Human-in-the-Loop Performance Evaluation of an Adaptive Control Framework with Long Short-Term Memory Augmentation

M. Yusuf Uzun, Emirhan Inanc, Abdullah Habboush and Yildiray Yildiz

***Abstract*— This study investigates the human-in-the-loop performance of a novel control framework, where a Long Short-Term Memory (LSTM) network augments an adaptive neural network (ANN) controller. The method drastically improves the transient response compared to conventional approaches, especially in the presence of significant and rapid changes in the uncertainties. LSTM network, which uses the knowledge of input sequence dependencies, predicts and compensates for the deviation of the ANN controller from its ideal behavior. Although this control framework is shown to provide improved transients, its interactions with a human operator need to be analyzed to ensure a safe operation. In this study, first, a human pilot model is used to investigate the overall system's behavior and analyze the controller's performance for a reference tracking task. Then, human-in-the-loop experiments are conducted to analyze how the system responds in the presence of a real human operator in the loop.**

I. Introduction

Online-tuned feed-forward neural network (NN) based adaptive control has shown to be effective in controlling uncertain systems [1], [2], [3], [4]. Nevertheless, such methods may suffer from oscillatory behavior resulting from sudden changes in plant dynamics. A recently introduced method [5] addresses this problem by augmenting a conventional adaptive neural network (ANN) with a Long Short-Term Memory (LSTM) network. In [5], LSTM is trained offline and designed to redeem the deficiencies of the ANN controller.

Although the control method [5] has been shown to dramatically improve the transient response for a pitch rate control task with predefined reference inputs, its interactions with a human operator still need to be investigated. It is a known issue that adaptive controllers and pilots may interact in an undesired fashion [6], [7]. Statistically, pilot error and unfavorable interactions between the pilot and the flight control system are responsible for two-thirds of all aviation accidents [8]. Therefore, it is essential to carefully analyze how well the adaptive controller and LSTM network conform to the human pilot's intended inputs. To achieve this, we have conducted several human-in-the-loop experiments to test the aforementioned controller.

Another goal of this study is to investigate the effect of enhancing the ANN controller with the LSTM network on the overall closed-loop performance. To understand this effect, simulations are performed using a human pilot model following a continuously changing pitch reference. With these simulations, the controller's performance is evaluated under different uncertainties and reference signals. It is shown that although the ANN controller performs reasonably well under certain conditions, it fails to provide adequate performance without the LSTM network enhancement when the uncertainties get larger. We have also demonstrated that LSTM enhancement dramatically improves the transient response in human-in-the-loop experiments.

This work was supported by the Scientific and Technological Research Council of Turkey under Grant 121E384. M. Y. Uzun, E. Inanc, A. Habboush, and Y. Yildiz are with the Department of Mechanical Engineering Bilkent University, Cankaya, Ankara 06800, Turkey. (emails:yusuf.uzun@bilkent.edu.tr, emirhan.inanc@bilkent.edu.tr, a.habboush@bilkent.edu.tr, yyildiz@bilkent.edu.tr).

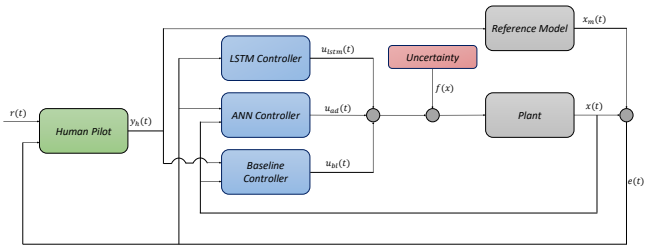

Fig. 1: Block diagram of the overall control architecture.

This paper is organized as follows. In Section II, the human pilot model, ANN controller, LSTM network augmentation, and LSTM network training method are explained. In Section III, simulation and experimental results are provided. Finally, a summary is given in Section IV.

II. Problem Formulation

A. Pilot Model

We use a generic model to represent the pilot as [9]

$$\dot{\xi}(t) = A_{\mathrm{h}}\xi(t) + B_{\mathrm{h}}e_r(t-\tau), \quad \xi(0) = \xi_0, \tag{1a}$$

$$y_h(t) = C_{\mathrm{h}}\xi(t) + D_{\mathrm{h}}e_r(t-\tau), \tag{1b}$$

where $\xi(t) \in \mathbb{R}^{n_\xi}$ is the internal human state vector, $\tau \in \mathbb{R}_+$is the internal human time delay, $A_{\mathrm{h}} \in \mathbb{R}^{n_\xi \times n_\xi}$, $B_{\mathrm{h}} \in \mathbb{R}^{n_\xi \times n_{\mathrm{r}}}$, $C_{\mathrm{h}} \in \mathbb{R}^{n_{\mathrm{c}} \times n_\xi}$, $D_{\mathrm{h}} \in \mathbb{R}^{n_{\mathrm{c}} \times n_{\mathrm{r}}}$ and $y_h(t) \in \mathbb{R}^{n_{\mathrm{c}}}$ is the output of the human or the commanded input to the inner loop architecture. Input into human dynamics is,

$$e_r(t) \triangleq r(t) - E_{\mathrm{h}}x(t), \tag{2}$$

where $e_r(t) \in \mathbb{R}^{n_{\mathrm{r}}}$ is the output tracking error and $r(t) \in \mathbb{R}^{n_{\mathrm{r}}}$ is the reference to the system, which is assumed to be bounded. $x(t) \in \mathbb{R}^n$ is the state vector, explained below, and $E_{\mathrm{h}} \in \mathbb{R}^{n_{\mathrm{r}} \times n}$ is the state selection matrix.

B. Plant Dynamics with a Baseline Controller

Consider the plant dynamics

$$\dot{x}(t) = Ax(t) + B(u(t) + f(x(t))), \tag{3a}$$
$$y(t) = C^T x(t), \tag{3b}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the control input, $f(x(t)) : \mathbb{R}^n \to \mathbb{R}^m$ is a state-dependent matched uncertainty, $A \in \mathbb{R}^{n\times n}$ is a known system matrix, $B \in \mathbb{R}^{n\times m}$ is a known control input matrix, and $C \in \mathbb{R}^{n\times s}$ is a known output matrix. $y(t) \in \mathbb{R}^s$ is the output of interest. The pair (A, B) is assumed to be controllable.

Assumption 1: The uncertainty $f(x(t))$ in (3) is continuous on a known compact set $S \triangleq \{x(t) : \|x(t)\| \leq b_x\} \subset \mathbb{R}^n$.

It follows from Assumption 1 that, a multi-layer neural network (NN) in the following form can be used to approximate $f(x(t))$ as

$$f(x(t)) = W^T \bar{\sigma}(V^T \bar{x}(t)) + \varepsilon(x(t)), \tag{4}$$

with

$$\|\varepsilon(x(t))\| \leq \varepsilon_N \quad , \quad \forall x \in S, \tag{5}$$

where a known positive constant ε_N bounds the NN error vector $\varepsilon(x(t))$ in $S \subset \mathbb{R}^n$. The NN input and the hidden layer output are

$$\bar{x}(t) \triangleq \begin{bmatrix} x(t)^T & 1 \end{bmatrix}^T \in \mathbb{R}^{n+1}, \tag{6a}$$
$$\bar{\sigma}(V^T \bar{x}(t)) \triangleq \begin{bmatrix} \sigma(V^T \bar{x}(t)) & 1 \end{bmatrix}^T \in \mathbb{R}^{n_h+1}, \tag{6b}$$

respectively. The nonlinear activation function $\sigma(.) : \mathbb{R}^{n_h} \to \mathbb{R}^{n_h}$ is a bounded function such as sigmoid or tanh. The weight matrices, V and W, include bias terms, and the unity elements are concatenated to account for these biases. We drop the overbar notation and rewrite (4) as

$$f(x(t)) = W^T \sigma(V^T x(t)) + \varepsilon(x(t)), \tag{7}$$

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

Assumption 2: The unknown ideal weights W and V are bounded by known positive constants such that $\|V\|_F \leq V_M$ and $\|W\|_F \leq W_M$ [1].

Let $u_{bl}(t) \in \mathbb{R}^m$ be a baseline control input, given as

$$u_{bl}(t) = -L_x x(t) + L_r y_h(t), \tag{8}$$

where $y_h(t) \in \mathbb{R}^s$ is the reference of the inner loop provided by the human pilot, and it is assumed that $L_x \in \mathbb{R}^{m\times n}$ can be selected such that $A_m \triangleq A - BL_x$ is stable. It is noted that the reference input $y_h(t)$ is bounded due to manipulator limits. Defining $B_m \triangleq BL_r$, where $L_r \in \mathbb{R}^{m\times s}$, a reference model is determined as

$$\dot{x}_m(t) = A_m x_m(t) + B_m y_h(t). \tag{9}$$

For a constant y_h, at steady state,

$$A_m x_m(\infty) + B_m y_h = 0, \tag{10a}$$
$$x_m(\infty) = -A_m^{-1} B L_r y_h. \tag{10b}$$

Hence, once $\lim_{t\to\infty} x(t) = x_m(t)$, the plant output $y(t)$, takes the form $y(\infty) = -C^T A_m^{-1} B L_r y_h$. Selecting

$$L_r = -\left(C^T A_m^{-1} B\right)^{-1}, \tag{11}$$

results in $\lim_{t\to\infty} y(t) = y_h$.

The state tracking error is defined as

$$e(t) = x(t) - x_m(t). \tag{12}$$

The control signal is given as

$$u(t) = u_{bl}(t) + u_{ad}(t) + u_{lstm}(t) + v(t), \tag{13}$$

where $u_{bl}(t)$ is the baseline controller, $u_{ad}(t)$ is an adaptive neural network (ANN) controller designed to compensate for the matched uncertainty $f(x(t))$, $v(t)$ is a robustifying term, and $u_{lstm}(t)$ is the output of the LSTM network.

C. Adaptive Neural Network Controller

The adaptive controller input can be expressed as

$$u_{ad}(t) = -\hat{f}(x(t)), \tag{14}$$

where $\hat{f}(x(t))$ represents an estimate of $f(x(t))$ defined in (7). $\hat{f}(x(t))$ is constructed as

$$\hat{f}(x(t)) = \hat{W}^T(t)\sigma(\hat{V}^T(t)x(t)), \tag{15}$$

where $\hat{V}(t) \in \mathbb{R}^{n_h\times n}$ and $\hat{W}(t) \in \mathbb{R}^{m\times n_h}$ are estimates for the unknown ideal weight matrices V and W, respectively. The weights are updated using the adaptive laws [1]

$$\dot{\hat{W}} = F(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)e^T PB - F\kappa\|e\|\hat{W}, \tag{16a}$$
$$\dot{\hat{V}} = Gxe^T PB\hat{W}^T\hat{\sigma}' - G\kappa\|e\|\hat{V}, \tag{16b}$$

where $F \in \mathbb{R}^{n_h\times n_h}$ and $G \in \mathbb{R}^{n\times n}$, learning rate matrices, are symmetric positive definite, and κ is a positive scalar gain. $\hat{\sigma}$ and $\hat{\sigma}'$ are defined as

$$\hat{\sigma} \triangleq \sigma(\hat{V}^T(t)x(t)), \quad \hat{\sigma}' \triangleq \left.\frac{d\sigma(z)}{dz}\right|_{z=\hat{V}^T(t)x(t)}. \tag{17}$$

Furthermore, the symmetric positive definite matrix $P \in \mathbb{R}^{n\times n}$ is obtained as the solution to the Lyapunov equation $A_m^T P + PA_m = -Q$, where $Q \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix.

The robustifying term $v(t)$ in (13) is given as [10]

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

with

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

where $Z_M \triangleq \sqrt{W_M^2 + V_M^2}$ and $C_2 \triangleq 2C_{\sigma'}Z_M$. For sigmoid and tanh activation functions, $\|\sigma'(.)\| \leq C_{\sigma'}$ for a known constant $C_{\sigma'} > 0$, where $\sigma'(z) = d\sigma(z)/dz$.

D. Long Short-Term Memory (LSTM) Network

Using the state tracking error (12) as an input, the LSTM network computes the control input $u_{lstm}(t)$ in (22) to compensate for the Adaptive Neural Network (ANN) controller's shortcomings.

The LSTM network propagates the information for a longer time, compared to recurrent neural network (RNN), using its gate operations given below, where e^n_{norm} is the normalized version of the input sequence e^n, which are sampled instances of the error signal (12), h^n is the hidden state, and c^n is the cell state. The recurrent weight matrices, input weight matrices, and bias vectors are denoted by R, W, and b, respectively. The gate activation function (sigmoid) is denoted by σ_g, and the state activation function (tanh) is denoted by σ_c. The forget gate, cell candidate, input gate, and output gate are denoted by the subscripts f, g, i, and o respectively. The current time step of the LSTM is denoted by the superscript n, while the previous time step is denoted by $n-1$. The output of the gate operations, which are denoted by the variables $out^n_f, out^n_o, out^n_i$, and $\bar{c}^n$, is given as

$$out^n_f = \sigma_g(W_f e^n_{norm} + R_f h^{n-1} + b_f), \tag{20a}$$
$$out^n_o = \sigma_g(W_o e^n_{norm} + R_o h^{n-1} + b_o), \tag{20b}$$
$$out^n_i = \sigma_g(W_i e^n_{norm} + R_i h^{n-1} + b_i), \tag{20c}$$
$$\bar{c}^n = \sigma_c(W_g e^n_{norm} + R_g h^{n-1} + b_g). \tag{20d}$$

The computation of hidden state h and the cell state c for the next time step is given as

$$c^n = out^n_f \odot c^{n-1} + out^n_i \odot \bar{c}^n, \tag{21a}$$
$$h^n = out^n_o \odot \sigma_c(c^n). \tag{21b}$$

The LSTM network output is calculated by an additional fully connected layer as

$$u^n_{lstm} = W_{fc} h^n + b_{fc}, \tag{22}$$

where W_{fc} is the weight matrix and b_{fc} is the bias vector. The control input $u_{lstm}(t)$ is formed by LSTM output applying *zero-order hold*.

E. LSTM Network Training

The closed-loop system given in Fig. 1 prevents the usage of an already existing data set for the LSTM network training. The dynamic nature of the system causes the training data to evolve after each iteration of training. Therefore, after initializing the LSTM network weights, we put it into a closed-loop setting, where the ANN controller, the baseline controller, and the plant exist. Then, we collect one set of data by running a simulation, train the network, and use the new weights for the next simulation.

Instead of estimating the uncertainty directly, we train the LSTM network to predict and compensate for the estimation error of the ANN, ensuring the stability of the system. The estimation error of the ANN is given as

$$\tilde{f}(x(t)) \triangleq f(x(t)) - \hat{f}(x(t)), \tag{23}$$

where $f(x(t))$ and $\hat{f}(x(t))$ are given in (7) and (15). Then, the *truth* for the LSTM is selected as

$$y(t) = -\tilde{f}(x(t)). \tag{24}$$

As a result, the LSTM network sends a signal to the system that is an estimate of the truth, denoted as

$$u_{lstm}(t) = \hat{y}(t). \tag{25}$$

An uncertainty $f_{train}(x(t))$ is given to the system, during the training. A combination of sinusoidal waves is introduced to the system as a reference, and we expect the LSTM network to learn $\tilde{f}_{train}$ which is defined in (23). The LSTM network is trained using the tracking error, defined in (12), as the input. Training the network in this dynamic environment helps LSTM generalize to never seen uncertainty functions.

F. Stability Guarantee

Theorem 1: Considering the uncertain plant dynamics (3) satisfying Assumptions 1 and 2, with the reference model (9), and the control input (13) consisting of the baseline controller (8), the ANN controller (14), (15), and (16), the robustifying term (18) and (19), and the LSTM network output (22). Then, if $y_h(t)$ remains bounded, and given that $x(0) \in S$ (see Assumption 1), the solution $(e(t), \tilde{W}(t), \tilde{V}(t))$, where $\tilde{W}(t) \triangleq W - \hat{W}(t)$ and $\tilde{V}(t) \triangleq V - \hat{V}(t)$, is *uniformly ultimately bounded* and converges to a compact set.

The proof of Theorem 1 can be found in the report presented in [10].

Theorem 2: If A_h in (1a) is Hurwitz, and the reference $r(t)$ in (2) is bounded, all system signals are bounded.

Proof: Combining (2) with (1) gives

$$\dot{\xi}(t) = A_{\rm h}\xi(t) + B_{\rm h}[r(t-\tau) - E_h x(t-\tau)], \tag{26a}$$
$$y_h(t) = C_{\rm h}\xi(t) + D_{\rm h}[r(t-\tau) - E_h x(t-\tau)]. \tag{26b}$$

Since the pilot output y_h is given to the inner loop by an interface with magnitude saturation, then y_h is bounded and hence, it follows from Theorem 1 that $x(t)$ is bounded. Therefore, assuming $r(t)$ is bounded, (26a) implies that, for a stable A_h, ξ, and therefore y_h, is bounded, which completes the proof. ■

III. Simulations and Experiments

This section considers a flight control task, where the pilot makes the aircraft follow a pitch angle reference. This task is simulated using both a pilot model and real human participants. Two different pitch angle references, fast and slow, are used. These references are

$$r(t) = \Sigma_1^7 A_i \sin(f_i t), \tag{27a}$$
$$A = 360/\pi(2\,\mathrm{rand}(7,1) - 1), \tag{27b}$$
$$f = \mathrm{rand}(7,1), \tag{27c}$$

and

$$r(t) = \Sigma_1^7 A_i \sin(f_i t), \tag{28a}$$
$$A = 360/\pi(2\,\mathrm{rand}(7,1) - 1), \tag{28b}$$
$$f = 2\,\mathrm{rand}(7,1), \tag{28c}$$

which are denoted as "slow" and "fast", respectively, where "$\mathrm{rand}(7,1)$" function generates uniformly distributed random numbers (between 0 and 1) of the size 7-by-1. Therefore, $A \in \mathbb{R}^{7\times 1}$, $A_i \in [-360/\pi \quad 360/\pi]$, and $f \in \mathbb{R}^{7\times 1}$ for both cases. For the slow reference case $f_i \in [0 \quad 1]$, whereas for the fast reference case $f_i \in [0 \quad 2]$.

At the outer loop architecture, the plant is assumed to be operated by a pilot whose model is given by [11]

$$k_p \frac{T_p s + 1}{T_z s + 1} e^{-\tau s}, \tag{29}$$

where k_p, T_p and T_z are positive scalars. Furthermore, τ corresponds to the pilot reaction time delay. For the simulations, the parameters of the human pilot model are selected as $T_p = 1$, $T_z = 2$, and $\tau = 0.3$. The pilot gain is taken as $k_p = 0.5$ for the slow reference given in (27), and $k_p = 0.9$ for the fast reference given in (28).

Longitudinal flight dynamics of a Boeing 747, cruising in level flight at an altitude of 40kft and a velocity of 774ft/sec is used as the plant dynamics. State and control input matrices for this plant is given as [12]

$$A_n = \begin{bmatrix} -0.003 & 0.039 & 0 & -0.322 \\ -0.065 & -0.319 & 7.740 & 0 \\ 0.020 & -0.101 & -0.429 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{30a}$$

$$B = \begin{bmatrix} 0.010 & -0.180 & -1.160 & 0 \end{bmatrix}^T. \tag{30b}$$

The state vector is

$$x(t) = \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) & x_4(t) \end{bmatrix}^T, \tag{31}$$

where $x_1(t)$, $x_2(t)$, $x_3(t) = q(t)$ and $x_4(t) = \theta(t)$ are the component of the aircraft's velocity (ft/sec) along the x axis, the component of the aircraft's velocity (ft/sec) along the z axis, the pitch rate (rad/sec), and the pitch angle of the aircraft (rad), respectively. The control input $u(t)$ is the elevator deflection (rad). Elevator deflection magnitude and rate saturation limits are set as $+17/-23$ (deg) and $+37/-37$ (deg/s), respectively [13].

Assigning the feedback gain L_x in (8) is done with an LQR controller with cost matrices $Q_{LQR} = \mathrm{diag}\left([0.001, 0.01, 10, 0.1]\right)$, and $R_{LQR} = 10$.

ANN controller is set to have one hidden layer with eight neurons. The learning rates of the controller are both set to $F = G = 300$, and the Lyapunov matrix Q is set as $Q = 10^{-3} Q_{LQR}$. The outer layer weights $(\hat{W})$ and biases $(\hat{b}_w)$ are initialized to zero, whereas the initial inner weights $(\hat{V})$ and biases $(\hat{b}_v)$ are generated randomly between 0 and 1. The gains k_z in (18) and κ in (16) are set to 0.

The LSTM network has one hidden layer with 128 neurons and since there are four states in the plant (see (31)), the LSTM network receives four state-tracking error components as input. Xavier initialization [14] is used for initialization of weights, and the network is trained using stochastic gradient descent and the Adam optimizer [15] with L2 regularization value of 0.0001, gradient decay of 0.9, and squared gradient decay of 0.999. The simulation step time is 0.01s, and the minibatch size is 1.

During training, the nonlinear uncertainty

$$f_{train}(x) = \begin{cases} 0.1x_3 + 0.2x_1 & \text{if } 0 \le t < 5 \\ 0.05\exp(x_3) + 0.5x_2 & \text{if } 5 \le t < 10 \\ -0.1(x_4 + x_3)^2 & \text{if } 10 \le t < 15 \\ -0.1\cos(x_4) & \text{if } 15 \le t < 20 \\ 0.2(x_4 - x_1) & \text{if } 20 \le t < 25 \\ x_4/(2 + |x_3|) & \text{if } 25 \le t < 30 \\ -\sin(2x_2x_3)/10 & \text{if } 30 \le t < 35 \\ 2(\sin(2x_3x_2) + x_1) & \text{if } 35 \le t < 45 \\ \|x\| & \text{if } 45 \le t < 55 \\ (x_3 + x_4)/3 & \text{if } 55 \le t \le 60 \end{cases} \tag{32}$$

is used. To enrich the training data, we trained the network both with f_{train} and a scaled version $0.1 f_{train}$. The nonlinear uncertainty used for the tests is defined as

$$f_{test}(x) = \begin{cases} 0 & \text{if } 0 \le t < 7 \\ \exp(\sin(x_1))/5 & \text{if } 7 \le t < 12 \\ -0.1\exp(x_1) & \text{if } 12 \le t < 19 \\ 0.1\sin(\|x\|) & \text{if } 19 \le t < 28 \\ 0.05(x_1x_2) & \text{if } 28 \le t < 34 \\ -\|x\|^2/1.5 & \text{if } 34 \le t \le 41 \\ \sin(x_1x_3) & \text{if } 41 \le t \le 49 \\ 0.1(x_4)^2 & \text{if } 49 \le t \le 53 \\ 2\cos(x_2)x_3 & \text{if } 53 \le t \le 60 \end{cases}, \tag{33}$$

where sub-function types and durations are different than that of the uncertainty used for the training.

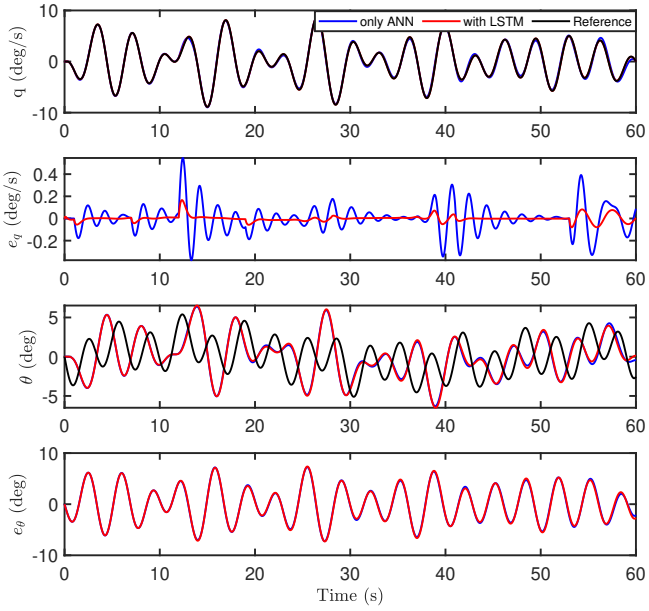

Fig. 2: Tracking performance and errors of pitch rate q and pitch angle θ with and without LSTM augmentation, in the presence of low uncertainty.

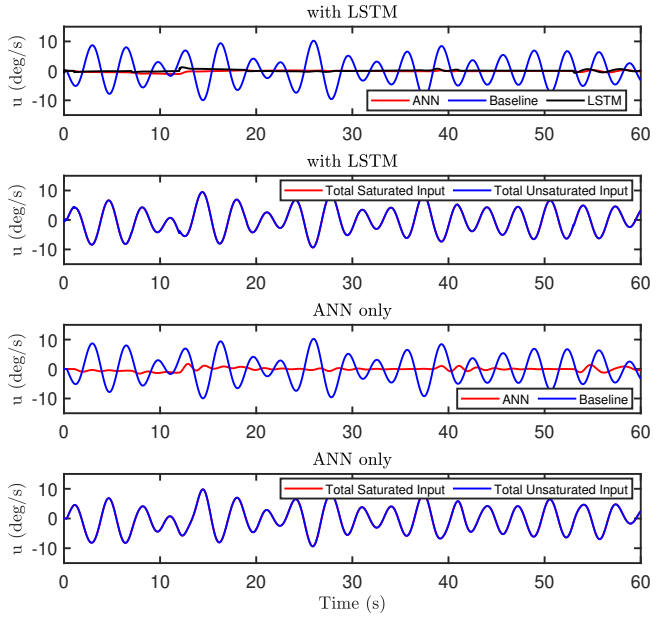

Fig. 3: Control inputs for low uncertainty. LSTM-augmented system: individual inputs (top), total input before/after saturation (2nd). ANN-only system: individual inputs (2nd from bottom), total inputs before/after saturation (bottom).

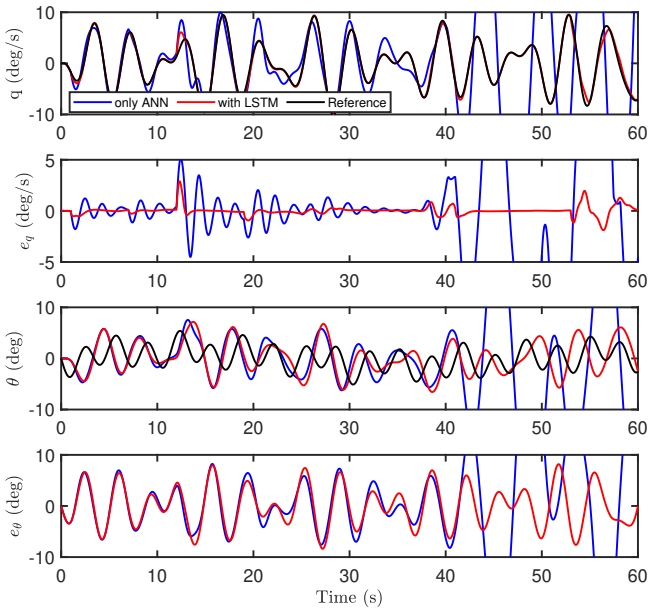

Fig. 4: Tracking performance and errors of pitch rate q and pitch angle θ with and without LSTM augmentation, in the presence of high uncertainty.

During training, the loss function for the LSTM network is selected as

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N} (y - \hat{y})^2, \tag{34}$$

where y is the truth of the LSTM network (see (24)), $\hat{y}$ is the network output (see (25)), and N is the number of data in the data set.

A. Simulations with a pilot model

The results presented in this section analyze the effect of the LSTM network augmentation with fast reference (28) given to the system.

Figures 2 and 3 show the state tracking and control input curves, in the presence of low uncertainty $0.1 f_{test}$ (33). Although LSTM augmented system provides fewer oscillations in the error dynamics, the results show that when the uncertainty is small, the tracking performance of the system with and without the LSTM augmentation is similar.

Figure 4 shows that contrary to the case with a small uncertainty, the closed-loop system provides a reasonable performance only with LSTM augmentation when the uncertainty is large (f_{test}). The size of the oscillations of the ANN-only scenario is significant, and after the $40s$, tracking curves diverge from the reference pitch rate and pitch angle, resulting in excessive oscillations. On the other hand, LSTM augmented system eliminates these oscillations, ensuring stable performance. This result can be further examined in Fig. 5, revealing that the control signals are smoother in the presence of the LSTM network. Unlike the ANN-only case, in the presence of the LSTM augmentation, the controller cancels the uncertainties while staying within the actuator saturation range ($+17/-23$ deg).

B. Human-in-the-loop experiments

During the experiments, participants are expected to perform the pitch reference tracking task using a monitor, and a Logitech Extreme 3D Pro joystick as the interface. Five participants from Bilkent University Mechanical Engineering Department attended the experiments. The performances of all five participants are shown in Fig. 6. To quantify the tracking performance, root mean square errors (RMSE) between the pitch angle output and reference pitch angles are calculated as RMSE $= \sqrt{\sum_{i=1}^{N} (\theta_i - \theta_r)^2 / N}$, where subscript i denotes the i^{th} participant. RMSE values for 4 different scenarios are presented in Table I. In the table, columns show individual participant results. The cells with "LOC" represent the loss of control events. L, H, S, and F represent "low", "high", "slow", and "fast" respectively. Also, "R" and "U" represent the "reference" and "uncertainty". Figure

TABLE I: RMSE of pitch angle (θ) tracking for low uncertainty (LU), high uncertainty (HU), slow reference (SR), and fast reference (FR). LOC represents a loss-of-control event.

Conditions		P1	P2	P3	P4	P5	Average
LU-SR	ANN	0.64	1.40	1.26	0.58	0.48	0.69
	LSTM	0.64	0.97	1.16	0.59	0.51	0.61
LU-FR	ANN	2.75	3.37	3.81	3.03	LOC	3.24
	LSTM	1.91	3.93	4.24	2.57	2.51	2.54
HU-SR	ANN	0.98	1.44	1.69	0.93	0.84	0.92
	LSTM	0.64	0.75	1.40	0.68	0.57	0.63
HU-FR	ANN	LOC	LOC	3.44	LOC	LOC	3.44
	LSTM	2.19	3.09	3.58	2.74	2.82	2.53

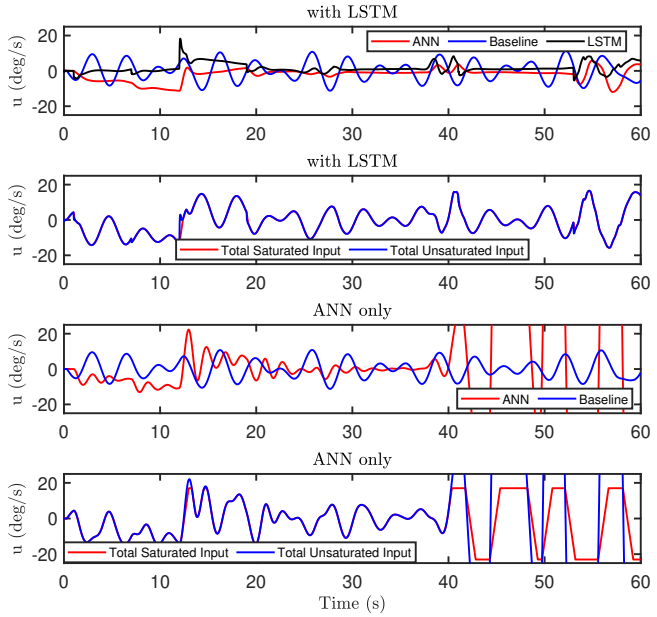

Fig. 5: Control inputs for high uncertainty. LSTM-augmented system: individual inputs (top), total input before/after saturation (2nd). ANN-only system: individual inputs (2nd from bottom), total inputs before/after saturation (bottom).

6 and Table I show that when the uncertainty is low and the reference is slow, LSTM does not make a meaningful contribution to the performance. However, as uncertainty becomes higher, or reference becomes faster, ANN-only tracking performance degrades significantly. Furthermore, in the ANN-only system, some subjects lose control of the system, when the fast reference is used. This loss of control events is mostly observed for high uncertainty cases, although one participant also lost control when the uncertainty is low. When LSTM is augmented, however, no loss of control is experienced.

IV. Summary

This study investigates the tracking performance of an adaptive neural network control structure with Long Short-Term Memory (LSTM) augmentation through simulations and human-in-the-loop experiments. Results reveal significantly improved tracking performance with LSTM augmentation. Future plans include expanding the participant pool for a comprehensive statistical analysis of the performance enhancements provided by LSTM augmentation.

References

[1] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 7, pp. 388–399, 1996.

[2] E. N. Johnson and A. J. Calise, "Neural network adaptive control of systems with input saturation," in *Proc. Amer. Control Conf.*, 2001, pp. 3527–3532.

[3] T. Hayakawa, W. M. Haddad, and N. Hovakimyan, "Neural network adaptive control for a class of nonlinear uncertain dynamical systems with asymptotic stability guarantees," *IEEE Trans. Neural Networks*, vol. 19, no. 1, pp. 80–89, 2008.

[4] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable adaptive neural network control*. Springer Science & Business Media, 2013, vol. 13.

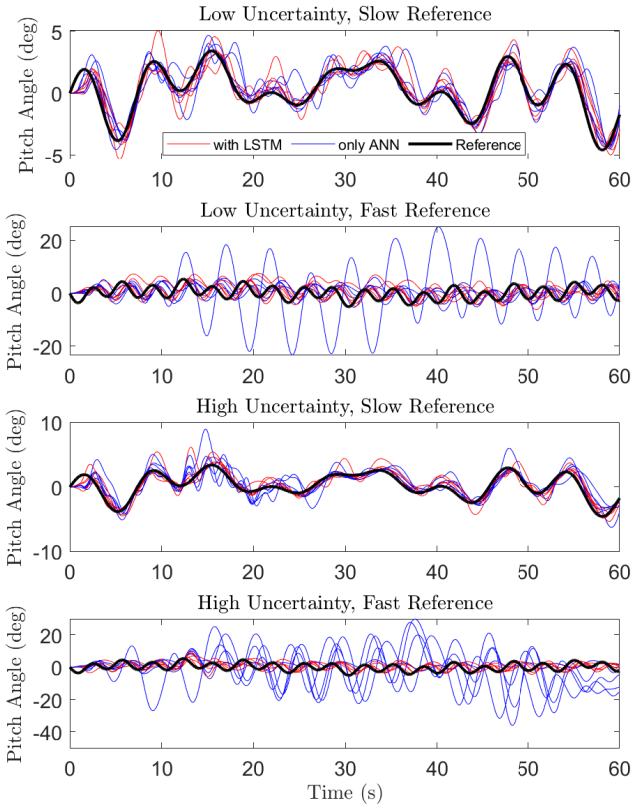

Fig. 6: Individual participants' tracking performance of pitch angle θ with and without LSTM augmentation, for different scenarios.

[5] E. Inanc, Y. Gurses, A. Habboush, and Y. Yildiz, "Neural network adaptive control with long short-term memory augmentation," *American control conference*, submitted for publication, 2023. Available at: https://yildirayyildizcom.files.wordpress.com/2022/11/y.pdf.

[6] D. Klyde, C.-Y. Liang, D. Alvarez, N. Richards, R. Adams, and B. Cogan, "Mitigating unfavorable pilot interactions with adaptive controllers in the presence of failures/damage," in *AIAA Atmospheric Flight Mechanics Conference*, 2011, p. 6538.

[7] A. C. Trujillo, I. M. Gregory, and L. E. Hempley, "Adaptive controller effects on pilot behavior," in *Proceedings of the IEEE 2014 International Conference on Systems, Man, and Cybernetics*, NASA Rept. No. NL1676L-18398, San Diego, CA, Oct. 2014, p. 6.

[8] D. T. McRuer, "Aviation safety and pilot control: Understanding and preventing unfavorable pilot-vehicle interactions," 1997.

[9] T. Yucelen, Y. Yildiz, R. Sipahi, E. Yousefi, and N. Nguyen, "Stability limit of human-in-the-loop model reference adaptive control architectures," *Int. J. Control*, vol. 91, no. 10, pp. 2314–2331, 2018.

[10] E. Inanc, Y. Gurses, A. Habboush, Y. Yildiz, and A. M. Annaswamy, "Neural network adaptive control with long short-term memory," *arXiv:2301.02316*, 2023.

[11] B. J. Bacon and D. K. Schmidt, "An optimal control approach to pilot/vehicle analysis and the neal-smith criteria," *Journal of Guidance, Control, and Dynamics*, vol. 6, no. 5, pp. 339–347, 1983.

[12] A. E. Bryson, *Control of spacecraft and aircraft*, Princeton, NJ, USA: Princeton Univ. Press, 1994.

[13] L. Sun, C. C. de 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*, vol. 37, no. 1, pp. 59–71, 2014.

[14] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, 2010.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.