International Journal of Adaptive Control and Signal Processing

WILEY

RESEARCH ARTICLE OPEN ACCESS

Neural Network Adaptive Control With Long Short-Term Memory

Emirhan Inanc[1] | Abdullah Habboush[2] | Yigit Gurses[3] | Yildiray Yildiz[4] | Anuradha M. Annaswamy[5]

[1]Biomedical Engineering, Illinois Institute of Technology, Illinois, USA | [2]Mechanical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands | [3]Informatics, Technical University of Munich, Munich, Germany | [4]Mechanical Engineering, Bilkent University, Ankara, Turkey | [5]Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, USA

Correspondence: Emirhan Inanc (einanc@hawk.iit.edu)

Received: 5 June 2024 | **Revised:** 24 February 2025 | **Accepted:** 2 May 2025

Funding: This work was supported by The Scientific and Technological Research Council of Türkiye under Grant 121E384.

Keywords: adaptive control | long short-term memory | neural networks | uncertain systems

ABSTRACT

In this study, we propose a novel adaptive control architecture that provides dramatically better transient response performance compared to conventional adaptive control methods. This is accomplished by the synergistic employment of a traditional adaptive neural network (ANN) controller and a long short-term memory (LSTM) network. LSTM structures can take advantage of the dependencies in an input sequence, which can help predict uncertainty. We introduce a training approach through which the LSTM network learns to compensate for the deficiencies of the ANN controller. This improves the transient response of the system and allows the controller to quickly react to unexpected events. Through careful simulation studies, we demonstrate that this architecture improves the estimation accuracy on a diverse set of uncertainties. We also provide an analysis of the contributions of the ANN controller and the LSTM network, identifying their roles in compensating low- and high-frequency error dynamics. This analysis provides insight into why and how the LSTM augmentation improves the system's transient response. The stability of the overall system is analyzed via a rigorous Lyapunov analysis.

1 | Introduction

The work of Narendra and Parthasarathy [1] in 1990 lays the foundations for using *neural networks* (NNs) to approximate nonlinear functions in the context of adaptive control and identification. Numerous subsequent studies have lead to well-established *adaptive neural network* (ANN) techniques. Initial efforts involve using linearly parameterized NNs, such as radial basis function NNs, allowing for a tractable Lyapunov analysis to update only the outer NN weights online [2–6]. As hidden-layer weights are maintained fixed, these approaches require prior selection of suitable basis functions or preliminary off-line hidden-layer tuning. These issues are addressed in the work of Lewis et al. [7], where an ANN with tunable hidden-layer weights is proposed for a general serial robot manipulator. By adjusting the hidden layer, the NN becomes nonlinear in its tunable weights, resulting in a significantly more complex problem, which is revisited by Patkar and Annaswamy [8] for a single hidden layer ANN, and by Patil et al. [9] for a deep ANN.

The performance of these nonlinear adaptive controllers is highly dependent on their transient response. In particular, mitigating unsafe transients, such as those caused by sudden system failures, is crucial for ensuring their safe implementation. A standard approach to addressing this issue is by simply increasing the learning rates to enhance adaptation speed. However, this can

https://doi.org/10.1002/acs.4029

lead to undesirable high-frequency oscillations in the control signal. Alternatively, recent efforts have been directed toward using more sophisticated NN architectures. One method that utilizes external memory to enhance the performance of an ANN controller is proposed by Muthirayan and Khargonekar [10], which is inspired by *neural turing machines* (NTMs) [11]. However, unlike the common practice in NTMs, a feed-forward network is used instead of a *recurrent neural network* (RNN). RNNs are known for their ability to process sequential data due to their internal memory, and they have been successfully applied in various fields [12–14]. A variant of RNNs is known as *long short-term memory* (LSTM) networks. Unlike traditional RNNs, LSTM networks can learn and retain information over extended time intervals [15]. They demonstrate success in various tasks and are able to solve several problems that remain impossible to solve using any other RNN architecture [16].

In this work, we propose an ANN control architecture that relies on the employment of an LSTM network to dramatically enhance the controller's performance. In particular, we develop a synergistic approach, where the long-term learning capabilities of LSTM networks are brought to the system to aid against the deficiencies of the acting adaptive controller, significantly improving the transient response. While the proposed approach is indifferent to various adaptive control methods, we adopt the ANN given in Lewis et al. [7] as the acting adaptive controller, since it allows for a general uncertainty structure and has led to successful implementation results [17–19]. The LSTM is trained in a closed-loop setting to predict and compensate for the ANN's error in approximating the nonlinear uncertainties, which leads to an overall improved performance. LSTM's training is performed through recursive simulations on a small set of predefined uncertainties, which, despite its smallness, enables the LSTM to generalize effectively with unseen uncertainties. The stability of the proposed approach is analyzed through a Lyapunov analysis. Finally, the effectiveness of the proposed architecture is demonstrated through detailed simulation studies.

The article is tailored for a generic *multi-input multi-output* (MIMO) dynamical system, for which a modification in the ANN controller developed by Lewis et al. [7] is necessary to maintain the same stability characteristics. Therefore, a subsidiary contribution of this article is a modified robustifying term, which extends the results in the work of Lewis et al. [7] for MIMO dynamical systems.

There exist other employments of LSTM networks in the context of adaptive control in the literature. Wen et al. [20] used an LSTM network for the control of a single-axis hydraulic shaking table, where some of the LSTM weights are updated online, while the remaining weights are tuned offline. An approach where all weights of an LSTM controller are updated online is proposed by Griffis et al. [21] for Euler-Lagrange systems. Unlike the work by Wen et al. [20], the proposed architecture is developed for generic MIMO dynamics in the form of (1) rather than a specific application. Additionally, we provide a systematic method to train the LSTM network offline such that it enhances the transients of the active adaptive controller. The proposed offline training technique enables the LSTM network to generalize effectively with unseen uncertainties despite being trained on a small set of uncertainties. There are several advantages of using an offline-trained neural network integrated with an online learning method. The system can take advantage of both without increasing the complexity of the controller. Specifically, while the adaptive control theory provides stability, the LSTM network enables a fast response to the changes in the system by learning the dynamics during training. However, using only an online learning-based method leads to longer adaptation times, as the system must adapt from scratch without prior knowledge of the dynamics. It is noted that the proposed method is developed to complement the performance of an active adaptive controller, which can be implemented in (1) a simple linearly parameterized form [22], (2) a feedforward neural network form [8, 23], or (3) an LSTM network form [21], depending on the plant uncertainty and the tolerable level of complexity in the controller structure.

To summarize, the contribution of this article is a novel control architecture where an LSTM network is used to enhance the transient performance of the active adaptive control element. We provide a systematic method in which the LSTM is trained through recursive simulations to complement the adaptive controller, yielding an overall improved performance. Preliminary versions of this work are partially published in earlier conference articles [24, 25], with human-in-the-loop evaluations. This article presents the theoretical development with a rigorous stability analysis and detailed simulation scenarios. Therefore, significantly enhanced theoretical and simulation results are provided in this work compared to earlier reports.

The article is organized as follows: in Section 2, we describe the formulation of the ANN controller. In Section 3, the proposed LSTM augmentation and the training method are explained. Simulation results are given in Section 4, and a summary is given in Section 5.

2 | Problem Formulation

Consider the following plant dynamics

$$\dot{x}_p(t) = A_p x_p(t) + B_p(u(t) + f(x_p(t))) \tag{1a}$$

$$y_p(t) = C_p^T x_p(t) \tag{1b}$$

where $x_p(t) \in \mathbb{R}^{n_p}$ is the measurable state vector, $u(t) \in \mathbb{R}^m$ is the plant control input, $A_p \in \mathbb{R}^{n_p \times n_p}$ is a known system matrix, $f(x_p(t)) : \mathbb{R}^{n_p} \to \mathbb{R}^m$ is a state-dependent, possibly nonlinear, matched uncertainty, which is continuous on a known compact set, $B_p \in \mathbb{R}^{n_p \times m}$ is a known control input matrix, $C_p \in \mathbb{R}^{n_p \times s}$ is a known output matrix, and $y_p(t) \in \mathbb{R}^s$ is the plant output. Furthermore, it is assumed that the pair (A_p, B_p) is controllable.

Remark 1. The proposed method also applies for an unknown system matrix: Suppose that the plant dynamics are given by

$$\dot{x}_p(t) = A_u x_p(t) + B_p(u(t) + f_u(x_p(t))) \tag{2}$$

where $A_u \in \mathbb{R}^{n_p \times n_p}$ is an unknown system matrix and $f_u(x_p(t))$ is a state-dependent, possibly nonlinear, matched uncertainty. The dynamics (2) can be written in the form of (1) and for a known matrix A_p, by defining $f(x_p(t)) \triangleq K_u x_p(t) + f_u(x_p(t))$, given that the matching condition $A_p = A_u - B_p K_u$ is satisfied for some feedback gain $K_u \in \mathbb{R}^{m\times n_p}$. This allows us to proceed assuming A_p is known, without loss of generality.

To achieve command following of a reference input $r(t) \in \mathbb{R}^s$, a state $x_e(t) \in \mathbb{R}^s$ is defined as

$$\dot{x}_e(t) = r(t) - y_p(t) \tag{3}$$

which, when augmented with (1), yields

$$\dot{x}(t) = Ax(t) + B_m r(t) + B(u(t) + f(x_p(t))) \tag{4a}$$

$$y(t) = C^T x(t) \tag{4b}$$

where $x(t) \triangleq [x_p(t)^T, x_e(t)^T]^T \in \mathbb{R}^n$ is the augmented state, whose dimension is $n \triangleq n_p + s$, and the system matrices A, B_m, B and C are

$$A \triangleq \begin{bmatrix} A_p & 0_{n_p\times s} \\ -C_p^T & 0_{s\times s} \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{5a}$$

$$B_m \triangleq \begin{bmatrix} 0_{n_p\times s} \\ I_{s\times s} \end{bmatrix} \in \mathbb{R}^{n\times s}, \quad B \triangleq \begin{bmatrix} B_p \\ 0_{s\times m} \end{bmatrix} \in \mathbb{R}^{n\times m} \tag{5b}$$

$$C \triangleq \begin{bmatrix} C_p^T & 0_{s\times s} \end{bmatrix}^T \in \mathbb{R}^{n\times s} \tag{5c}$$

A baseline state feedback controller is designed as

$$u_{bl}(t) = -Kx(t) \tag{6}$$

where $K \in \mathbb{R}^{m\times n}$ is selected such that $A_m \triangleq A - BK$ is Hurwitz. Then, a reference model is defined as

$$\dot{x}_m(t) = A_m x_m(t) + B_m r(t) \tag{7}$$

where $x_m(t) \in \mathbb{R}^n$ is the state vector of the reference model.

By defining the state tracking error as

$$e(t) \triangleq x(t) - x_m(t) \tag{8}$$

the adaptive control objective is to make $e(t)$ converge to zero while keeping all system signals bounded, with smooth transient behavior. To achieve this, we use the control input

$$u(t) = u_{bl}(t) + u_{ad}(t) + u_{lstm}(t) \tag{9}$$

where $u_{bl}(t)$ is the baseline controller (6) that achieves reference model matching (7) in the absence of uncertainty, $u_{ad}(t)$ is an adaptive neural-network (ANN) controller designed to compensate for the matched uncertainty $f(x_p(t))$, and $u_{lstm}(t)$ is the output of an LSTM network.

The LSTM is trained to improve the transient performance of the ANN controller. Toward that end, we start by designing a suitable $u_{ad}(t)$ to compensate for the matched uncertainty $f(x_p(t))$, for which there exist several approaches subject to different assumptions on the prior knowledge about the matched uncertainty $f(x_p(t))$ [22, 26–28].

2.1 | Adaptive Neural Network Controller

The ANN control approach revolves around generating an estimate of the unknown function $f(x_p(t))$ [7, 17–19, 23, 29]. We provide the basics in this section that are needed for the description of the main contribution of this article, given in the next section. Here, we also discuss a modification to the original work [7].

Assumption 1. The unknown matched uncertainty $f(x_p(t)) : \mathbb{R}^{n_p} \to \mathbb{R}^m$, in (1), is continuous on a known compact set $S_p \triangleq \{x_p(t) : \|x_p(t)\| \le b_x\} \subset \mathbb{R}^{n_p}$, where b_x is a known positive constant.

Assumption 1 implies that $f(x_p(t))$ can be approximated by a neural network (NN) [7] as

$$f(x_p(t)) = W^T\bar{\sigma}(V^T\bar{x}_p(t)) + \varepsilon(x_p(t)) \tag{10}$$

such that

$$\|\varepsilon(x_p(t))\| \le \varepsilon_N, \qquad \forall x_p(t) \in S_p \tag{11}$$

where W and V are the NN weight matrices, concatenated with the bias vectors.

Remark 2. Assumption 1 does NOT assume that the states x_p of the system in (1) always stay in S_p, a priori. It simply states that *when* x_p is in S_p, then the premise of the assumption holds. In the stability analysis in the sequel, this is further explained in detail.

The input to the NN is given as $\bar{x}_p(t) \triangleq \begin{bmatrix} x_p(t)^T & 1\end{bmatrix}^T \in \mathbb{R}^{n_p+1}$, and the output of the hidden layer is $\bar{\sigma}(V^T\bar{x}_p(t)) \triangleq \begin{bmatrix}\sigma(V^T\bar{x}_p(t)) & 1\end{bmatrix}^T \in \mathbb{R}^{n_h+1}$. The nonlinear activation function $\sigma(.) : \mathbb{R}^{n_h} \to \mathbb{R}^{n_h}$ can be either sigmoid or tanh, where n_h is the number of hidden neurons. Furthermore, $\varepsilon(x_p(t))$ is unknown but bounded by a known constant $\varepsilon_N > 0$ in the domain of interest $S_p \subset \mathbb{R}^{n_p}$.

For convenience, we drop the overbar and write (10) as

$$f(x_p(t)) = W^T\sigma(V^T x_p(t)) + \varepsilon(x_p(t)) \tag{12}$$

where the dimensions of the weight matrices are loosely defined as $W \in \mathbb{R}^{n_h\times m}$ and $V \in \mathbb{R}^{n_p\times n_h}$.

Assumption 2. W and V are bounded by known positive constants such that $\|W\|_F \le W_M$ and $\|V\|_F \le V_M$.

The ANN controller $u_{ad}(t)$ in (9) is designed to compensate for $f(x_p(t))$, and is expressed as,

$$u_{ad}(t) = -\hat{f}(x_p(t)) + v(t) \tag{13}$$

where $\hat{f}(x_p(t))$ represents an estimate of $f(x_p(t))$, and $v(t)$ is a robustifying term, which is discussed in detail in the next subsection. Using (12), we define $\hat{f}(x_p(t))$ as

$$\hat{f}(x_p(t)) = \hat{W}^T(t)\sigma(\hat{V}^T(t)x_p(t)) \tag{14}$$

where $\hat{V}(t) \in \mathbb{R}^{n_p \times n_h}$ and $\hat{W}(t) \in \mathbb{R}^{n_h \times m}$ are estimates of the unknown ideal weights V and W, respectively. The weights are updated using the adaptive laws [7]

$$\dot{\hat{W}} = F(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x_p)e^T PB - F\kappa\|e\|\hat{W} \tag{15a}$$

$$\dot{\hat{V}} = Gx_p e^T PB\hat{W}^T\hat{\sigma}' - G\kappa\|e\|\hat{V} \tag{15b}$$

where $F \in \mathbb{R}^{n_h \times n_h}$ and $G \in \mathbb{R}^{n_p \times n_p}$ are symmetric positive definite matrices, serving as learning rates, $\kappa > 0$ is a scalar gain, and $\hat{\sigma}'$ and $\hat{\sigma}$ are defined as

$$\hat{\sigma}' \triangleq \left.\frac{d\sigma(z)}{dz}\right|_{z=\hat{V}^T(t)x_p(t)}, \quad \hat{\sigma} \triangleq \sigma(\hat{V}^T(t)x_p(t)) \tag{16}$$

Furthermore, $P \in \mathbb{R}^{n \times n}$ is the symmetric positive definite solution of the Lyapunov equation

$$A_m^T P + PA_m = -Q \tag{17}$$

for some symmetric positive definite matrix $Q \in \mathbb{R}^{n \times n}$.

2.2 | Robustifying Term

Using the baseline controller (6) and the ANN controller (13) is sufficient to solve the adaptive control problem, as demonstrated in the work of Lewis et al. [7] for nonlinear robot arm dynamics. However, for the generic multi-input multi-output (MIMO) dynamical system (1), a difficulty arises due to the control input matrix B. A subsidiary contribution of this article is a modified robustifying term $v(t)$ to address this difficulty, which achieves the same stability results given in Lewis et al. [7], but for generic MIMO systems in the form of (1).

The robustifying term, $v(t)$ is used to provide robustness against disturbances that arise from the high-order Taylor series terms. For the considered MIMO plant dynamics (1), we propose a modified robustifying term as

$$v(t) = \begin{cases} 0, & \text{if } \|B^T Pe\| = 0 \\ -\frac{B^T Pe}{\|B^T Pe\|} k_z \|e\|(\|\hat{Z}\|_F + Z_M), & \text{otherwise} \end{cases} \tag{18}$$

with

$$k_z > C_2 \tag{19}$$

where $e(t)$ is the state-tracking error given in (8), $\hat{Z}(t)$ is defined in (A19) in the Appendix, $Z_M \triangleq \sqrt{W_M^2 + V_M^2}$ (see Assumption 2), B is the input matrix in (1), and P is the solution of (17). Furthermore, $C_2 \triangleq 2C_{\sigma'}Z_M$, where $C_{\sigma'}$ is a known upper bound for the absolute value of the sigmoid or tanh activation function derivatives, that is, $||\sigma'(.)|| \leq C_{\sigma'}$, where $\sigma'(z) = d\sigma(z)/dz$.

As shown in the Appendix, the robustifying term is key to establishing the main stability result of the overall architecture provided in the sequel (see Theorem 1). The following section introduces the main focus of this article.

3 | LSTM Network Design

In this section, we precisely define the separate roles of LSTM and ANN in the closed-loop system. Figure 1 illustrates the overall control architecture.

An LSTM block is shown in Figure 2, where the input e^n_{norm} is the normalized version of the sequence e^n, which is generated from $e(t)$ defined in (8) as $e^n = e(nT)$, where $T > 0$ is the LSTM sampling time, and n denotes the current time step. The superscript $n-1$ denotes the previous time step. Furthermore, $h^n \in \mathbb{R}^{N_h}$ is the hidden state, and $c^n \in \mathbb{R}^{N_h}$ is the cell state, where N_h is the number of hidden units in the LSTM network.

In addition to the input sequence e^n_{norm}, LSTM uses its previous hidden state h^{n-1} to 1) update the cell state through the outputs of a *forget gate* out^n_f, an *input gate* out^n_i, and a *cell candidate* $\bar{c}^n$ as

$$c^n = out^n_f \odot c^{n-1} + out^n_i \odot \bar{c}^n \tag{20}$$

and 2) update the hidden state through the updated cell state and the output of an *output gate* out^n_o as

$$h^n = out^n_o \odot \sigma_c(c^n) \tag{21}$$

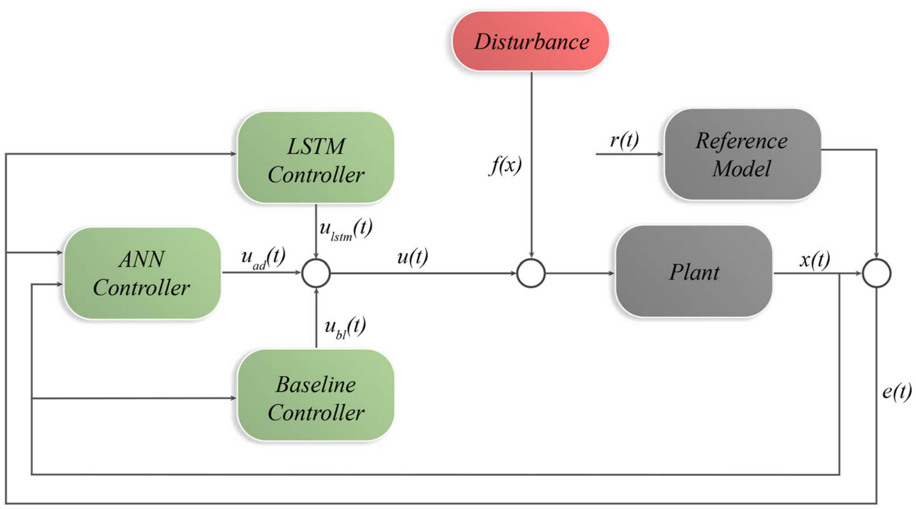

FIGURE 1 | Block diagram of the proposed control architecture.

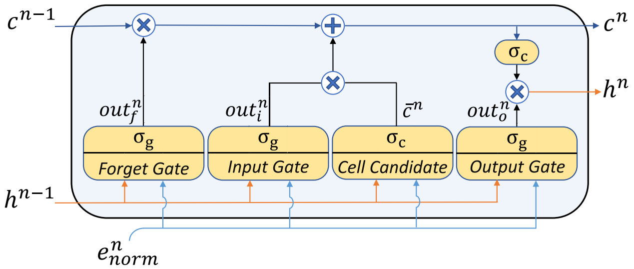

FIGURE 2 | Detailed sketch of an LSTM block.

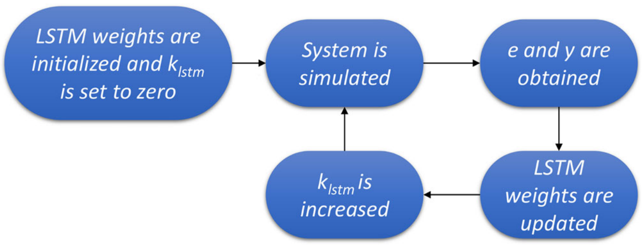

FIGURE 3 | Flow chart of the training process.

These gate operations are applied using the activation functions $\sigma_g(.)$ and $\sigma_c(.)$, which denote sigmoid and tanh, respectively, and suitable weights $\{R_f, W_f, b_f\}$, $\{R_c, W_c, b_c\}$, $\{R_i, W_i, b_i\}$, and $\{R_o, W_o, b_o\}$, as shown below.

(a) Forget Gate: The forget gate determines the relevant information that the cell state c^n should retain from its previous time step, as indicated by the first term in (20). The output is calculated as

$$out_f^n = \sigma_g(W_f e^n_{norm} + R_f h^{n-1} + b_f) \tag{22}$$

(b) Input Gate and Cell Candidate: The flow of information into the cell state (see the second term in (20)) is given by a cell candidate $\bar{c}^n$ as

$$\bar{c}^n = \sigma_c(W_c e^n_{norm} + R_c h^{n-1} + b_c) \tag{23}$$

and is controlled by the output of the input gate

$$out_i^n = \sigma_g(W_i e^n_{norm} + R_i h^{n-1} + b_i) \tag{24}$$

(c) Output Gate: The output gate determines how much of the cell state is going to be used as the hidden state for the next time step (see 21), the output of which is defined as

$$out_o^n = \sigma_g(W_o e^n_{norm} + R_o h^{n-1} + b_o) \tag{25}$$

Finally, to formulate the output sequence of the LSTM network, the hidden state is fed to a fully connected layer yielding an $m-$dimensional output that matches the dimension of the control input as

$$u^n_{lstm} = W_{fc} h^n + b_{fc} \tag{26}$$

where W_{fc} and b_{fc} are the parameters of the fully connected layer. Then, the control input $u_{lstm}(t)$ is constructed as

$$u_{lstm}(t) = u^n_{lstm}, \quad \text{for } nT \le t < (n+1)T, \quad n = 0, 1, 2, \dots \tag{27}$$

3.1 | Training Method

The LSTM network is trained to predict and compensate for the estimation error of the ANN controller defined by the deviation of the ANN controller input, $\hat{f}(x_p(t))$ (see (13)), from the actual uncertainty, $f(x_p(t))$, which is given as

$$\tilde{f}(x_p(t)) \triangleq f(x_p(t)) - \hat{f}(x_p(t)) \tag{28}$$

The "truth" for the LSTM, or the desired signal for the LSTM to produce, is selected as

$$y(t) = -\tilde{f}(x_p(t)) \tag{29}$$

Therefore, LSTM provides a signal to the system that is an estimate of this truth, expressed as

$$u_{lstm}(t) = \hat{y}(t) \tag{30}$$

During training, an uncertainty $f_{train}(x_p(t))$ is introduced to the closed-loop system, and LSTM is expected to learn $\tilde{f}_{train}(x_p(t)) = f_{train}(x_p(t)) - \hat{f}(x_p(t))$, where $-\hat{f}(x_p(t))$ is an estimate generated by the ANN to cancel the uncertainty (see (13)). Apart from $\tilde{f}_{train}(x_p(t))$, LSTM uses the state-tracking error $e(t)$, defined in (8), as its input (see Figure 1). Every iteration of training affects the sequence that the LSTM is trained on. This dynamic nature of training helps the LSTM generalize to a set of functions that are not used in the training set.

Figure 3. explains the training process of the LSTM network. First, LSTM weights are initialized, and training data $(e(t), y(t))$ are generated through numerical simulations. After LSTM weights are updated, the training data are recollected from simulations using the updated weights. This is necessary since LSTM is used in a close-loop setting: updating LSTM weights affects its output, which in turn alters the evolution of its input. To avoid an untrained LSTM network output from negatively affecting the system's response at the beginning of training, LSTM output u_{lstm} is scaled by a gain "k_{lstm}". k_{lstm} is initiated from 0 and approaches 1 as the number of training iterations increases, then stays at 1 for the rest of the training.

3.2 | Normalization

Before training, the components of the LSTM network input e^n (see Figure 2) are normalized. For collecting the normalization parameters only, the simulation is run with f_{train} (see explanations after (30)) for z number of times. A gain is uniformly sampled from [0, 1] in every simulation run to scale f_{train}. Within the collected data, the minimum and the maximum values of each component of the error vector, e_i^{min} and e_i^{max}, are utilized to obtain the normalized input parameters as

$$e^n_{norm_i} = (e^n_i - e^{min}_i)/(e^{max}_i - e^{min}_i) \tag{31}$$

where "i" is used to refer to the vector's i^{th} component.

3.3 | Stability Analysis

The next theorem outlines the overall architecture's stability properties, with the proof deferred to the Appendix.

Theorem 1. *Consider the uncertain plant dynamics* (1), *subject to Assumptions 1 and 2, and the reference model (7). Let the control input be defined by (9), which consists of the baseline controller (6), the ANN controller defined by (13), (14) and* (15), *the robustifying term given in (18) and (19), and the LSTM controller given in (22–30). Furthermore, let the Lyapunov matrix Q in (17) be selected such that*

$$\lambda_{min}(Q) > \frac{\kappa q_1^2 + 4q_0}{2(b_x - C_m)} \tag{32}$$

and let $b_x > C_m$, where $q_0 \triangleq \|B^T P\|_F (C_0 + \bar{u}_{lstm})$, $q_1 \triangleq C_1 \|B^T P\|_F/\kappa + Z_M$, κ is the scalar gain used in (15), *b_x is defined in Assumption 1, and C_0, C_1, C_m, and $\bar{u}_{lstm}$ are defined in the Appendix. Then, given that $x_p(0) \in S_p$ (see Assumption 1), the solution $(e(t), \tilde{W}(t), \tilde{V}(t))$ is uniformly ultimately bounded (UUB)* [30] *and converges to a predefined compact set, where $\tilde{V}(t) \triangleq V - \hat{V}(t)$ and $\tilde{W}(t) \triangleq W - \hat{W}(t)$.*

It should be noted that the condition in (32) is not guaranteed to be satisfied by simply increasing Q. This is because, according to the Lyapunov equation (17), increasing Q leads to an increased P, which implies higher values q_0 and q_1. One solution is using the pair $(\gamma P_0, \gamma Q_0 = \gamma I)$, which satisfies (17) for any scalar $\gamma > 0$. A suitable γ can then be found to satisfy (32): Substituting this pair to (32), yields

$$b\gamma^2 + (c - a)\gamma + d < 0 \tag{33}$$

where $b \triangleq C_1^2 \|B^T P_0\|_F^2/\kappa$, $c \triangleq 2C_1 \|B^T P_0\|_F Z_M + 4\|B^T P_0\|_F (C_0 + \bar{u}_{lstm})$, $a \triangleq 2(b_x - C_m)$, and $d \triangleq \kappa Z_M^2$. The condition (33) is in a quadratic form, for which there exists a positive real $\gamma > 0$ if and only if $a \geq \sqrt{4bd} + c$.

Remark 3. The proposed LSTM augmentation scheme can also be applied to a standard adaptive controller without using an NN, where the matched uncertainty $f(x_p(t))$ in (1) is assumed to be linearly parameterized. For completeness, we provide the details of the LSTM augmentation with a standard adaptive control architecture and a stability result (Theorem 2) in Appendix C.

4 | Simulations

In this section, the effectiveness of the proposed framework is elucidated for the control of the short-period longitudinal flight dynamics, which, for a B-747 aircraft flying at a speed of 274 m/s and a 6000 m altitude, are given as [10]

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -0.32 & 0.86 \\ -0.93 & -0.43 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} -0.02 \\ -1.16 \end{bmatrix} (u + f(x_p)) \tag{34}$$

where α is the angle of attack (rad), q is the pitch rate (rad/s), u is the elevator deflection (rad), $x_p \triangleq [\alpha,\ q]^T$, and $f(x_p)$ is a state dependent matched uncertainty. The control objective is to achieve pitch rate tracking. Furthermore, elevator magnitude and rate saturation limits are set as $+17/-23$ (deg) and $+37/-37$ (deg/s) [31]. To put the dynamics in the form of (4), we define the augmented state as $x(t) \triangleq [x_1(t),\ x_2(t),\ x_3(t)]^T$, where $x_1 = \alpha$ (rad), $x_2 = q$ (rad/s) and x_3 is the error-integral state (3).

The input to the LSTM network is $in^n_{lstm} = \left[e^n_{norm_1}\ \ e^n_{norm_2}\ \ e^n_{norm_3}\right]^T$, where $e^n_{norm_1}$, $e^n_{norm_2}$, and $e^n_{norm_3}$ are the components of the normalized version of the state tracking error vector (31). The baseline controller (6) is an LQR controller with cost matrices $Q_{LQR} = I$ and $R_{LQR} = 1$. The LSTM network is trained in the presence of an ANN controller (see Section 2.1) with a hidden layer of $n_h = 4$ neurons. The learning rates in (15a) and (15b) are set as $F = G = 10$, and the Lyapunov matrix Q in (17) is set as $Q = I$. The outer weights $\hat{W}$ and bias $\hat{b}_w$ are initialized from zero. The inner weights $\hat{V}$ and biases $\hat{b}_v$ are initialized randomly between 0 and 1. The LSTM network contains one hidden layer with $N_h = 128$ neurons. The number of neurons in the input layer is 3 due to the number of state-tracking error components given as the input to the network. The LSTM weights are initialized using Xavier Initialization [32]. The network is trained with stochastic gradient descent and uses the Adam optimizer with the gradient clipping method with a threshold value of 1. The simulation step time is set to 0.01 s, and the minibatch size is taken as 1. To obtain the normalization constants given in (31), the system is simulated 1000 times ($z = 1000$, see Section 3.2). The loss function is chosen as $L(y, \hat{y}) = \frac{1}{N}\sum_{i=0}^{N}(y - \hat{y})^2$, where y and $\hat{y}$ are defined in (29) and (30), and N is the number of data in the data set.

Below, we investigate two scenarios where we show simulation results in the presence of (1) a generic nonlinear uncertainty $f(x_p(t))$ that satisfies the conditions given in Assumption 1, and 2 an uncertainty representation that has the structure given in (12).

4.1 | Scenario 1

The nonlinear uncertainty, f_{train} (see explanations after 30), that the LSTM network (26) is trained on, and the test uncertainty, f_{test}, that the proposed control framework is tested on are defined respectively as

$$f_{train}(x_p) = \begin{cases} 0.1x_1x_2 & \text{if } 0 \leq t < 5 \\ \exp(x_2) & \text{if } 5 \leq t < 10 \\ 2x_1x_2 & \text{if } 10 \leq t < 15 \\ -0.1\cos(x_1) & \text{if } 15 \leq t < 20 \\ 0.5(x_1 - x_2) & \text{if } 20 \leq t < 25 \\ 0.1x_1x_2 & \text{if } 25 \leq t < 30 \\ -x_1x_2(\sin(5x_1x_2) + 5\sin(x_2)) & \text{if } 30 \leq t < 35 \\ x_1x_2(3\sin(2x_1x_2) + 2x_1) & \text{if } 35 \leq t < 45 \\ -x_1x_2(2(\tan(2x_1x_2) + x_2^2) & \text{if } 45 \leq t < 55 \\ x_1x_2(x_1 + x_2) & \text{if } 55 \leq t \leq 60 \end{cases} \tag{35}$$

$$f_{test}(x_p) = \begin{cases} 0 & \text{if } 0 \le t < 2 \\ -0.1\exp(x_1x_2) & \text{if } 2 \le t < 8 \\ 0.5x_2^2 & \text{if } 8 \le t < 12 \\ 0.05\exp(x_1 + 2x_2) & \text{if } 12 \le t < 20 \\ -0.1\sin(0.05x_1x_2) & \text{if } 20 \le t < 28 \\ 0.1(x_1^2 + x_2^3) & \text{if } 28 \le t < 34 \\ -0.1|\cos(x_2)| & \text{if } 34 \le t < 40 \\ -0.2\sin(x_1x_2) & \text{if } 40 \le t < 49 \\ 0.1(x_1x_2)^2 & \text{if } 49 \le t < 53 \\ 0.5x_2 & \text{if } 53 \le t \le 60 \end{cases} \quad (36)$$

where (36) is chosen to have different types of subfunctions with different time intervals, compared to (35). To cover both low- and high-valued uncertainties, f_{train} is scaled by a parameter that took the alternating values of 0.2 and 2 during training. Furthermore, the robustifying term $v(t)$ in (13) is taken to be zero [7, 10]. There are mainly two reasons for this. The first reason is that satisfactory control performance is obtained without the help of this term, and therefore, omitting the term makes the implementation simpler. Secondly, the implementation requires the calculation of Z_M in (18), which requires the upper bounds on the ideal NN weights. Although this can be achieved, it may also complicate the implementation. One way to achieve this, for example, is by training an additional NN to approximate probable uncertainties and using upper bounds that bound the upper limits of the trained NN weights.

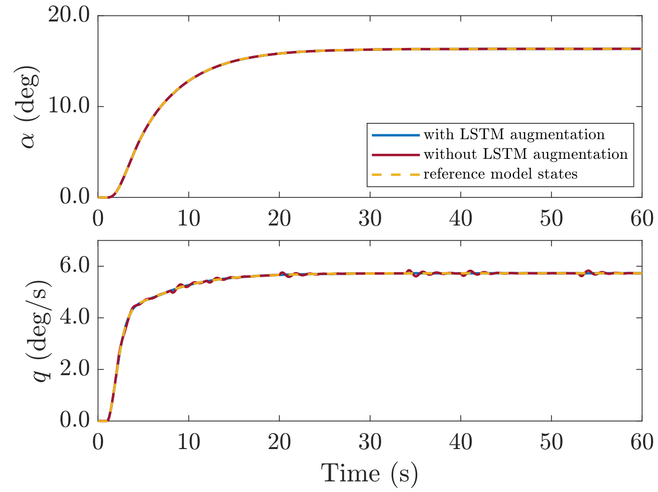

FIGURE 4 | Tracking performances with and without LSTM augmentation, in the presence of small uncertainty.

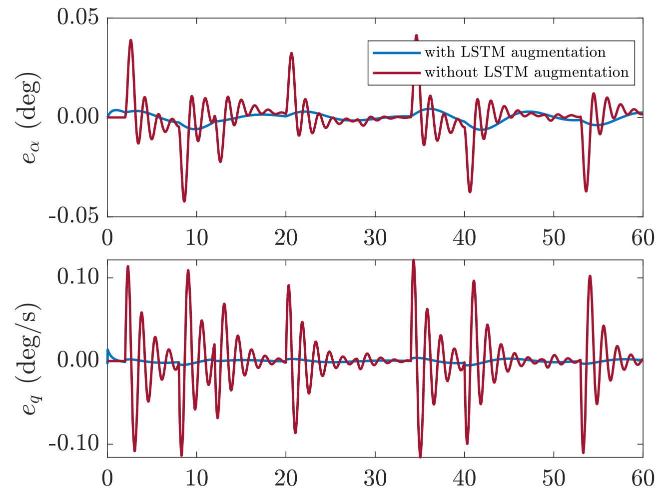

FIGURE 5 | Tracking errors with and without LSTM augmentation, in the presence of small uncertainty.

4.1.1 | Controller Performance With Small Uncertainty

In this section, the effect of the proposed LSTM augmentation is examined using f_{test} but with a scaling factor of 0.1.

Figures 4 and 5 show the state-tracking and tracking-error curves, respectively. While the error plots demonstrate that LSTM augmentation dramatically reduces both the error magnitudes and the transient oscillations, the error values are small enough to be ignored compared to the absolute values of the states. Therefore, one can conclude that in this scenario, LSTM-augmented and non-augmented cases show similar performances. In Figure 6, the control inputs are presented. This figure shows that although LSTM augmentation does not affect the magnitude of the total control input in a meaningful manner, it provides damping to the oscillations by providing small but agile compensation for uncertainties.

4.1.2 | Controller Performance With Large Uncertainty

In this section, the effect of the proposed LSTM augmentation is examined using f_{test} in (36), without a scaling factor. Figures 7 and 8 demonstrate that LSTM augmentation substantially improves the transient response of the system, especially in pitch rate tracking, which is the output of interest. It is noted that for this case, the same ANN controller and the LSTM network are used as in the small uncertainty case. Individual control inputs are shown in Figure 9. In this case, unlike the low-uncertainty case, the LSTM augmentation makes the total control signal observably more responsive, which is the main reason why excessive oscillations are prevented.

4.1.3 | Addressing Saturation

It is seen, for example, at the zoomed version of the top subfigure of Figure 9, that the total control input is experiencing rate saturation at certain times. Although this did not cause any severe detrimental effects, it is advisable to address rate-saturation to create safe and reliable control systems. To handle the saturation issue, we modified LSTM training by (a) introducing the saturation limits to the plant dynamics during training and (b) informing the LSTM network whenever the control signal rate saturates. The latter is achieved by providing additional input to the LSTM network as

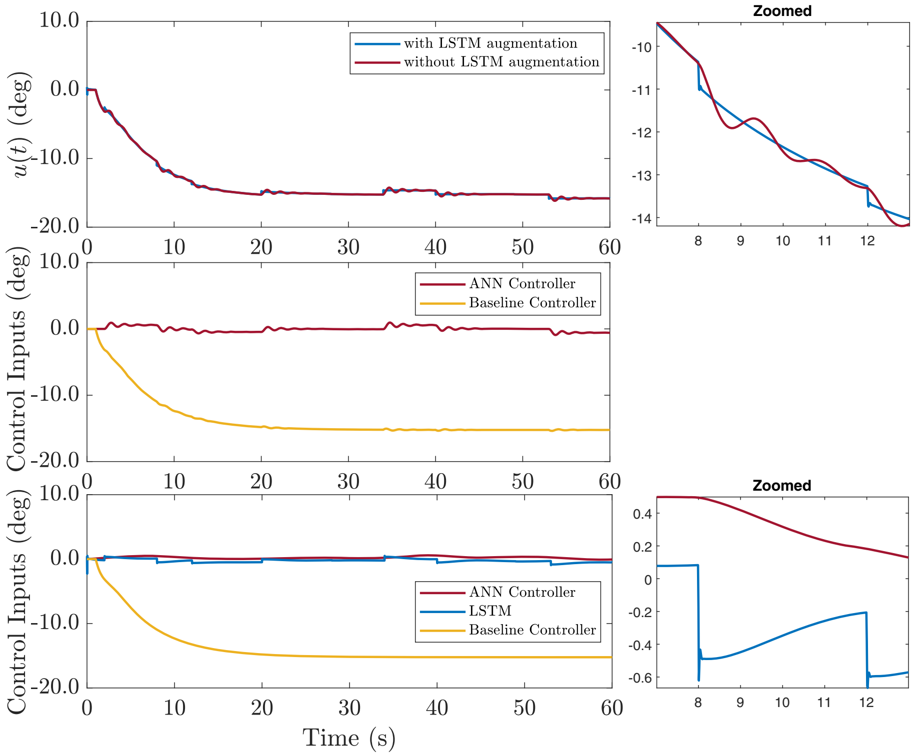

FIGURE 6 | Control inputs in the presence of small uncertainty. Top: total control input. Middle: the contributions of individual control inputs without the LSTM. Bottom: the contributions of individual control inputs with the LSTM.

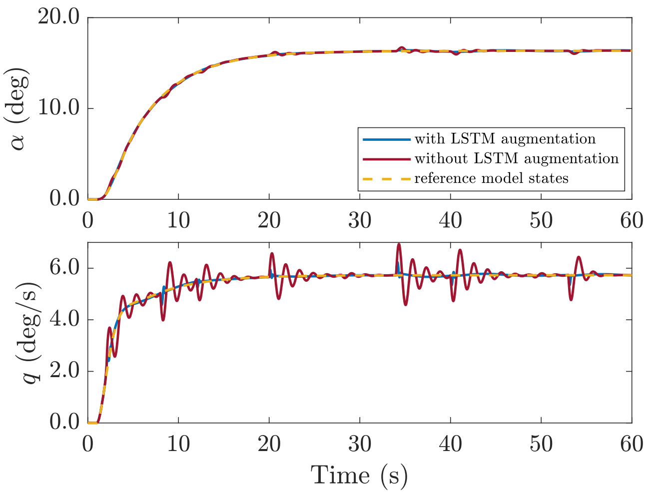

FIGURE 7 | Tracking performances with and without LSTM augmentation, in the presence of large uncertainty.

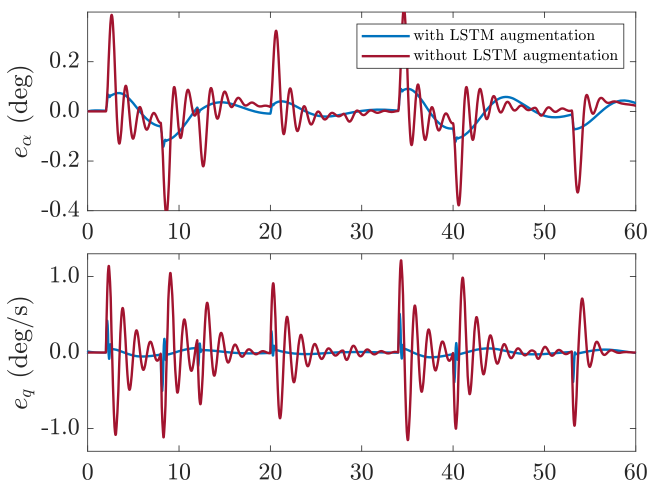

FIGURE 8 | Tracking errors with and without LSTM augmentation, in the presence of large uncertainty.

$$u_r(t) = \begin{cases} 0.1, & \text{if rate saturation is positive,} \\ -0.1, & \text{if rate saturation is negative,} \\ 0, & \text{otherwise} \end{cases} \tag{37}$$

The number of neurons in the input layer of the LSTM network is increased to 4 since the modified input to the LSTM network becomes

$$in_{lstm}(t) = [e_{norm_1}(t)\ e_{norm_2}(t)\ e_{norm_3}(t)\ u_r(t)]^T \tag{38}$$

Figure 10 shows the LSTM network contribution and the total control inputs, with and without rate-limit information during training. The LSTM network uses the rate-limit information to provide a smoother compensation. Figs. 11 and 12 show the tracking error and tracking curves, respectively, when the LSTM network is trained using the rate-limit information. It is seen that the performance of the proposed controller remains similar for the case when LSTM is not trained using this information, although the pitch rate shows small deviations at the instances of uncertainty switches.

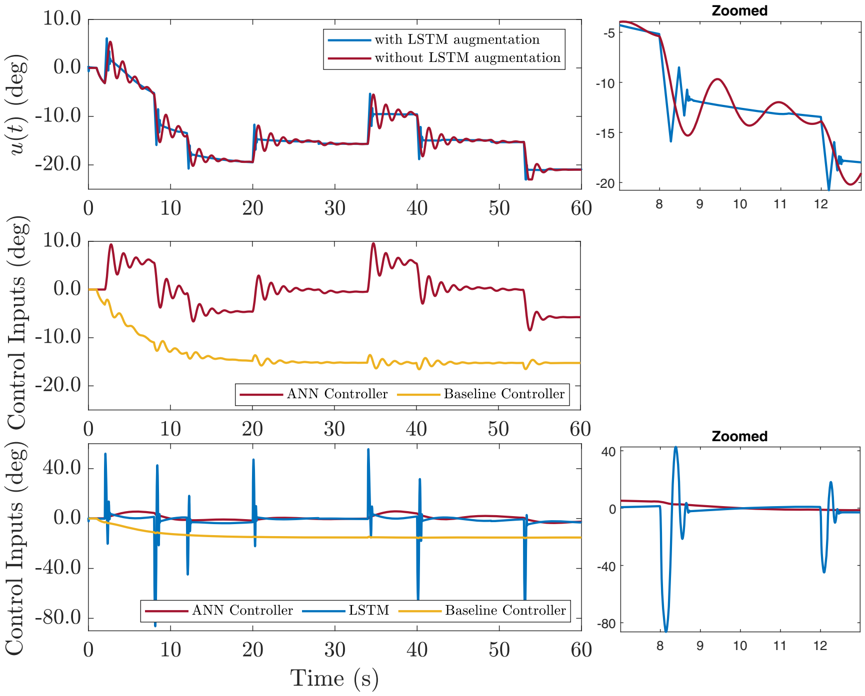

FIGURE 9 | Control inputs in the presence of high uncertainty. Top: total control input. Middle: the contributions of individual control inputs without the LSTM. Bottom: the contributions of individual control inputs with the LSTM.

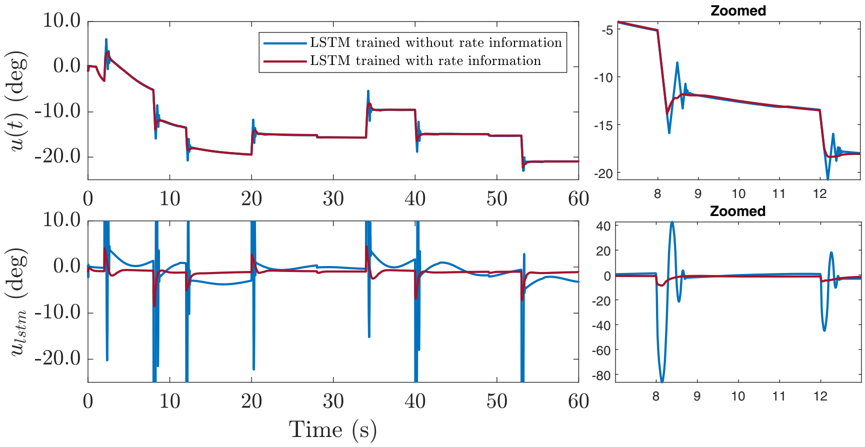

FIGURE 10 | Control signals with and without rate-limit-informed LSTM. Top: total control input. Bottom: LSTM network contribution to the total control input.

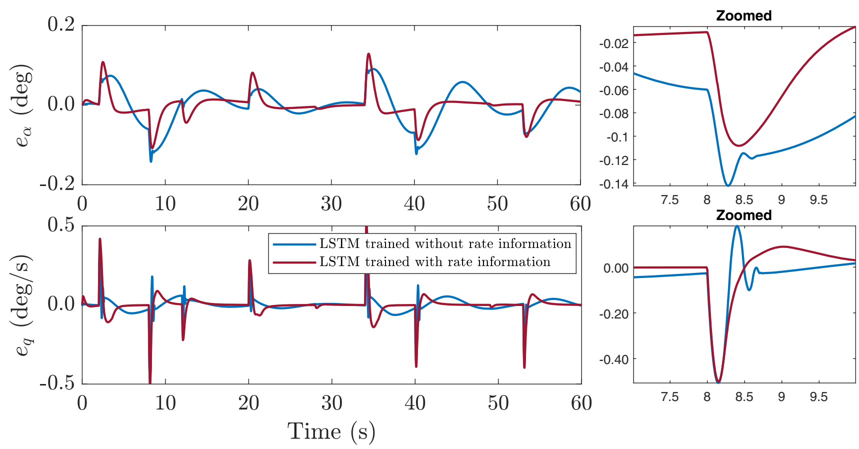

FIGURE 11 | Tracking error with LSTM trained without rate information and rate-limit-informed LSTM.

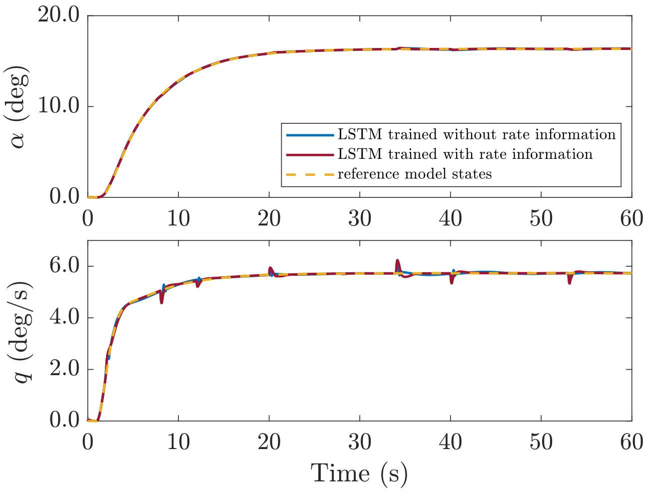

FIGURE 12 | Tracking performances with LSTM trained without rate information and rate-limit-informed LSTM.

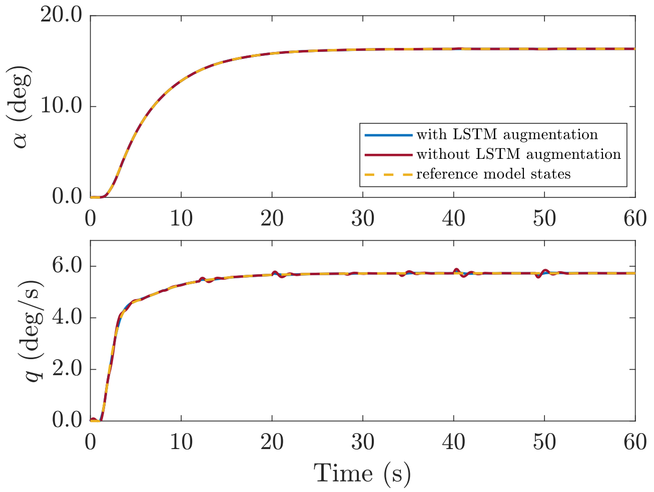

FIGURE 13 | Tracking performances with and without LSTM augmentation, in the presence of small uncertainty.

4.2 | Scenario 2

In this section, the uncertainty $f(x_p(t))$ in (1) is introduced in the form of (12), where the weight matrices, W and V, are generated in a uniformly random manner within the range of $[-0.05, 0.05]$ for both training and test functions at specified time intervals as in (35) and (36). The NN reconstruction error vector $\epsilon(x_p(t))$ is defined for the training function and testing function respectively as,

$$\epsilon(x_p(t))_{train} = \begin{cases} 0.1cos(\|x_p\|)^2 & \text{if } 0 \le t < 5 \\ -0.1cos(\|x_p\|) & \text{if } 5 \le t < 10 \\ 0.1sin(\|x_p\|) & \text{if } 10 \le t < 15 \\ 0.1sin(\|x_p\|)^2 & \text{if } 15 \le t < 20 \\ 0.1sin(\|x_p\|) & \text{if } 20 \le t < 25 \\ 0.1cos(\|x_p\|) & \text{if } 25 \le t < 30 \\ 0.1sin(\|x_p\|)^2 & \text{if } 30 \le t < 35 \\ -0.1cos(\|x_p\|) & \text{if } 35 \le t < 45 \\ 0.1sin(\|x_p\|) & \text{if } 45 \le t < 55 \\ -0.1sin(\|x_p\|)^3 & \text{if } 55 \le t \le 60 \end{cases} \tag{39}$$

$$\epsilon(x_p(t))_{test} = \begin{cases} 0.1sin(\|x_p\|)^2 & \text{if } 0 \le t < 2 \\ 0.1cos(\|x_p\|) & \text{if } 2 \le t < 8 \\ 0.1cos(\|x_p\|)^2 & \text{if } 8 \le t < 12 \\ 0.1sin(\|x_p\|)^2 & \text{if } 12 \le t < 20 \\ 0.1sin(\|x_p\|) & \text{if } 20 \le t < 28 \\ -0.1sin(\|x_p\|)^2 & \text{if } 28 \le t < 34 \\ 0.1cos(\|x_p\|) & \text{if } 34 \le t < 40 \\ -0.1cos(\|x_p\|)^3 & \text{if } 40 \le t < 49 \\ 0.1sin(\|x_p\|) & \text{if } 49 \le t < 53 \\ 0.1cos(\|x_p\|) & \text{if } 53 \le t \le 60 \end{cases} \tag{40}$$

The reconstruction error, which is represented by (39) and (40), is the second term of the uncertainty representation in (12). This term is dramatically smaller compared to the first term in (12). Therefore, the requirement that the test and the training uncertainties must be different is ensured mainly by selecting the weight matrices V and W differently, which is satisfied by sampling them uniformly randomly within a given range. This is why it is sufficient to assign different time intervals to similar terms in (39) and (40).

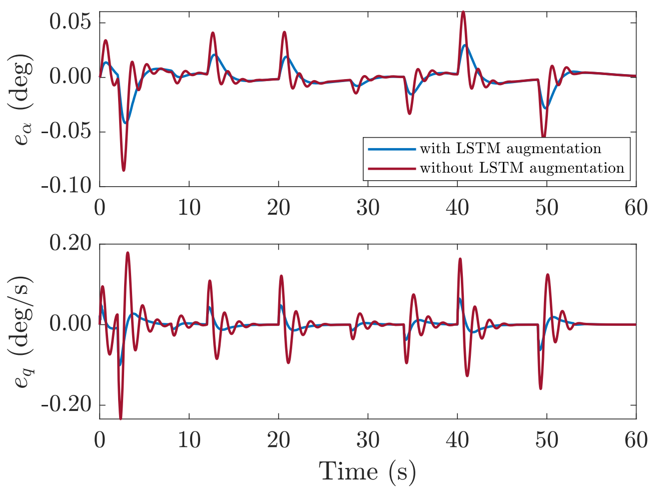

FIGURE 14 | Tracking error with and without LSTM augmentation, in the presence of small uncertainty.

As discussed in the previous subsection, omitting the implementation of the robustifying term in (13) simplifies the process. However, in the case of an explicit uncertainty representation (12), the implementation of the robustifying term becomes straightforward, since no additional NN training is required. For (33), we use weight-clipping at the range $[\pm 0.4]$ to obtain $\bar{u}_{lstm}$, and choose $C_m = 0.4$ (see the Appendix). Then, $\gamma = 1$, $\kappa = 1$, and $b_x = 84$ satisfies (33).

4.2.1 | Controller Performance With Small Uncertainty

This section analyzes the impact of the proposed LSTM augmentation in the presence of small-valued uncertainties, obtained by scaling the test uncertainty by 0.1.

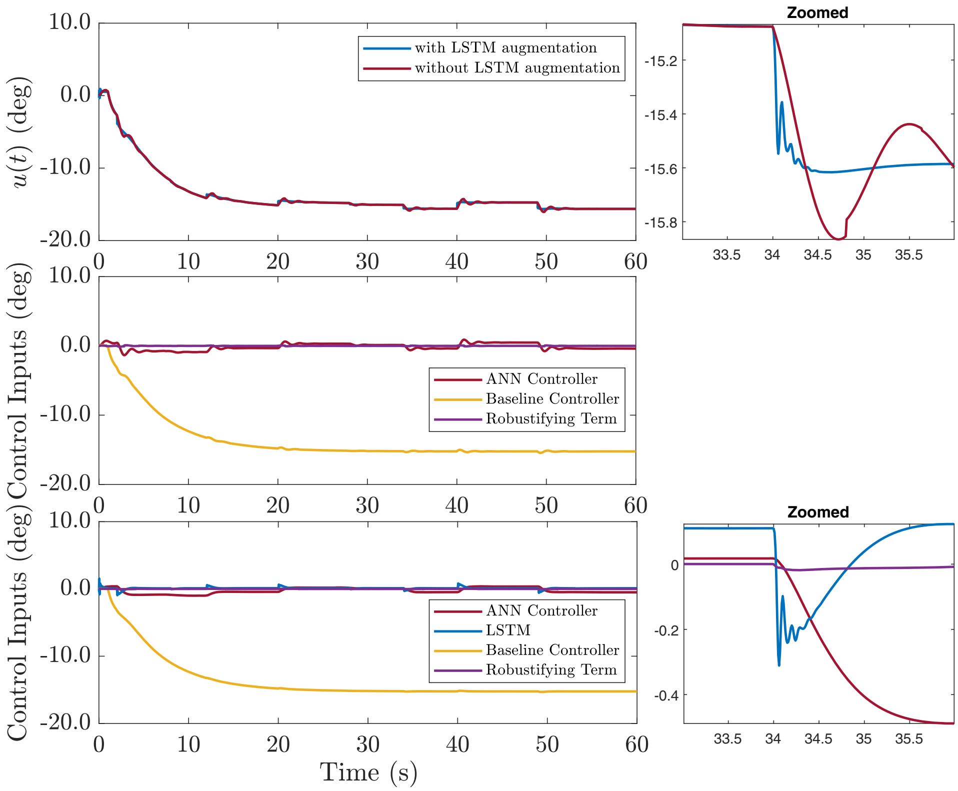

FIGURE 15 | Control inputs in the presence of small uncertainty. Top: total control input. Middle: the contributions of individual control inputs without the LSTM. Bottom: the contributions of individual control inputs with the LSTM.

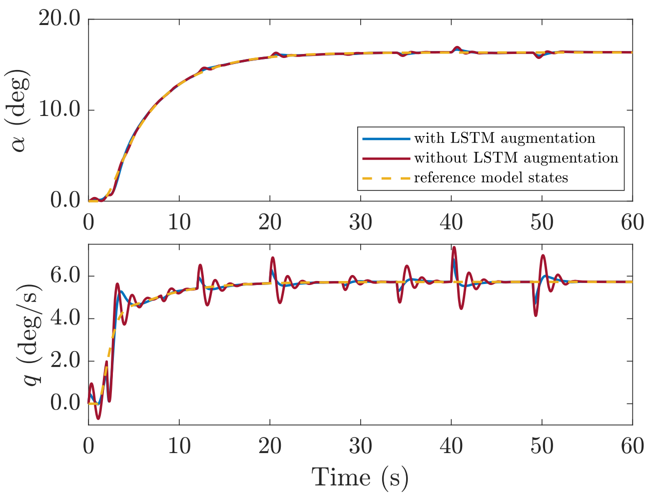

FIGURE 16 | Tracking performances with and without LSTM augmentation, in the presence of high uncertainty.

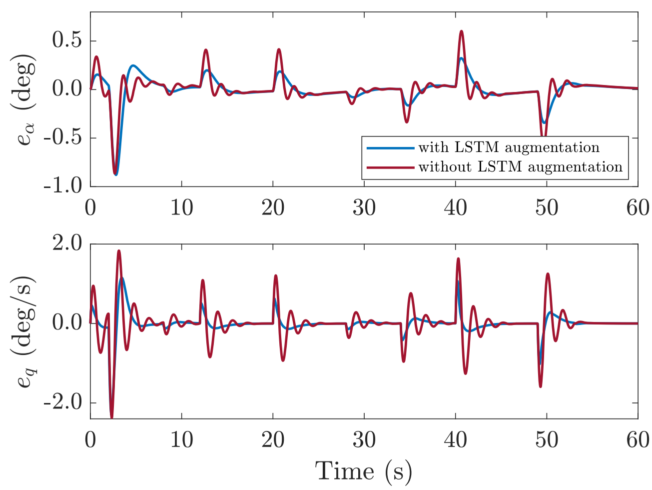

FIGURE 17 | Tracking error with and without LSTM augmentation, in the presence of high uncertainty.

The tracking and tracking-error curves are given in Figures 13 and 14, respectively. The results reveal that the LSTM effectively reduces the tracking error and minimizes oscillations. However, these benefits are not substantial, which can be seen from observing the tracking curve, Figure 13, where the ANN controller with the robustifying term is shown to perform in a satisfactory manner. Similar to the previous scenario, it is seen from Figure 15 that the LSTM augmentation contributes to the system by providing more agile control signals and therefore making the system more responsive.

4.2.2 | Controller Performance With Large Uncertainty

This section aims to investigate the impact of the proposed LSTM augmentation in the presence of high-valued uncertainties, which are introduced by directly using the test uncertainty.

Unlike the small uncertainty case, the results shown in Figures 16 and 17 indicate a considerable improvement in the transient behavior of the system. Additionally, a close examination of the individual control inputs in Figure 18 shows that the ability of

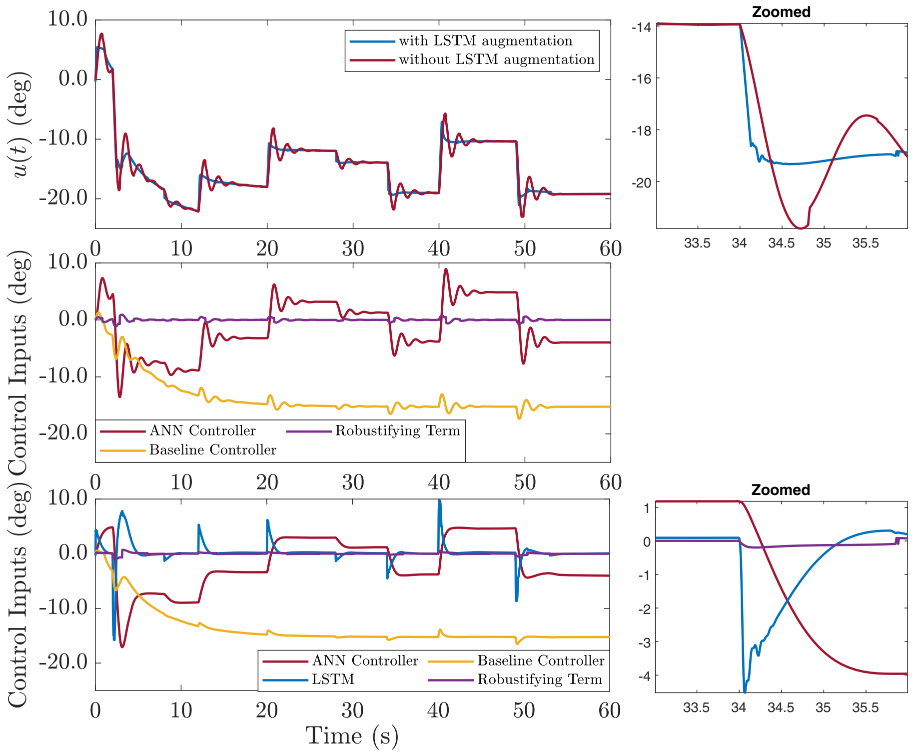

FIGURE 18 | Control inputs in the presence of high uncertainty. Top: total control input. Middle: the contributions of individual control inputs without the LSTM. Bottom: the contributions of individual control inputs with the LSTM.

the overall system to quickly respond to rapidly changing uncertainties originates from the contribution of the LSTM network. Further analysis of Figure 18 indicates that the LSTM augmentation reduces the oscillatory behavior of the ANN controller and the robustifying term.

5 | Summary

In this work, an LSTM augmented ANN control structure is proposed to improve the transient response of adaptive closed-loop control systems. We demonstrate that, thanks to its time-series prediction capabilities, LSTM helps the ANN controller compensate for the uncertainties in a more agile fashion, resulting in dramatically improved tracking performance.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The authors have nothing to report.

References

1. S. N. Kumpati and P. Kannan, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks* 1, no. 1 (1990): 4–27.

2. R. M. Sanner and J. J. E. Slotine, "Gaussian Networks for Direct Adaptive Control," in *1991 American Control Conference* (IEEE, 1991), 2153–2159.

3. G. A. Rovithakis and M. A. Christodoulou, "Adaptive Control of Unknown Plants Using Dynamical Neural Networks," *IEEE Transactions on Systems, Man, and Cybernetics* 24, no. 3 (1994): 400–412.

4. M. M. Polycarpou, "Stable Adaptive Neural Control Scheme for Nonlinear Systems," *IEEE Transactions on Automatic Control* 41, no. 3 (1996): 447–451.

5. B. Ren, S. S. Ge, K. P. Tee, and T. H. Lee, "Adaptive Neural Control for Output Feedback Nonlinear Systems Using a Barrier Lyapunov Function," *IEEE Transactions on Neural Networks* 21, no. 8 (2010): 1339–1345.

6. R. Sun, M. L. Greene, D. M. Le, Z. I. Bell, G. Chowdhary, and W. E. Dixon, "Lyapunov-Based Real-Time and Iterative Adjustment of Deep Neural Networks," *IEEE Control Systems Letters* 6 (2021): 193–198.

7. F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer Neural-Net Robot Controller With Guaranteed Tracking Performance," *IEEE Transactions on Neural Networks* 7, no. 2 (1996): 388–399.

8. A. Patkar and A. M. Annaswamy, "An Adaptive Controller for a Class of Nonlinear Plants Based on Neural Networks and Convex Parameterization," in *2020 59th IEEE Conference on Decision and Control (CDC)* (IEEE, 2020), 126–131.

9. O. S. Patil, D. M. Le, M. L. Greene, and W. E. Dixon, "Lyapunov-Derived Control and Adaptive Update Laws for Inner and Outer Layer Weights of a Deep Neural Network," *IEEE Control Systems Letters* 6 (2021): 1855–1860.

10. D. Muthirayan and P. Khargonekar, "Memory Augmented Neural Network Adaptive Controllers: Performance and Stability," *IEEE Transactions on Automatic Control* 68, no. 2 (2023): 825–838, https://doi.org/10.1109/TAC.2022.3144382.

11. A. Graves, G. Wayne, and I. Danihelka, "Neural Turing Machines," (2014), *arXiv preprint arXiv:1410.5401*.

12. F. Visin, M. Ciccone, A. Romero, et al., "Reseg: A Recurrent Neural Network-Based Model for Semantic Segmentation," in *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition Workshops (IEEE, 2016), 41–48.

13. G. Mesnil, Y. Dauphin, K. Yao, et al., "Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23, no. 3 (2014): 530–539.

14. N. Mohajerin and S. L. Waslander, "Multistep Prediction of Dynamic Systems With Recurrent Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems* 30, no. 11 (2019): 3370–3383.

15. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation* 9, no. 8 (1997): 1735–1780.

16. A. Graves, "Long Short-Term Memory," in *Supervised Sequence Labelling With Recurrent Neural Networks* (Springer Berlin Heidelberg, 2012), 37–45.

17. W. Żylski and P. Gierlak, "Verification of Multilayer Neural-Net Controller in Manipulator Tracking Control," in *Solid State Phenomena*, vol. 164 (Trans Tech Publ, 2010), 99–104.

18. Z. Yao, J. Yao, and W. Sun, "Adaptive RISE Control of Hydraulic Systems With Multilayer Neural-Networks," *IEEE Transactions on Industrial Electronics* 66, no. 11 (2018): 8638–8647.

19. G. L. Burrola, M. Á. L. Leal, and J. S. Mijares, "Adaptive Neural Control for a Two Wheeled Inverted Pendulum System," in *2022 XXIV Robotics Mexican Congress (COMRob)* (IEEE, 2022), 48–53.

20. J. Wen, C. Zhao, and Z. Shi, "LSTM-Based Adaptive Robust Nonlinear Controller Design of a Single-Axis Hydraulic Shaking Table," *IET Control Theory & Applications* 17, no. 7 (2022): 825–836, https://doi.org/10.1049/cth2.12410.

21. E. J. Griffis, O. S. Patil, Z. I. Bell, and W. E. Dixon, "Lyapunov-Based Long Short-Term Memory (Lb-LSTM) Neural Network-Based Control," *IEEE Control Systems Letters* 7 (2023): 2976–2981.

22. E. Lavretsky and K. Wise, *Robust and Adaptive Control: With Aerospace Applications* (Springer, 2013).

23. F. Lewis, S. Jagannathan, and A. Yesildirak, *Neural Network Control of Robot Manipulators and Non-Linear Systems* (CRC Press, 1998).

24. E. Inanc, Y. Gurses, A. Habboush, and Y. Yildiz, "Long Short-Term Memory for Improved Transients in Neural Network Adaptive Control," in *2023 American Control Conference (ACC), San Diego, CA, USA* (IEEE, 2023), 3614–3619.

25. M. Y. Uzun, E. Inanc, A. Habboush, and Y. Yildiz, "Human-In-The-Loop Performance Evaluation of an Adaptive Control Framework With Long Short-Term Memory Augmentation," in *2023 European Control Conference (ECC), Bucharest, Romania* (IEEE, 2023), 1–6.

26. A. M. Annaswamy and A. L. Fradkov, "A Historical Perspective of Adaptive Control and Learning," *Annual Reviews in Control* 52 (2021): 18–41.

27. T. Yucelen, "Model Reference Adaptive Control," in *Encyclopedia of Electrical and Electronics Engineering* ed. J. G. Webster (Wiley, 2019), 1–13.

28. B. C. Gruenwald, T. Yucelen, G. De La Torre, and J. A. Muse, "Adaptive Control for Uncertain Dynamical Systems With Nonlinear Reference Systems," *International Journal of Systems Science* 51, no. 4 (2020): 687–703.

29. S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*, vol. 13 (Springer Science & Business Media, 2013).

30. H. K. Khalil, *Nonlinear Systems*, 3rd ed. (Prentice Hall, 2002).

31. L. Sun, d C C. Visser, Q. P. Chu, and W. Falkena, "Hybrid Sensor-Based Backstepping Control Approach With Its Application to Fault-Tolerant Flight Control," *Journal of Guidance, Control, and Dynamics* 37, no. 1 (2014): 59–71.

32. X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proc. 13th Int. Conf. Artif. Intell. Stat. (AISTATS), Sardinia, Italy* (JMLR Workshop and Conf. Proc, 2010), 249–256.

Appendix A

In this section, we provide the proof of Theorem 1. According to Assumption 1, the NN approximation property (12) holds when $x_p(t) \in S_p$. The proof provided in this Appendix shows that if $x_p(0) \in S_p$, then $x_p(t)$ evolves within S_p for all $t > 0$, preventing a circular argument. Toward that end, we start, in Appendix A, by deriving error dynamics that hold when $x_p(t) \in S_p$. Useful inequalities are then listed in several remarks, which are used in the proof that is given in Appendix B, showing that $x_p(t)$ does not leave S_p, once it is initialized in S_p. Therefore, the states of the system given in (1) are not assumed to be in S_p for all times, a priori, but it is proven. This is important to prevent a circular argument.

Error Dynamics and Useful Remarks for $\mathbf{x_p(t) \in S_p}$

We start by substituting the control input (9) and the baseline controller (6) into (4), which yields

$$\dot{x}(t) = A_m x(t) + B_m r(t) + B(u_{ad}(t) + u_{lstm}(t) + f(x_p(t))) \tag{A1}$$

Using (8), (A1) and (7), the state-tracking error dynamics can be written as

$$\dot{e}(t) = A_m e(t) + B(u_{ad}(t) + u_{lstm}(t) + f(x_p(t))) \tag{A2}$$

which, by using the NN approximation property (12), can be written as

$$\dot{e}(t) = A_m e(t) + B(u_{ad}(t) + W^T\sigma(V^T x_p(t)) + u_{lstm}(t) + \varepsilon(x_p(t))) \tag{A3}$$

It is noted that (A3) and all subsequent remarks hold when $x_p(t) \in S_p$, while in this article, we only assume that $x_p(0) \in S_p$. The following remarks are used later in the proof in the next section, where it is shown that once x_p is initialized in S_p, it stays in S_p.

The ANN controller $u_{ad}(t)$ is chosen to be the output of a multi-layer NN, as defined in (13) and (14). Substituting (13) and (14) into (A3), yields

$$\begin{aligned}\dot{e}(t) = A_m e(t) + B(&-\hat{W}^T(t)\sigma(\hat{V}^T(t)x_p(t)) + W^T\sigma(V^T x_p(t))\\ &+ u_{lstm}(t) + \upsilon(t) + \varepsilon(x_p(t)))\end{aligned} \tag{A4}$$

where $\hat{V}(t) \in \mathbb{R}^{n_h \times n}$ and $\hat{W}(t) \in \mathbb{R}^{m \times n_h}$ are adaptive NN weights serving as estimates for the unknown ideal weights V and W, respectively. The weight estimation errors are defined as

$$\tilde{W}(t) = W - \hat{W}(t) \tag{A5a}$$

$$\tilde{V}(t) = V - \hat{V}(t) \tag{A5b}$$

Throughout the remainder of this section, we drop the time dependency notation for simplicity. Further, we denote $\sigma \triangleq \sigma(V^T x_p(t))$ and $\hat{\sigma} \triangleq \sigma(\hat{V}^T x_p(t))$. Then, (A4) can be written as

$$\dot{e} = A_m e + B(-\hat{W}^T\hat{\sigma} + W^T\sigma + u_{lstm} + \upsilon + \varepsilon(x_p)) \tag{A6}$$

Using (A5a) in (A6) yields

$$\dot{e} = A_m e + B(\tilde{W}^T\hat{\sigma} + W^T(\sigma - \hat{\sigma}) + u_{lstm} + \upsilon + \varepsilon(x_p)) \tag{A7}$$

Remark 4. Since $r(t)$ is bounded and A_m is Hurwitz, it follows from (7) that

$$\|x_m(t)\| \le C_m \tag{A8}$$

where $C_m > 0$ is a known constant, which depends on the user-defined reference input $r(t)$. Therefore, it follows from (8), and the fact that $x(t) \triangleq [x_p(t)^T, x_e(t)^T]^T \in \mathbb{R}^n$, that

$$\|x_p(t)\| \leq \|x(t)\| \leq C_m + \|e(t)\| \tag{A9}$$

Remark 5. From the Taylor series expansion

$$\sigma(V^T x_p) = \sigma(\hat{V}^T x_p) + \hat{\sigma}' \tilde{V}^T x_p + O(\tilde{V}^T x_p)^2 \tag{A10}$$

one can write

$$\sigma(V^T x_p) - \sigma(\hat{V}^T x_p) = \hat{\sigma}' \tilde{V}^T x_p + O(\tilde{V}^T x_p)^2 \tag{A11}$$

where $\hat{\sigma}'$ is defined in (16) and $O(\tilde{V}^T x_p)^2$ denote the higher order terms in the series. Furthermore, since

$$O(\tilde{V}^T x_p)^2 = \sigma(V^T x_p) - \sigma(\hat{V}^T x_p) - \hat{\sigma}' \tilde{V}^T x_p \tag{A12}$$

then, for sigmoid and tanh activation functions, $\|\sigma(.)\| \leq C_\sigma$ and $\|\sigma'(.)\| \leq C_{\sigma'}$ for known constants $C_\sigma > 0$ and $C_{\sigma'} > 0$. Hence, using (A9), the higher order terms in the Taylor series are bounded by

$$\|O(\tilde{V}^T x_p)^2\| \leq 2C_\sigma + C_{\sigma'} \|\tilde{V}\|_F C_m + C_{\sigma'} \|\tilde{V}\|_F \|e\| \tag{A13}$$

Substituting (A11) for $(\sigma - \hat{\sigma})$ in (A7), yields

$$\begin{aligned} \dot{e} = {} & A_m e + B(\tilde{W}^T \hat{\sigma} + W^T \hat{\sigma}' \tilde{V}^T x_p \\ & + W^T O(\tilde{V}^T x_p)^2 + u_{lstm} + v + \varepsilon(x_p)) \end{aligned} \tag{A14}$$

which, by using (A5a) and (A5b), can be written as

$$\dot{e} = A_m e + B(\tilde{W}^T (\hat{\sigma} - \hat{\sigma}' \hat{V}^T x_p) + \hat{W}^T \hat{\sigma}' \tilde{V}^T x_p + w + u_{lstm} + v) \tag{A15}$$

where

$$w \triangleq \tilde{W}^T \hat{\sigma}' V^T x_p + W^T O(\tilde{V}^T x_p)^2 + \varepsilon(x_p) \tag{A16}$$

Remark 6. Let

$$Z \triangleq \begin{bmatrix} W & 0_{n_h \times n_h} \\ 0_{n_p \times m} & V \end{bmatrix} \in \mathbb{R}^{(n_h+n_p)\times(m+n_h)} \tag{A17}$$

then, using (A17) and Assumption 2, one can write

$$\|Z\|_F^2 = \|W\|_F^2 + \|V\|_F^2 \leq W_M^2 + V_M^2 \tag{A18}$$

and hence, $\|Z\|_F \leq Z_M$, where $Z_M \triangleq \sqrt{W_M^2 + V_M^2}$. Furthermore, it follows from (A18) that $\|W\|_F^2 \leq Z_M$ and $\|V\|_F^2 \leq Z_M$. Similarly, by defining

$$\hat{Z}(t) \triangleq \begin{bmatrix} \hat{W}(t) & 0_{n_h \times n_h} \\ 0_{n_p \times m} & \hat{V}(t) \end{bmatrix} \in \mathbb{R}^{(n_h+n_p)\times(m+n_h)} \tag{A19}$$

one can show that $\|\tilde{W}(t)\|_F \leq \|\tilde{Z}(t)\|_F$ and $\|\tilde{V}(t)\|_F \leq \|\tilde{Z}(t)\|_F$, where $\tilde{Z}(t) \triangleq Z - \hat{Z}(t)$.

Remark 7. Using (11), (A13) and Remark 6, one can write

$$\begin{aligned} \|w\| \leq {} & \|\tilde{Z}\|_F C_{\sigma'} Z_M (C_m + \|e\|) \\ & + Z_M (2C_\sigma + C_{\sigma'} \|\tilde{Z}\|_F C_m + C_{\sigma'} \|\tilde{Z}\|_F \|e\|) + \varepsilon_N \end{aligned} \tag{A20}$$

or

$$\|w\| \leq C_0 + C_1 \|\tilde{Z}\|_F + C_2 \|\tilde{Z}\|_F \|e\| \tag{A21}$$

where

$$C_0 \triangleq 2C_\sigma Z_M + \varepsilon_N \tag{A22a}$$

$$C_1 \triangleq 2C_{\sigma'} C_m Z_M \tag{A22b}$$

$$C_2 \triangleq 2C_{\sigma'} Z_M \tag{A22c}$$

Remark 8. Sigmoid and tanh activation functions help define a bound on the LSTM output: We can write (21) as

$$h^n = \begin{pmatrix} \sigma_{g_1}(\cdots)\sigma_{c1}(\cdots) \\ \vdots \\ \sigma_{g_{N_h}}(\cdots)\sigma_{c_{N_h}}(\cdots) \end{pmatrix} \in \mathbb{R}^{N_h} \tag{A23}$$

where N_h is the number of last hidden layer neurons. Then,

$$\|h^n\|^2 = \sum_{i=1}^{N_h} \sigma_{g_i}^2(\cdots)\sigma_{c_i}^2(\cdots) \leq N_h \tag{A24}$$

Hence, $\|h^n\| \leq \sqrt{N_h}$ and the bound on the LSTM output can then be defined as

$$\|u_{lstm}^n\| \leq \|W_{fc}\|_F \sqrt{N_h} + \|b_{fc}\| \triangleq \bar{u}_{lstm} \tag{A25}$$

which by using (27) implies that $\|u_{lstm}(t)\| \leq \bar{u}_{lstm}$.

Appendix B

Proof of Theorem 1

Let the approximation property (12) hold on a known compact set $S_p \triangleq \{x_p : \|x_p\| \leq b_x\}$ as stated in Assumption 1, for some $b_x > C_m$. Defining $S \triangleq \{x : \|x\| \leq b_x\}$, and using the fact that $\|x_p\| \leq \|x\|$, $x \in S$ implies that $x_p \in S_p$. Consider the compact set $S_e \triangleq \{e : \|e\| \leq b_x - C_m\}$, which imply that once $e \in S_e$, then $x_p \in S_p$. Let $e(0) \in S_e$. Then, $x_p(0) \in S_p$ and (12) holds. The proof proceeds by showing that $e(t) \in S_e, \forall t \geq 0$.

Consider the Lyapunov function

$$V = e^T P e + \mathrm{tr}\{\tilde{W}^T F^{-1} \tilde{W}\} + \mathrm{tr}\{\tilde{V}^T G^{-1} \tilde{V}\} \tag{B1}$$

Differentiating (B1) along the trajectories (15) and (A15), and by using (17), $\mathrm{tr}\{A_1 A_2\} = \mathrm{tr}\{A_2 A_1\}$, and $\dot{\tilde{W}} = -\dot{\hat{W}}$, it can be shown that

$$\begin{aligned} \dot{V} = {} & -e^T Q e + 2e^T P B(w + u_{lstm} + v) \\ & + 2\mathrm{tr}\{\tilde{W}^T \kappa \|e\| \hat{W}\} + 2\mathrm{tr}\{\tilde{V}^T \kappa \|e\| \hat{V}\} \end{aligned} \tag{B2}$$

Using (A.5), (A17), and (A19), one can write (B2) as

$$\begin{aligned} \dot{V} &= -e^T Q e + 2e^T P B(w + u_{lstm} + v) + 2\kappa \|e\| \mathrm{tr}\{\tilde{W}^T (W - \tilde{W})\} \\ &\quad + 2\kappa \|e\| \mathrm{tr}\{\tilde{V}^T (V - \tilde{V})\} \\ &= -e^T Q e + 2e^T P B(w + u_{lstm} + v) + 2\kappa \|e\| \mathrm{tr}\{\tilde{Z}^T (Z - \tilde{Z})\} \end{aligned} \tag{B3}$$

Since $\mathrm{tr}\{\tilde{Z}^T (Z - \tilde{Z})\} = \mathrm{tr}\{\tilde{Z}^T Z\} - \|\tilde{Z}\|_F^2 \leq \|\tilde{Z}\|_F (Z_M - \|\tilde{Z}\|_F)$, it follows from (B3) that

$$\begin{aligned} \dot{V} \leq {} & -\lambda_{min}(Q) \|e\|^2 + 2\|B^T P e\| (\|w\| + \|u_{lstm}\|) \\ & + 2e^T P B v + 2\kappa \|e\| \|\tilde{Z}\|_F (Z_M - \|\tilde{Z}\|_F) \end{aligned} \tag{B4}$$

which, by using (A21) and Remark 8, can be written as

$$\begin{aligned} \dot{V} \leq {} & -\lambda_{min}(Q) \|e\|^2 + 2\|B^T P e\| (C_0 + \bar{u}_{lstm}) \\ & + 2C_1 \|B^T P e\| \|\tilde{Z}\|_F + 2C_2 \|B^T P e\| \|e\| \|\tilde{Z}\|_F \\ & + 2e^T P B v + 2\kappa \|e\| \|\tilde{Z}\|_F (Z_M - \|\tilde{Z}\|_F) \end{aligned} \tag{B5}$$

The proposed robustifying term (18) is designed to cancel the 4th term of (B5). Two cases follow from using (18) in (B5): (*a*) $\|B^T Pe\| = 0$ and (*b*) $\|B^T Pe\| \neq 0$.

Case a $\|B^T Pe\| = 0$: Since $v = 0$, (B5) reduces to

$$\begin{aligned}\dot{V} &\leq -\lambda_{min}(Q)\|e\|^2 + 2\kappa\|e\|\|\tilde{Z}\|_F(Z_M - \|\tilde{Z}\|_F) \\ &= -\|e\|(\lambda_{min}(Q)\|e\| + 2\kappa\|\tilde{Z}\|_F(\|\tilde{Z}\|_F - Z_M)) \\ &= -\|e\|\left(\lambda_{min}(Q)\|e\| + 2\kappa(\|\tilde{Z}\|_F - \frac{Z_M}{2})^2 - \kappa\frac{Z_M^2}{2}\right)\end{aligned} \tag{B6}$$

which implies that $\dot{V} < 0$ if either

$$\|e\| > \frac{\kappa Z_M^2}{2\lambda_{min}(Q)} \tag{B7}$$

or

$$\|\tilde{Z}\|_F > Z_M \tag{B8}$$

Case b $\|B^T Pe\| \neq 0$: Then, by using $\|\tilde{Z}\|_F \leq \|\hat{Z}\|_F + Z_M$ in the 4th term of (B5), and substituting (18), it follows that

$$\begin{aligned}\dot{V} \leq &-\lambda_{min}(Q)\|e\|^2 + 2\|B^T Pe\|(C_0 + \bar{u}_{lstm}) \\ &+ 2C_1\|B^T Pe\|\|\tilde{Z}\|_F + 2C_2\|B^T Pe\|\|e\|(\|\hat{Z}\|_F + Z_M) \\ &- 2k_z\|B^T Pe\|\|e\|(\|\hat{Z}\|_F + Z_M) + 2\kappa\|e\|\|\tilde{Z}\|_F(Z_M - \|\tilde{Z}\|_F)\end{aligned} \tag{B9}$$

Since $k_z > C_2$, (B9) reduces to

$$\begin{aligned}\dot{V} &\leq -\lambda_{min}(Q)\|e\|^2 + 2\|B^T Pe\|(C_0 + \bar{u}_{lstm}) \\ &\quad + 2C_1\|B^T Pe\|\|\tilde{Z}\|_F + 2\kappa\|e\|\|\tilde{Z}\|_F(Z_M - \|\tilde{Z}\|_F) \\ &\leq -\lambda_{min}(Q)\|e\|^2 + 2\|B^T P\|_F\|e\|(C_0 + \bar{u}_{lstm}) \\ &\quad + 2C_1\|B^T P\|_F\|e\|\|\tilde{Z}\|_F + 2\kappa\|e\|\|\tilde{Z}\|_F(Z_M - \|\tilde{Z}\|_F) \\ &= -\|e\|\big(\lambda_{min}(Q)\|e\| - 2\|B^T P\|_F(C_0 + \bar{u}_{lstm}) \\ &\quad -2C_1\|B^T P\|_F\|\tilde{Z}\|_F - 2\kappa\|\tilde{Z}\|_F(Z_M - \|\tilde{Z}\|_F)\big) \\ &= -\|e\|\left(\lambda_{min}(Q)\|e\| + 2\kappa(\|\tilde{Z}\|_F - \frac{q_1}{2})^2 - \kappa\frac{q_1^2}{2} - 2q_0\right)\end{aligned} \tag{B10}$$

where $q_0 \triangleq \|B^T P\|_F(C_0 + \bar{u}_{lstm})$ and $q_1 \triangleq C_1\|B^T P\|_F/\kappa + Z_M$. Therefore, it follows that $\dot{V} < 0$ if either

$$\|e\| > \frac{\kappa q_1^2 + 4q_0}{2\lambda_{min}(Q)} \tag{B11}$$

or

$$\|\tilde{Z}\|_F > \frac{q_1}{2} + \sqrt{\frac{q_1^2}{4} + \frac{q_0}{\kappa}} \tag{B12}$$

Hence, using the fact that the bounds given in (B11) and (B12) are bigger than the bounds in (B7) and (B8), and by selecting the Lyapunov matrix Q such that

$$\lambda_{min}(Q) > \frac{\kappa q_1^2 + 4q_0}{2(b_x - C_m)} \tag{B13}$$

it can be shown that the solution $(e(t), \tilde{Z}(t))$ is *uniformly ultimately bounded* [30], and converges to the compact set

$$E \triangleq \left\{(e, \tilde{Z}) : \|e\| \leq \frac{\kappa q_1^2 + 4q_0}{2\lambda_{min}(Q)} < b_x - C_m \right. \\ \left. \text{and } \|\tilde{Z}\|_F \leq \frac{q_1}{2} + \sqrt{\frac{q_1^2}{4} + \frac{q_0}{\kappa}}\right\} \tag{B14}$$

This implies that, since $e(0) \in S_e$ and $\dot{V} < 0$ on the boundary of S_e, $e(t)$ stays within S_e for all $t \geq 0$. Therefore $x_p(t) \in S_p$ for all $t \geq 0$.

Since the reference input $r(t)$ is bounded, $x_m(t)$ is bounded. Therefore, the boundedness of $e(t) = x(t) - x_m(t)$ implies that $x(t)$, and therefore all system signals, are bounded. ■

Appendix C

Adaptive Control: Linearly Parameterized Uncertainty

A typical, most common design of the adaptive control term $u_{ad}(t)$ in (9) revolves around the following assumption.

Assumption A1. The matched uncertainty in (1) is linearly parameterized as

$$f(x_p(t)) = W_p^T \phi(x_p(t)) \tag{C1}$$

where $W_p \in \mathbb{R}^{n_\phi \times m}$ is an unknown weight matrix and $\phi(x_p(t)) : \mathbb{R}^{n_p} \to \mathbb{R}^{n_\phi}$ is a known basis function, whose components are locally Lipschitz-continuous functions of the plant states $x_p(t)$.

For a dynamical system in the form of (5), the adaptive controller takes the form

$$u_{ad}(t) = -\hat{f}(x_p(t)) \tag{C2}$$

where $\hat{f}(x_p(t))$ represents an estimate of $f(x_p(t))$, which by using (C1) can be defined as

$$\hat{f}(x_p(t)) = \hat{W}_p^T(t)\phi(x_p(t)) \tag{C3}$$

where $\hat{W}_p(t) \in \mathbb{R}^{n_\phi \times m}$ is an adjustable parameter, serving as an estimate of the ideal value W_p. Using the state-tracking error $e(t) \triangleq x(t) - x_m(t)$ in (8), $\hat{W}_p(t)$ is updated using the adaptive law

$$\dot{\hat{W}}(t) = \Gamma\phi(x_p(t))e^T(t)PB - \Gamma\kappa\|e(t)\|\hat{W}(t) \tag{C4}$$

where $\Gamma \in \mathbb{R}^{n_\phi \times n_\phi}$ is a symmetric positive definite matrix, serving as a learning rate, $\kappa > 0$ is a scalar gain, and $P \in \mathbb{R}^{n \times n}$ is the symmetric positive definite solution of the Lyapunov equation (17), for some symmetric positive definite matrix $Q \in \mathbb{R}^{n \times n}$.

Theorem 2. *Consider the uncertain plant dynamics* (1), *subject to Assumption A1, and the reference model (7). Let the control input be defined by (9), which consists of the baseline controller (6), the adaptive controller defined by (C2), (C3), and (C4), and the LSTM controller explained in Section 3. Then, the solution $(e(t), \tilde{W}_p(t))$ is uniformly ultimately bounded (UUB) and converges to a predefined compact set, where $\tilde{W}_p(t) \triangleq W_p - \hat{W}_p(t)$.*

Proof. Substituting the control input (9) and the baseline controller (6) into (4), and by using (7), the state-tracking error dynamics can be written as in (A2), which, by using Assumption A1, can be written as

$$\dot{e}(t) = A_m e(t) + B\left(u_{ad}(t) + u_{lstm}(t) + W_p^T\phi(x_p(t))\right) \tag{C5}$$

Substituting the adaptive control law (C2) into (C6) yields

$$\dot{e}(t) = A_m e(t) + B\left(\tilde{W}_p^T(t)\phi(x_p(t)) + u_{lstm}(t)\right) \tag{C6}$$

where $\tilde{W}_p(t) \triangleq W_p - \hat{W}_p(t)$.

Consider the Lyapunov function candidate

$$V_p = e^T Pe + \mathrm{tr}\{\tilde{W}_p^T\Gamma^{-1}\tilde{W}_p\} \tag{C7}$$

the derivative of which, along the trajectories (C6) and (C4), can be written as

$$\begin{aligned}\dot{V}_p = &-e^T Qe - 2\kappa\|e\|\mathrm{tr}\{\tilde{W}_p^T\tilde{W}_p\} \\ &+ 2\kappa\|e\|\mathrm{tr}\{\tilde{W}_p^T W_p\} + 2e^T PBu_{lstm}\end{aligned} \tag{C8}$$

It follows from Remark 9 that $\|u_{lstm}(t)\| \leq \bar{u}_{lstm}$, for a known $\bar{u}_{lstm}$, which is defined in (A25). Therefore, (C8) implies that

$$\begin{aligned}\dot{V}_p &\leq -\lambda_{min}(Q)\|e\|^2 - 2\kappa\|e\|\|\tilde{W}_p\|_F^2 \\ &\quad + 2\kappa\|e\|\|\tilde{W}_p\|_F\|W_p\|_F + 2\|e\|\|PB\|\bar{u}_{lstm} \\ &= -\|e\|\left(\lambda_{min}(Q)\|e\| + 2\kappa\left(\|\tilde{W}_p\|_F - \frac{\|W_p\|_F}{2}\right)^2\right. \\ &\qquad \left. - \kappa\frac{\|W_p\|_F^2}{2} - 2\|PB\|\bar{u}_{lstm}\right)\end{aligned} \tag{C9}$$

Thus, it follows that $\dot{V} < 0$ if either

$$\|e\| > \frac{\kappa\|W_p\|_F^2/2 + 2\|PB\|\bar{u}_{lstm}}{\lambda_{min}(Q)} \triangleq p_e \tag{C10}$$

or

$$\|\tilde{W}_p\|_F > \frac{\|W_p\|_F^2}{2} + \sqrt{\frac{\|W_p\|_F^2}{4} + \frac{\|PB\|\bar{u}_{lstm}}{\kappa}} \triangleq p_w \tag{C11}$$

and hence, the solution $(e, \tilde{W}_p)$ is UUB [30] and converges to the set $E_p \triangleq \left\{(e, \tilde{W}_p) : \|e\| \leq p_e \text{ and } \|\tilde{W}_p\|_F \leq p_w\right\}$. □