} \times L_{j,i}}$ is defined as $\phi'_{j,i}(y) \triangleq \frac{\partial}{\partial Q} \phi_{j,i}(Q)|_{Q=y}$, for all $y \in \mathbb{R}^{L_{j,i}}$.

V. STABILITY ANALYSIS

To address the additional level of complexity due to the nonlinearity of the DNN and LSTM weight estimates, a first-order Taylor series approximation-based error model is given by [2, eq. (22)].

$$\chi_i(x_i, \theta_i, \hat{c}_i, \hat{h}_i, \vartheta_i) - \hat{\chi}_i = \hat{\chi}'_i \tilde{Z}_i + \mathcal{O}_i^2(\|\tilde{Z}_i\|), \quad (21)$$

where $\mathcal{O}_i^2(\|\tilde{Z}_i\|)$ denotes the higher-order terms, $\tilde{\vartheta} \triangleq \vartheta - \hat{\vartheta}$, $\tilde{\theta} \triangleq \theta - \hat{\theta}$ and $\tilde{Z}_i \triangleq [\tilde{\vartheta}_i^\top, \tilde{\theta}_i^\top]^\top$ for $i \in \{M, V, G, F\}$. Adding and subtracting $\chi(x_i, \theta_i, \hat{c}_i, \hat{h}_i, \vartheta_i)$, $\forall i \in \{M, V, G, F\}$, and using (21), the closed loop error system can be written as

$$\begin{aligned} M(q)\dot{r} = & -\left((\dot{q}_d - \alpha e)^\top \otimes I_n\right) \left(\hat{\chi}'_V \tilde{Z}_V + N_{1,V}\right) \\ & - \left((\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\right) \left(\hat{\chi}'_M \tilde{Z}_M + N_{1,M}\right) \\ & - \hat{\chi}'_G \tilde{Z}_G - N_{1,G} - \hat{\chi}'_F \tilde{Z}_F - N_{1,F} \\ & - \text{sgn}(r) \left(k_2 + k_3 \|(\dot{q}_d - \alpha e)^\top \otimes I_n\|\right) \\ & - k_4 \text{sgn}(r) \|(\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\| \\ & - k_1 r - e - \tau_d - V(q, \dot{q}), \end{aligned} \quad (22)$$

where the auxiliary function $N_{1,i} \in \mathbb{R}^{l_{3,i}}$ is defined as $N_{1,i} \triangleq \chi_i(x_i, \theta_i, c_i, h_i, \vartheta_i) - \hat{\chi}_i(x_i, \theta_i, \hat{c}_i, \hat{h}_i, \vartheta_i) + \varepsilon_i(x_i) + \mathcal{O}_i^2(\|\tilde{Z}_i\|)$ for all $i \in \{M, V, G, F\}$, and $N_{1,i}$ can be bounded as $\|N_{1,i}\| \leq \bar{N}_{1,i}$, for a known constant $\bar{N}_{1,i} \in \mathbb{R}_{>0}$ [12, Lemma 1].

To facilitate the subsequent stability analysis, let the concatenated state vector $\zeta \in \mathbb{R}^\psi$ be defined as $\zeta \triangleq [e^\top, r^\top, \tilde{Z}_M^\top, \tilde{Z}_V^\top, \tilde{Z}_F^\top, \tilde{Z}_G^\top]^\top$, where $\psi \triangleq 2n + \sum_{i \in \mathcal{I}} l_{4,i}$ with $\mathcal{I} \triangleq \{M, V, G, F\}$, and let the open and connected set $\mathcal{B}_\Lambda \subset \mathbb{R}^\psi$ be defined as $\mathcal{B}_\Lambda \triangleq \{\xi \in \mathbb{R}^\psi : \|\xi\| \leq \sqrt{\frac{\beta_1}{\beta_2}} \omega\}$, where $\omega \in \mathbb{R}_{>0}$ denotes a user selected constant. The following theorem establishes tracking error convergence using the developed PI-LSTM-based adaptive controller.

Theorem 1: For the dynamical system in (1), the controller in (13) and the adaptation laws developed in (17)-(20) ensure asymptotic tracking in the sense that $\|e(t)\| \rightarrow 0$ and $\|r(t)\| \rightarrow 0$ as $t \rightarrow \infty$, provided Assumption 1 holds, $\zeta(t_0) \in \mathcal{B}_\Lambda$, and the gain conditions $k_2 \geq \bar{d} + \bar{N}_{1,F} + \bar{N}_{1,G}$, $k_3 \geq \bar{N}_{1,V}$, and $k_4 \geq \bar{N}_{1,M}$ are satisfied.

Proof: Consider the candidate Lyapunov function $\mathcal{V}_L \in \mathbb{R}_{\geq 0}$ defined as

$$\mathcal{V}_L(\zeta) \triangleq \frac{1}{2} e^\top e + \frac{1}{2} r^\top r + \sum_{i \in \mathcal{I}} \frac{1}{2} \tilde{Z}_i^\top \Gamma_i^{-1} \tilde{Z}_i. \quad (23)$$

The candidate Lyapunov function in (23) satisfies the inequality $\beta_1 \|\zeta\|^2 \leq \mathcal{V}_L(\zeta) \leq \beta_2 \|\zeta\|^2$, where $\beta_1 \triangleq \min\{\frac{1}{2}, \frac{1}{2}m_1, \frac{1}{2}\min_{i \in \mathcal{I}}(\lambda_{\min}(\Gamma_i))\}$ and $\beta_2 \triangleq \max\{\frac{1}{2}, \frac{1}{2}m_2, \frac{1}{2}\max_{i \in \mathcal{I}}(\lambda_{\max}(\Gamma_i))\}$. Taking the time-derivative of $\mathcal{V}_L(\zeta)$, applying the chain rule for differential inclusions in [22, Th. 2.2], and applying (16) and Property 2 yields

$$\begin{aligned} \dot{\mathcal{V}}_L(\zeta) \stackrel{a.a.t.}{\leq} & -e^\top \alpha e + r^\top \left(-\hat{\chi}'_G \tilde{Z}_G - N_{1,G} - \hat{\chi}'_F \tilde{Z}_F\right) \\ & - N_{1,F} - \left((\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\right) \left(\hat{\chi}'_M \tilde{Z}_M + N_{1,M}\right) \\ & - \left((\dot{q}_d - \alpha e)^\top \otimes I_n\right) \left(\hat{\chi}'_V \tilde{Z}_V + N_{1,V}\right) \\ & - \varepsilon_G - \varepsilon_F - \tau_d - k_1 r - K[\text{sgn}](r) \left(k_2\right. \end{aligned}$$

$$\begin{aligned} & \left. + k_3 \|(\dot{q}_d - \alpha e)^\top \otimes I_n\| + k_4 \|(\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\|\right) \\ & - \sum_{i \in \mathcal{I}} \left(\frac{1}{2} \tilde{Z}_i^\top \Gamma_i^{-1} \tilde{Z}_i\right). \end{aligned} \quad (24)$$

Substituting (17)-(21), and combining like terms yields

$$\begin{aligned} \dot{\mathcal{V}}_L(\zeta) \stackrel{a.a.t.}{\leq} & -e^\top \alpha e - r^\top \left(\left((\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\right) N_{1,M} + N_{1,G}\right) \\ & + k_1 r + K[\text{sgn}](r) \left(k_2 + k_3 \|(\dot{q}_d - \alpha e)^\top \otimes I_n\|\right) \\ & + k_4 K[\text{sgn}](r) \|(\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\| + \tau_d \\ & + \left((\dot{q}_d - \alpha e)^\top \otimes I_n\right) N_{1,V} + N_{1,F}. \end{aligned} \quad (25)$$

Provided the stated gain conditions are satisfied, (25) can be bounded as

$$\dot{\mathcal{V}}_L(\zeta) \stackrel{a.a.t.}{\leq} -\alpha \|e\|^2 - k_1 \|r\|^2. \quad (26)$$

Using (23) and (26) implies $e, r, \tilde{\vartheta}_i, \tilde{\theta}_i \in \mathcal{L}_\infty$. The fact that $q_d, \dot{q}_d, e, r \in \mathcal{L}_\infty$ implies $q, \dot{q} \in \mathcal{L}_\infty$. To show $x_i \in \Omega_i$, and therefore that the universal function approximation property holds for all $i \in \{M, V, G, F\}$, let the open and connected sets $\Upsilon_i \subseteq \Omega_i$ be defined as $\Upsilon_M \triangleq \{\xi \in \Omega_M : \|\xi\| < \bar{q}_d + \omega\}$, $\Upsilon_V \triangleq \{\xi \in \Omega_V : \|\xi\| < \bar{q}_d + (2 + \alpha)\omega + \bar{q}_d\}$, $\Upsilon_F \triangleq \{\xi \in \Omega_F : \|\xi\| < (1 + \alpha)\omega + \bar{q}_d\}$, and $\Upsilon_G \triangleq \{\xi \in \Omega_G : \|\xi\| < \bar{q}_d + \omega\}$. The facts that $\dot{\mathcal{V}}_L(\zeta(t)) \stackrel{a.a.t.}{\leq} 0$, and $\beta_1 \|\zeta\|^2 \leq \mathcal{V}_L(\zeta) \leq \beta_2 \|\zeta\|^2$ imply that $\zeta(t)$ can be bounded as $\|\zeta(t)\| \leq \sqrt{\frac{\beta_2}{\beta_1}} \|\zeta(t_0)\|$. If $\|\zeta(t_0)\| \in \sqrt{\frac{\beta_1}{\beta_2}} \omega$, then $\|\zeta(t)\| \leq \omega$, and therefore $\|e(t)\| \leq \omega$ and $\|r(t)\| \leq \omega$. Therefore, if $\zeta(t_0) \in \mathcal{B}_\Lambda$, then $x_i \in \Upsilon_i \subseteq \Omega_i$. Then, using (23), (26), the extension of the LaSalle-Yoshizawa corollary in [23, Corollary 1] can be invoked to show that $\|e(t)\| \rightarrow 0$ and $\|r(t)\| \rightarrow 0$ as $t \rightarrow \infty$. ■

VI. SIMULATIONS

Simulation results are provided to demonstrate the performance of the developed method using a two-link planar revolute robot as modeled in [12]. Simulations were ran for 50 s with the selected desired trajectory $q_d(t) \triangleq [q_{d1}, q_{d2}]^\top \in \mathbb{R}^2$ as $q_d \triangleq (1 - \exp(-0.1)) \begin{bmatrix} \frac{3\pi}{8} \sin(\frac{\pi}{2}t) \\ \frac{3\pi}{8} \sin(\frac{\pi}{2}t) \end{bmatrix} \in \mathbb{R}^2$ [rad] with sensor and process noise generated from a Gaussian distribution with standard deviation of 0.001 was injected into the simulation data. The simulation is initialized at $q(0) = [0.4, -0.3]^\top$ [rad] and $\dot{q}(0) = [0, 0]^\top$ [rad/s]. To highlight the contribution of the combined estimation power of DNNs and LSTMs, the PI-LSTM was composed of 3 of DNNs with 4 layers and 7 neurons and tanh activation functions for the M, V , and F matrices while the LSTM model was used with $l_{2,i} = 2$ neurons for $i \in \{M, V, F\}$. The resulting architecture has 817 individual weights, with the DNN and LSTM having 661 and 156 individual weights, respectively. A comparative simulation was performed with the architecture developed in [14] as a baseline (i.e., without the contribution of the memory properties gained from the LSTMs). The weights of the DNNs and LSTMs were randomly initialized from a uniform distribution $U(-1, 1)$. The gains were selected as shown in Table I with the learning gains Γ_i selected as $\Gamma_i = \text{diag}[\Gamma_{1,i}, \Gamma_{2,i}]$, where $\Gamma_{1,i} = \Gamma_{i,LSTM} \cdot I_{4l_{2,i}+l_{2,i}l_{3,i}}$,

TABLE I
CONTROL GAINS

	Control gains					Γ_{DNN}			Γ_{LSTM}			b_e			b_h		
	α	k_1	k_2	k_3	k_4	M	V	F	M	V	F	M	V	F	M	V	F
Baseline Architecture [14]	2.2	9.6	0.7	0.1	0.4	1.6	3.1	2.3	-	-	-	-	-	-	-	-	-
Developed Architecture	7	0.8	0.6	0.7	0.2	1.2	3.6	1.9	1	10.8	9.2	5.9	7.6	1.6	2.7	7.7	2

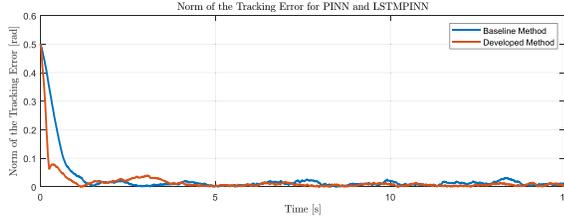


Fig. 1. Comparison of tracking error norm for the baseline developed in [14] and the developed method.

and $\Gamma_{2,i} = \Gamma_{i,DNN} \cdot I_{\sum_{j=0}^{k_i} L_{j,i} L_{j+1,i}}$ for $i \in \{M, V, F\}$. The root mean squared (RMS) tracking error for the Lb-PINN and PI-LSTM controllers were of 0.0279 rad and 0.0185 rad, respectively. Although both architectures achieved rapid tracking error convergence as shown in Figure 1, the developed PI-LSTM converged faster and was more robust to the injected noise. The developed method yielded a 33.76% improvement compared to the developed architecture in [14].

VII. CONCLUSION

This letter provides the first result on Lyapunov-derived adaptation laws for the weights of a novel PI-LSTM architecture. The combination of the LSTM term within a physics-informed structure allows the architecture to effectively capture long term dynamic behavior. Comparative simulation results resulted in a 0.0185 rad RMS tracking error yielding a 33.76% improvement over the baseline. Future work would include constraining the output of the PI-LSTM to further respect other physical properties such as the positive definiteness of the inertia matrix.

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 agency.

REFERENCES

- [1] G. Joshi and G. Chowdhary, "Deep model reference adaptive control," in *Proc. IEEE Conf. Decis. Control*, 2019, pp. 4601–4608.
- [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] E. Vacchini, N. Sacchi, G. P. Incremona, and A. Ferrara, "Design of a deep neural network-based integral sliding mode control for nonlinear systems under fully unknown dynamics," *IEEE Control Syst. Lett.*, vol. 7, pp. 1789–1794, 2023.
- [4] S. Li, H. T. Nguyen, and C. C. Cheah, "A theoretical framework for end-to-end learning of deep neural networks with applications to robotics," *IEEE Access*, vol. 11, pp. 21992–22006, 2023.
- [5] D. Rolnick and M. Tegmark, "The power of deeper networks for expressing natural functions," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–14.
- [6] M. Lutter, C. Ritter, and J. Peters, "Deep lagrangian networks: Using physics as model prior for deep learning," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.
- [7] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," 2020, *arXiv:2003.04630*.
- [8] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nat. Rev. Phys.*, vol. 3, pp. 422–440, May 2021.
- [9] M. Lutter, J. Silberbauer, J. Watson, and J. Peters, "Differentiable physics models for real-world offline model-based reinforcement learning," in *Proc. IEEE Int. Conf. Robot.*, 2021, pp. 4163–4170.
- [10] M. Finzi, K. A. Wang, and A. G. Wilson, "Simplifying hamiltonian and lagrangian neural networks via explicit constraints," in *Proc. 34th Conf. Neural Inf. Process. Syst.*, 2020, pp. 13880–13889.
- [11] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, 2019, pp. 1–11.
- [12] E. Griffis, O. Patil, Z. Bell, and W. E. Dixon, "Lyapunov-based long short-term memory (Lb-LSTM) neural network-based control," *IEEE Control Syst. Lett.*, vol. 7, pp. 2976–2981, 2023.
- [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 Conf. Decis. Control (CDC)*, 2022, pp. 3487–3492.
- [14] R. Hart, O. Patil, E. Griffis, and W. E. Dixon, "Deep Lyapunov-based physics-informed neural networks (DeLb-PINN) for adaptive control design," in *Proc. IEEE Conf. Decis. Control*, 2023.
- [15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Dec. 1997.
- [16] 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. [Online]. Available: <https://doi.org/10.1155/2021/8882312>
- [17] E. Inanc, Y. Gurses, A. Habboush, and Y. Yildiz, "Long short-term memory for improved transients in neural network adaptive control," *IEEE Control Syst. Lett.*, 2023, pp. 3614–3619.
- [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] R. Ortega, A. Loria, P. J. Nicklasson, and H. J. Sira-Ramirez, *Passivity-Based Control of Euler-Lagrange Systems: Mechanical, Electrical and Electromechanical Applications*. London, U.K.: Springer, 1998.
- [20] P. Kidger and T. Lyons, "Universal approximation with deep narrow networks," in *Proc. Conf. Learn. Theory*, 2020, pp. 2306–2327.
- [21] F. L. Lewis, A. Yegildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 388–399, Mar. 1996.
- [22] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 39, no. 9, pp. 1910–1914, Sep. 1994.
- [23] 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.