RESILIENT HUMAN-AUTONOMY COLLABORATION: LEARNING-BASED ASSISTANCE AND SAFE SHARED CONTROL

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

MECHANICAL ENGINEERING

By

Muhammed Yusuf Uzun

August 2025

Resilient Human-Autonomy Collaboration: Learning-Based Assistance and Safe Shared Control
By Muhammed Yusuf Uzun
August 2025

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Yıldıray Yıldız(Advisor)

Ömer Morgül

Mustafa Mert Ankaralı

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

RESILIENT HUMAN-AUTONOMY COLLABORATION: LEARNING-BASED ASSISTANCE AND SAFE SHARED CONTROL

Muhammed Yusuf Uzun
M.S. in Mechanical Engineering
Advisor: Yıldıray Yıldız
August 2025

This thesis presents novel control architectures for shared control systems, where human operator and the automation share the control responsibility. Proposed architectures are formally verified by mathematical methods, and shown to improve performance/reduce workload in experiments and simulations for flight control scenarios. Firstly, with use of an adaptive human operator model and Long Short-Term Memory (LSTM) network, a neural network based pilot assistance system is proposed and shown to improve tracking performance and reduce workload in simulations. Then, this architecture is improved with a more advanced pilot model and a modulation mechanism, such that it does not violate pilot autonomy in case of a malfunction or a disagreement between the pilot and the assistance signal. Lastly, a general, automation-agnostic architecture for automation assistance regulation for shared control systems is proposed. This architecture employs control barrier functions (CBFs) to modulate the automation assistance, where shared control metrics are specified in terms of barrier functions. Finally, results are verified with simulations and human-in-the-loop experiments.

Keywords: Shared control, adaptive control, human in-the-loop.

ÖZET

DAYANIKLI İNSAN-OTONOMİ İŞ BİRLİĞİ: ÖĞRENMEYE DAYALI YARDIMCI SİSTEMLER VE GÜVENLİ PAYLAŞIMLI KONTROL

Muhammed Yusuf Uzun
Makine Mühendisliği, Yüksek Lisans
Tez Danışmanı: Yıldıray Yıldız
Ağustos 2025

Bu tez, insan operatör ile otomasyonun kontrol sorumluluğunu paylaştığı paylaşımlı kontrol sistemleri için yeni kontrol mimarileri sunmaktadır. Önerilen mimariler, matematiksel yöntemlerle biçimsel olarak doğrulanmış ve uçuş kontrol senaryoları için yapılan deneyler ve benzetimlerde performansı artırdığı / iş yükünü azalttığı gösterilmiştir. İlk olarak, uyarlanabilir bir insan operatör modeli ve Uzun Kısa Süreli Bellek (LSTM) ağı kullanılarak, yapay sinir ağı tabanlı bir pilot yardım sistemi önerilmiş ve bu sistemin benzetimlerde izleme performansını artırdığı ve iş yükünü azalttığı gösterilmiştir. Daha sonra, bu mimari daha gelişmiş bir pilot modeli ve bir modülasyon mekanizması ile geliştirilmiştir, böylece bir arıza ya da pilot ile yardımcı sistem arasında bir anlaşmazlık durumunda pilot özerkliği ihlal edilmemektedir. Son olarak, paylaşımlı kontrol sistemleri için otomasyon yardımı düzenlemesine yönelik genel ve otomasyondan bağımsız bir mimari önerilmektedir. Bu mimari, otomasyon yardımını modüle etmek için kontrol bariyer fonksiyonlarını (CBF) kullanmakta, burada paylaşımlı kontrol metrikleri bariyer fonksiyonları cinsinden tanımlanmaktadır. Son olarak, sonuçlar benzetimlerle ve insanlı deneylerle doğrulanmıştır.

Anahtar sözcükler: Paylaşımlı kontrol, uyarlamalı kontrol, döngüde insan.

Acknowledgement

I would like to sincerely thank my advisor, Dr. Yıldıray Yıldız, for his invaluable guidance, patience, and consistent support throughout my academic journey. His insight and commitment to research have greatly shaped both my scientific approach and personal growth.

I am also deeply grateful to my family for their unconditional love, understanding, and the many sacrifices that made this work possible. Their encouragement gave me the strength to persevere through the most challenging times.

I would like to acknowledge my colleagues Emirhan İnanç and Abdullah Habboush, with whom I had the privilege to work and learn. Their collaboration and perspective enriched my research experience and contributed significantly to my development.

I am equally thankful to my friends for their encouragement, laughter, and companionship, which eased the challenges of this journey and made it a memorable experience.

Finally, I would like to thank the Scientific and Technological Research Council of Türkiye (TÜBİTAK) for supporting this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, the capacity of autonomous systems to produce high-performance and computationally efficient control has resulted in an increased pace of integration into safety-critical applications [3]. Nonetheless, when faced with uncertainties or unmodeled dynamics, a purely automated system may degrade in performance or fail completely [4]. Human action in these circumstances has several advantages, including resilience, cognitive flexibility, and adaptive reasoning. This observation of complementarity gave rise to shared control architectures, implemented to delegate authority to perform tasks from both the human and automation in a context aware manner while also being safety aware [5, 6, 7]. These frameworks need to be designed to maintain the performance and safety of the system while also maintaining the authority of the human operator especially under safety critical environments [8]. This requires reliable arbitration mechanisms that can dynamically distribute the control authority between human operators and automation, based on considerations of safety, workload, and performance [3, 9, 10].

As with any system of this type, the major challenge as to how to model and predict human input behavior. In flight control applications, control-theoretic pilot models have historically provided insight into human-in-the-loop behavior such as the crossover model [11, 12]. More recently, adaptive pilot models are introduced, in which model parameters are updated based on the feedback from controlled systems [13, 14, 15].

Meanwhile, neural networks (NNs) have been applied to control with increasing scope and interest [16, 17]. Long Short-Term Memory (LSTM) networks, in particular, are attractive because they allow modeling of temporal dependencies. Recent research has demonstrated the capability to include LSTMs in adaptive control formulations and accelerate adaptation transients[18]. In this context, adaptive pilot models are used to generate consistent and realistic synthetic training data, enabling LSTM training to assist human operators [1].

In shared control applications, where humans and automation collaborate to achieve a given task, safety is a hard constraint. There has been an extensive literature around the use of Control Barrier Functions (CBFs) to apply real-time safety constraints to automated systems using quadratic programming formulations [19]. However, he literature has considerably focused on safety in terms of constraining the behavior of the physical plant. In shared control applications, CBFs have mainly been used to correct human operators for not driving the systems into unsafe regions [20, 21, 22], or focused on ensuring that humans are not harmed by the automation, treating them as passive entities [23, 24].

This thesis builds upon and extends these developments by addressing fundamental questions in shared control through the following contributions. (1) We present a new automation assistance system that uses an LSTM network trained on data produced by an adaptive pilot model. Whereas other systems and frameworks combine LSTM with adaptive controllers that simply pursue human commands [15], the proposed system utilizes predictions about the operator's error and proactively compensates for it, improving performance and decreasing operator workload, as illustrated with simulation results. (2) We introduce a novel arbitration mechanism to the work mentioned in (1), which modulates the LSTM's assistance based on the correlation between the pilot's intended effect and plant output influenced by assistance. This allows for dynamic blending of the intended input while maintaining operator authority, even in the event of assistance malfunction or erroneous estimation of operator intention [25]. (3) We describe the shared control objectives as inequality constraints in the context of a Control Barrier Function (CBF)-based framework. This framework allows for automation intervention to be treated as a constrained optimization problem, using human-centered metrics such as attention, trust, and workload, as constrained variables in the control formulation. Many previous efforts examining arbitration methods for integrating automation and human variables depend on complex functions that are system-specific [26, 27], whereas the framework presented is modular and automation agnostic. We are able to incorporate an efficient and interpretable method of fusing automation assistance with human control input, while preserving safety and compliance for shared control requirements.

Together, these contributions advance the state of the art in shared control and human-autonomy collaboration. They establish mathematically grounded and experimentally validated foundations for designing future systems, where humans and automation co-exist and collaborate.

Chapter 2

Enhancing Human Operator Performance with Long Short-Term Memory Networks in Adaptively Controlled Systems

2.1 Introduction

Automation provides more accurate results, faster response times, and allows for complicated computations [3]. Nonetheless, fully autonomous systems usually focus on certain applications, which sometimes results in a failure due to uncertainty or fault [4]. Therefore, a more prescriptive and safer approach, including a human-automation collaboration that constitutes a cyber-physical human system, seems more acceptable compared to an automation system with no human interventions [28]. There are a variety of different collaborations that exist with human and automation. Such as, the automated system assists the human operator [5], the human operator oversees the automated system [7], or a combination of these situations [6].

The use of neural networks in automation systems has grown exponentially in recent years [16, 17]. Similarly, research which considers human operator interactions with control systems using neural networks has also developed. A previous study proposes using Long Short-Term Memory (LSTM) networks with adaptive control systems since they may effectively capture dependencies in input sequences, which would help predict and counteract adaptation errors [18]. A recent work considered how the framework in [18] performs in an overall human-automation system, where the human operator commands references to the control system [29].

A large amount of data in required to train neural networks (NNs) in closed-loop control systems where human interactions are present. For such purpose, a human operator model may be employed to facilitate this process, taking advantage of modeling operator behavior to train and subsequently test the NN for various scenarios. One possible method for developing human operator models for flight control applications is based on the crossover model, derived from the premise that human pilots shape their responses, thereby allowing the overall system to behave similar to an appropriately designed feedback system [11]. More recently, motivated by this notion, adaptive pilot models are developed for achieving a behavior very similar to the crossover model [13, 14].

In this chapter we examine a research gap in the control systems literature: what is a mathematically rigorous way to develop a control framework focused on enhancing the performance of, and assisting the human operator in an adaptively controlled system? To achieve such aim, we construct a LSTM network that predicts and compensates for the errors attributed to the human operator. The adaptive pilot model [14] serves as the mathematical representation of the human operator in our development. It is important to emphasize that the proposed control framework is functioning as an assistance system to support/improve human pilot performance while reducing the pilot's cognitive workload, whereas in our previous work [29] the focus is merely on following the human pilot's command.

Simulations indicate a clear improvement in tracking accuracy of the human

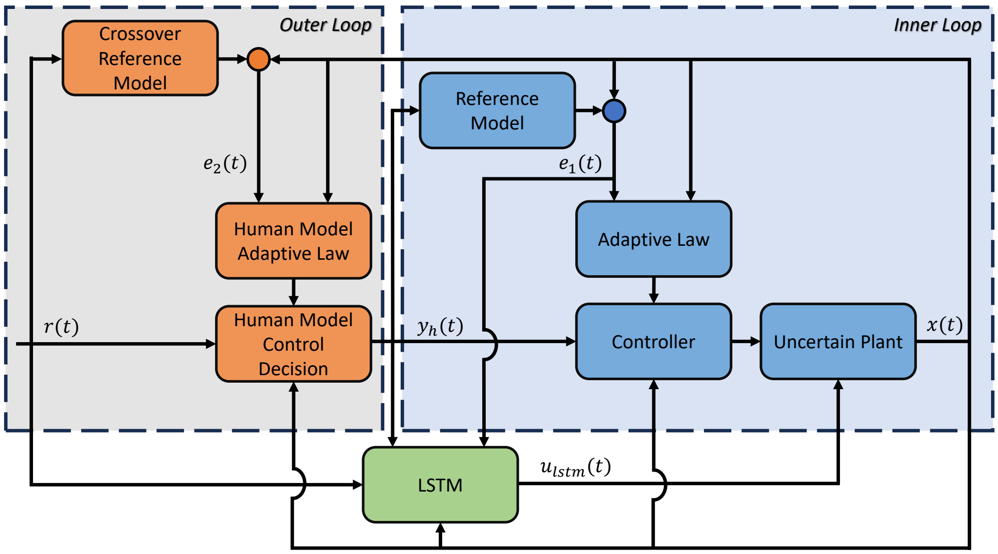

Figure 2.1: Proposed control architecture [1].

operator with the use of proposed method with various uncertainties present. Additionally, it was shown that the pilot workload is reduced as well.

2.2 Design Approach

Figure 2.1 depicts the block diagram of the overall closed-loop control system. The outer loop is controlled by a human operator that attempts to make the plant output follow the reference input. The command given by the human operator, $y_h(t)$, is fed as a reference input to the inner loop, which is controlled by an adaptive controller. To help support the human operator's performance, we construct an LSTM network trained to minimize the error between the human operator's objective and the plant output. The human operator acts as an outer loop adaptive controller and pushes the actual plant states to match the crossover-reference model states, which resembles the real pilot behavior [11].

2.2.1 Inner Loop

We begin by considering the plant dynamics subject to uncertainty

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t) + B_p W^T \sigma\left(x_p(t)\right), \tag{2.1a}$$

$$y_1(t) = C_1^T x_p(t), y_2(t) = C_2^T x_p(t), \tag{2.1b}$$

where $x_p(t) \in \mathbb{R}^{n_p}$ denotes the accessible state vector, and $u_p(t) \in \mathbb{R}^m$ is the plant input. The term $\sigma\left(x_p(t)\right) : \mathbb{R}^{n_p} \to \mathbb{R}^s$ represents a known, possibly nonlinear, basis function. The matrix $W \in \mathbb{R}^{s\times m}$ is unknown. Matrices $A_p \in \mathbb{R}^{n_p\times n_p}$ and $B_p \in \mathbb{R}^{n_p\times m}$ are known and define the system dynamics. The output matrices $C_1 \in \mathbb{R}^{n_p\times m}$ and $C_2 \in \mathbb{R}^{n_p\times n_r}$ are also known. The output signals $y_1(t) \in \mathbb{R}^m$ and $y_2(t) \in \mathbb{R}^{n_r}$ are of interest for the inner and outer control loops, respectively. Additionally, we assume that the pair (A_p, B_p) is controllable. The nominal plant dynamics are expressed as

$$\dot{x}_n(t) = A_p x_n(t) + B_p u_n(t). \tag{2.2}$$

The nominal control input $u_n(t) \in \mathbb{R}^m$ is defined by

$$u_n(t) = -L_x x_n(t) + L_{\mathrm{r}} y_h(t), \tag{2.3}$$

where $y_h(t) \in \mathbb{R}^m$ is the human command (Fig. 2.1). The gain matrix $L_x \in \mathbb{R}^{m\times n_p}$ is chosen such that $A_r \triangleq A_p - B_p L_x$ is Hurwitz. Furthermore, due to the physical limitations of the manipulator, the human input $y_h(t)$ is constrained. These input constraints are explicitly addressed in the stability analysis of the outer loop, as detailed in the next section. Defining $B_r \triangleq B_p L_{\mathrm{r}}$, the corresponding reference model is described as

$$\dot{x}_r(t) = A_r x_r(t) + B_r y_h(t). \tag{2.4}$$

When y_h is constant, the steady-state of the reference model satisfies $x_r(\infty) = -A_r^{-1} B_p L_{\mathrm{r}} y_h$. Consequently, if $x_p(t)$ converges to $x_r(t)$, then from (2.1b) we find that $y_1(\infty) = -C_1^T A_r^{-1} B_p L_{\mathrm{r}} y_h$. To ensure that $\lim_{t\to\infty} y_1(t) = y_h$, we select

$$L_{\mathrm{r}} = -\left(C_1^T A_r^{-1} B_p\right)^{-1}, \tag{2.5}$$

The control input is then defined as

$$u_p(t) = -L_x x_p(t) + L_{\mathrm{r}} y_h(t) - \hat{W}^T(t)\sigma\left(x_p(t)\right) + u_{\mathrm{lstm}}(t), \tag{2.6}$$

where $\hat{W}(t) \in \mathbb{R}^{s \times m}$ is the adaptive weight estimate, and $u_{\text{lstm}}(t)$ denotes the output of the LSTM network, whose details are discussed in the following sections. Substituting (2.6) into (2.1), the system dynamics become

$$\dot{x}_p(t) = A_r x_p(t) + B_r y_h(t) - B_p \tilde{W}^T(t) \sigma\left(x_p(t)\right) + B_p u_{\text{lstm}}(t), \tag{2.7}$$

where the weight error is defined as $\tilde{W}(t) \triangleq \hat{W}(t) - W$.

Subtracting (2.4) from (2.7) yields

$$\dot{e}_1(t) = A_r e_1(t) - B_p \tilde{W}^T(t) \sigma\left(x_p(t)\right) + B_p u_{\text{lstm}}(t), \tag{2.8}$$

with $e_1(t) \triangleq x_p(t) - x_r(t)$ representing the inner loop tracking error. The adaptive law for updating $\hat{W}(t)$ is given by

$$\dot{\hat{W}}(t) = \gamma_W \sigma\left(x_p(t)\right) e_1(t)^T P_1 B_p - \gamma_W \kappa \left\|e_1\right\| \hat{W}, \tag{2.9}$$

where $\gamma_W \in \mathbb{R}_+$ is the learning rate, and $P_1 \in \mathbb{R}_+^{n_p \times n_p} \cap \mathbb{S}^{n_p \times n_p}$ is the solution of Lyapunov equation

$$A_r^T P_1 + P_1 A_r = -Q_1, \tag{2.10}$$

for a symmetric positive definite matrix $Q_1 \in \mathbb{R}_+^{n_p \times n_p} \cap \mathbb{S}^{n_p \times n_p}$ and $\kappa \in \mathbb{R}_+$ is a design parameter.

Lemma 2.1. *Consider the uncertain system described by (2.1), the reference model (2.4), and the feedback control law defined by (2.6) together with the adaptive update law in (2.9). Then, the trajectories* $\left(e_1(t), \tilde{W}(t)\right)$ *are uniformly ultimately bounded and converge to a compact set that is predefined.*

For clarity, the time-dependence notation "(t)" is omitted where appropriate throughout the proofs.

Proof: Consider the following Lyapunov function candidate:

$$V_1 = e_1^T P_1 e_1 + \gamma_W^{-1} \operatorname{Tr}\left\{\tilde{W}^T \tilde{W}\right\}. \tag{2.11}$$

Differentiating V_1 along the trajectories of (2.8) and (2.9) gives

$$\begin{aligned}\dot{V}_1 =& \dot{e}_1^T P_1 e_1 + e_1^T P_1 \dot{e}_1 + 2\gamma_W^{-1} \operatorname{Tr}\left\{\tilde{W}^T \dot{\tilde{W}}\right\} \\ =& - e_1^T Q_1 e_1 + 2\gamma_W^{-1} \operatorname{Tr}\left\{\tilde{W}^T \dot{\tilde{W}}\right\} \\ & - 2 \operatorname{Tr}\left\{\tilde{W}^T \sigma\left(x_p\right) e_1^T P_1 B_p\right\} \\ & + 2 \operatorname{Tr}\left\{e_1^T P_1 B_p u_{\text{lstm}}\right\} \\ =& - e_1^T Q_1 e_1 + 2 e_1^T P_1 B_p u_{\text{lstm}} \\ & - 2\kappa \left\|e_1\right\| \operatorname{Tr}\left\{\tilde{W}^T \tilde{W}\right\} - 2\kappa \left\|e_1\right\| \operatorname{Tr}\left\{\tilde{W}^T W\right\}. \end{aligned} \tag{2.12}$$

This leads to the inequality

$$\begin{aligned}\dot{V}_1 \leq & - \lambda_{\min}(Q) \left\|e_1\right\|^2 + 2 \left\|e_1\right\| \left\|P_1 B_p\right\| \left\|\bar{u}_{\text{lstm}}\right\| \\ & - 2\kappa \left\|e_1\right\| \left\|\tilde{W}\right\|_F^2 + 2\kappa \left\|e_1\right\| \left\|\tilde{W}\right\|_F^2 \left\|W\right\|_F \\ = & - \left\|e_1\right\| \Bigg(\lambda_{\min}(Q) \left\|e_1\right\| + 2\kappa \Big(\left\|\tilde{W}\right\|_F - \frac{\left\|W\right\|_F}{2}\Big)^2 \\ & \qquad - \kappa \frac{\left\|W\right\|_F^2}{2} - 2 \left\|P_1 B_p\right\| \bar{u}_{\text{lstm}}\Bigg), \end{aligned} \tag{2.13}$$

where $\bar{u}_{\text{lstm}}$ denotes an upper bound on u_{lstm} , as established in [18].

Remark 2.1. *The LSTM output is bounded due to both the saturation effects of the activation functions and the fixed nature of the network parameters obtained through offline training.*

Therefore, $\dot{V}_1 \leq 0$ holds if either

$$\left\|e_1\right\| > \frac{\kappa \left\|W\right\|_F^2 / 2 + 2 \left\|P_1 B_p\right\| \bar{u}_{\text{lstm}}}{\lambda_{\min}(Q)} \triangleq b_e, \tag{2.14}$$

or

$$\left\|\tilde{W}\right\|_F > \frac{\left\|W\right\|_F^2}{2} + \sqrt{\frac{\left\|W\right\|_F^2}{4} + \frac{\left\|P_1 B_p\right\| \bar{u}_{\text{lstm}}}{\kappa}} \triangleq b_w. \tag{2.15}$$

Thus, the pair $(e_1, \tilde{W})$ converges to the compact set E_p, which is uniformly ultimately bounded [30], and defined by $E_p \triangleq \left\{(e_1, \tilde{W}) : \left\|e_1\right\| \leq b_e \text{ and } \left\|\tilde{W}\right\|_F \leq b_w\right\}$.
■

2.2.2 Outer Loop

We define the control deficiency arising from saturation of the human input as $\Delta y(t) \triangleq y_h(t) - v(t)$, where $v(t)$ denotes the unsaturated human control input. Using this, we define the integrator dynamics as

$$\dot{x}_c(t) = y_2(t) - r(t) + M\Delta y(t), \tag{2.16}$$

where $x_c(t) \in \mathbb{R}^{n_r}$ is the integrator state, $r(t) \in \mathbb{R}^{n_r}$ is a bounded reference signal, and $M \in \mathbb{R}^{n_r \times m}$ is a control gain. The additional term $M\Delta y(t)$, which will be discussed in more detail later in this section, serves to mitigate integrator windup and becomes active only when human input saturation occurs [31].

By combining (2.7) with (2.16), we arrive at the augmented system dynamics

$$\begin{aligned}\dot{x}(t) =& Ax(t) + B_m r(t) - B_m M\Delta y(t) + BH^T(t)\sigma\left(x_p(t)\right) \\ &+ BL_{\mathrm{r}} y_h(t) + Bu_{\mathrm{lstm}}(t),\end{aligned} \tag{2.17}$$

where the augmented state is defined as $x(t) \triangleq \left[x_p^T(t), x_c^T(t)\right]^T \in \mathbb{R}^n$ with $n = n_p + n_r$, and $H^T(t) \triangleq -\tilde{W}^T(t)$ (refer to (2.7)) is an unknown time-varying parameter. The matrices are defined as $A \triangleq [A_r \ 0_{n_p \times n_r} \ ; \ C_2^T \ 0_{n_r \times n_r}] \in \mathbb{R}^{n \times n}$, $B \triangleq [B_p \ ; \ 0_{n_r \times m}] \in \mathbb{R}^{n \times m}$, $B_m \triangleq [0_{n_p \times n_r} \ ; \ -I_{n_r \times n_r}] \in \mathbb{R}^{n \times n_r}$. It is also assumed that the pair $(A, \ BL_{\mathrm{r}})$ is controllable.

The human operator aims to guide the plant states to track those of the crossover-reference model [11], defined as

$$\dot{x}_m(t) = A_m x_m(t) + B_m r(t), \tag{2.18}$$

where $x_m(t) \in \mathbb{R}^n$ denotes states, and $A_m \in \mathbb{R}^{n \times n}$ is a Hurwitz matrix. This model characterizes a closed-loop system with sufficiently large stability margins [11].

In the absence of saturation on the human input, i.e., when $y_h(t) = v(t)$, and assuming $H(t)$ is known, an ideal human control law

$$v^*(t) = -\theta_x x(t) - L_{\mathrm{r}}^{-1} H^T(t)\sigma\left(x_p(t)\right), \tag{2.19}$$

with $\theta_x \in \mathbb{R}^{m\times n}$ chosen such that $A_m = A - BL_{\mathrm{r}}\theta_x$, results in closed-loop dynamics that replicate the crossover model.

Remark 2.2. *By partitioning θ_x as*

$$\theta_x \triangleq [\phi \quad \psi] \in \mathbb{R}^{m\times n}, \tag{2.20}$$

with $\phi \in \mathbb{R}^{m\times n_p}$ and $\psi \in \mathbb{R}^{m\times n_r}$, the matrix A_m can be chosen as

$$A_m \triangleq \begin{bmatrix} A_r - B_r\phi & -B_r\psi \\ C_2^T & 0_{n_r\times n_r} \end{bmatrix} \in \mathbb{R}^{n\times n}, \tag{2.21}$$

where A_r and B_r are the reference model matrices for the inner loop defined in (2.4), ensuring that $A_m = A - BL_r\theta_x$. This partitioning permits expressing $\theta_x x = \phi x_p + \psi x_c$, where x_p and x_c are the plant and integrator states introduced following (2.17). This form is used for the stability analysis discussed later in this section.

To address the unknown term $H(t)$, the human control input is defined as

$$v(t) = -\theta_x x(t) - L_{\mathrm{r}}^{-1}\hat{H}^T(t)\sigma\left(x_p(t)\right), \tag{2.22a}$$

$$y_{h_i}(t) = \begin{cases} v_i(t), & \text{if } |v_i(t)| \le y_{o_i}, \\ y_{o_i}\,\mathrm{sgn}\left(v_i(t)\right), & \text{if } |v_i(t)| > y_{o_i}, \end{cases} \tag{2.22b}$$

where $\hat{H}(t)$ denotes the estimate of the unknown matrix $H(t)$. Equation (2.22b) defines an element-wise saturation function, with $y_{o_i} \in \mathbb{R}_+$ representing the saturation bound for the i^{th} component of $y_h(t)$.

By substituting (2.22) into the augmented dynamics in (2.17), we obtain

$$\begin{aligned} \dot{x}(t) =& A_m x(t) + B_m r(t) - B_m M\Delta y(t) + BL_{\mathrm{r}}\Delta y(t) \\ &- B\tilde{H}^T(t)\sigma\left(x_p(t)\right) + Bu_{\mathrm{lstm}}(t), \end{aligned} \tag{2.23}$$

where $\tilde{H}(t) \triangleq \hat{H}(t) - H(t)$ denotes the outer loop adaptive parameter error. Subtracting the crossover model dynamics in (2.18) from (2.23) yields the outer loop tracking error dynamics

$$\begin{aligned} \dot{e}_2(t) =& A_m e_2(t) - B_m M\Delta y(t) + BL_{\mathrm{r}}\Delta y(t) \\ &- B\tilde{H}^T(t)\sigma\left(x_p(t)\right) + Bu_{\mathrm{lstm}}(t), \end{aligned} \tag{2.24}$$

where the outer loop tracking error is defined as $e_2(t) \triangleq x(t) - x_m(t)$.

To handle the disturbance introduced by $\Delta y(t)$, we define an auxiliary signal $e_\Delta(t)$ following [32] as

$$\dot{e}_\Delta(t) = A_m e_\Delta(t) - B_m M \Delta y(t) + B L_{\mathrm{r}} \Delta y(t), \tag{2.25a}$$

$$e_\Delta(t_0) = 0. \tag{2.25b}$$

Introducing an augmented error signal $e_y(t) \triangleq e_2(t) - e_\Delta(t)$ and subtracting (2.25) from (2.24) results in following error dynamics

$$\dot{e}_y(t) = A_m e_y(t) - B\tilde{H}^T(t)\sigma(x_p(t)) + B u_{\mathrm{lstm}}(t). \tag{2.26}$$

To update the estimate $H(t)$, we propose the following adaptive law

$$\dot{\hat{H}}(t) = \gamma_H \operatorname{Proj}\left(\hat{H}(t), \sigma(x_p(t))\, e_y^T(t) P_2 B\right), \tag{2.27}$$

where $\gamma_H \in \mathbb{R}_+$ is the learning rate, and $P_2 \in \mathbb{R}_+^{n\times n} \cap \mathbb{S}^{n\times n}$ is the solution of

$$A_m^T P_2 + P_2 A_m = -Q_2, \tag{2.28}$$

with $Q_2 \in \mathbb{R}_+^{n\times n} \cap \mathbb{S}^{n\times n}$.

Remark 2.3. *From Lemma 2.1, it follows that both $\tilde{W}(t)$ and $\dot{\tilde{W}}(t)$ are bounded. Given that $H(t) = -\tilde{W}^T$, this implies that $H(t)$ and $\dot{H}(t)$ are also bounded. Hence, there exist constants $h \in \mathbb{R}_+$ and $\bar{h} \in \mathbb{R}_+$ such that $\|H(t)\| \leq h$ and $\|\dot{H}(t)\| \leq \bar{h}$ for all $t \geq 0$.*

Assumption 2.1. *Consider the matrix ψ defined in (2.20). The matrix M in (2.16) can be chosen such that $M\psi \in \mathbb{R}^{n_r \times n_r}$ is Hurwitz.*

If this assumption is violated, it would imply that ψ is column-rank-deficient. However, since ψ is a design parameter, it is always possible to construct a full column rank approximation $\psi_f = \psi + \epsilon_1$, where ϵ_1 is a matrix with infinitesimally small entries. This leads to the perturbed system matrix $A - BL_r[\phi \quad \psi_f] = A_m + \epsilon_2$, where ϵ_2 is also infinitesimal and depends on ϵ_1. As discussed in Remark 2.2, $A_m + \epsilon_2$ remains Hurwitz, thereby justifying the validity of Assumption 2.1.

Theorem 2.1. *Consider the uncertain dynamical system given in (2.1), the adaptive controller defined by (2.4), (2.6), and (2.9), and the adaptive pilot model specified by (2.16), (2.18), (2.22), and (2.27). Then, the trajectories* $\left(e_y(t), \tilde{H}(t)\right)$ *remain bounded for all* $t \geq 0$ *and converge to the compact set*

$$E \triangleq \left\{ \left(e_y(t), \tilde{H}(t)\right) : \|e_y(t)\| \leq \eta,\ \|\tilde{H}(t)\| \leq \tilde{h} \right\}, \tag{2.29}$$

where,

$$\eta \triangleq \frac{\|P_2 B\| \, \bar{u}_{lstm} + \sqrt{\|P_2 B\|^2 \, u_{lstm}^2 + 2\lambda_{\min}(Q_2)\gamma_H^{-1}\tilde{h}\bar{h}}}{\lambda_{\min}(Q_2)}, \tag{2.30}$$

and $\tilde{h} \triangleq |\hat{H}\max| + h$, $\bar{u}lstm$ *is the upper bound of the LSTM output, and* $h, \bar{h}$ *are defined in Remark 2.3. Furthermore, all system signals are bounded.*

Proof: Consider the Lyapunov function candidate

$$V_2\left(e_y, \tilde{H}\right) = e_y^T P_2 e_y + \gamma_H^{-1} \operatorname{Tr}\left\{\tilde{H}^T \tilde{H}\right\}. \tag{2.31}$$

Differentiating along the trajectories (2.26) and (2.27) yields

$$\begin{aligned}
\dot{V}_2 =& \dot{e}_y^T P_2 e_y + e_y^T P_2 \dot{e}_y + 2\gamma_H^{-1} \operatorname{Tr}\left\{\tilde{H}^T \dot{\tilde{H}}\right\} \\
=& - e_y^T Q_2 e_y - 2 \operatorname{Tr}\left\{\tilde{H}^T \sigma\left(x_p\right) e_y^T P_2 B\right\} \\
& + 2\gamma_H^{-1} \operatorname{Tr}\left\{\tilde{H}^T \dot{\hat{H}}\right\} - 2\gamma_H^{-1} \operatorname{Tr}\left\{\tilde{H}^T \dot{H}\right\} \\
& + 2 e_y^T P_2 B u_{\mathrm{lstm}}
\end{aligned} \tag{2.32}$$

$$\begin{aligned}
=& - e_y^T Q_2 e_y - 2\gamma_H^{-1} \operatorname{Tr}\left\{\tilde{H}^T \dot{H}\right\} + 2 e_y^T P_2 B u_{\mathrm{lstm}} \\
& + 2 \operatorname{Tr}\left\{\tilde{H}^T \left(\operatorname{Proj}\left(\hat{H}, Y_H\right) - Y_H\right)\right\},
\end{aligned} \tag{2.33}$$

where $Y_H \triangleq \sigma\left(x_p\right) e_y^T P_2 B_p$. Using the projection property

$$\left(\theta_{i,j} - \theta_{i,j}^*\right)\left(\operatorname{Proj}\left(\theta_{i,j}, Y_{i,j}\right) - Y_{i,j}\right) \leq 0, \tag{2.34}$$

and applying Remark 2.3, we obtain the bound

$$\begin{aligned}
\dot{V}_2 &\leq -e_y^T Q_2 e_y - 2\gamma_H^{-1} \operatorname{Tr}\left\{\tilde{H}^T \dot{H}\right\} + 2 e_y^T P_2 B u_{\mathrm{lstm}} \\
&\leq -\lambda_{\min}\left(Q_2\right) \|e_y\|^2 + 2\gamma_H^{-1}\tilde{h}\bar{h} + 2\|e_y\| \, \|P_2 B\| \, \bar{u}_{\mathrm{lstm}}.
\end{aligned} \tag{2.35}$$

This implies that when $\|e_y\| > \eta$ (see 2.30), $\dot{V}_2 \leq 0$, and thus the solution remains within the compact set (2.30). Therefore, the signals $(e_y(t), \hat{H}(t))$ are bounded. By Lemma 2.1, all signals in the inner loop, including $x_p(t)$, are also bounded. As $r(t)$ is bounded, it follows from (2.18) that $x_m(t)$ is also bounded. To complete the proof, we need to establish the boundedness of $x_c(t)$.

We consider two cases: a) when $\Delta y(t) = 0$ and b) when $\Delta y(t) \neq 0$.

Case a) $\Delta y(t) = 0$: From (2.25), $e_\Delta(t)$ is bounded. Since $e_y(t) = e_2(t) - e_\Delta(t)$ is bounded, it follows that $e_2(t)$ is also bounded. As $e_2(t) = x(t) - x_m(t)$ and $x_m(t)$ is bounded, $x(t)$ must also be bounded.

Case b) $\Delta y(t) \neq 0$: Using $\Delta y(t) = y_h(t) - v(t)$, we rewrite (2.16) as

$$\dot{x}_c(t) = C_2^T x_p(t) - r(t) + M y_h(t) - M v(t). \tag{2.36}$$

Substituting (2.22b) and (2.22a) into (2.36), we obtain

$$\dot{x}_c(t) = C_2^T x_p(t) - r(t) + M y_h(t) + M\theta_x x(t) + M L_{\mathrm{r}}^{-1} \hat{H}^T(t) \sigma\left(x_p(t)\right). \tag{2.37}$$

Using (2.20), we express $\theta_x x(t) = \phi x_p(t) + \psi x_c(t)$ and substitute into (2.37) to get

$$\begin{aligned}
\dot{x}_c(t) =& M\psi x_c(t) + w(t), \\
w(t) \triangleq& \left(C_2^T + M\phi\right) x_p(t) - r(t) + M y_h(t) \\
&+ M L_{\mathrm{r}}^{-1} \hat{H}^T(t) \sigma\left(x_p(t)\right).
\end{aligned} \tag{2.38}$$

Since $r(t)$, $\hat{H}(t)$, $y_h(t)$, and $x_p(t)$ are all bounded, it follows that $w(t)$ is bounded. By Assumption 2.1 and the structure of (2.38), we conclude that $x_c(t)$ is bounded. As $x(t) = \left[x_p^T(t), x_c^T(t)\right]^T$, the full state vector $x(t)$ is bounded. Since $x_m(t)$ is bounded and $e_2(t) = x(t) - x_m(t)$, $e_2(t)$ is also bounded. Then, because $e_y(t) = e_2(t) - e_\Delta(t)$ is bounded, it follows that $e_\Delta(t)$ is bounded. Using (2.25), we can conclude that $\Delta y(t)$ is bounded as well. ■

2.3 Special Case: Systems with Free Integrators

The assignment of L_{r} in (2.5) becomes invalid when the matrix $C_1^T A_r^{-1} B_p$ is singular. This situation can arise, for example, when a pilot commands attitude rates through the inner loop to control the vehicle's attitude angles. In such cases, L_r cannot be computed because angular rates with zero steady-state values cannot track a non-zero human command. To address this limitation, we modify the plant dynamics in (2.1) by separating the states as

$$\begin{bmatrix} \dot{x}_s \\ \dot{x}_\theta \end{bmatrix} = \begin{bmatrix} A_s & 0 \\ C_{1s}^T & 0 \end{bmatrix} \begin{bmatrix} x_s \\ x_\theta \end{bmatrix} + \begin{bmatrix} B_s \\ 0 \end{bmatrix} \left(u_p(t) + W^T \sigma(x_s(t))\right),$$
$$y_1(t) = \dot{x}_\theta(t) = C_{1s}^T x_s(t), y_2(t) = x_\theta(t) = C_2^T x_p(t), \tag{2.39}$$

where $x_s(t) \in \mathbb{R}^{n_s}, x_\theta(t) \in \mathbb{R}^m$, and the full plant state is $x_p(t) \triangleq \left[x_s^T, x_\theta^T\right]^T \in \mathbb{R}^{n_p}$. To guarantee the existence of a suitable L_r, the adaptive controller is tasked solely with regulating the $x_s(t)$ dynamics, such that the plant output tracks the human input $y_h(t)$. By applying the control law defined in (2.6) and substituting it into the $x_s(t)$ dynamics in (2.39), we get

$$\dot{x}_s(t) = A_{rs} x_s(t) + B_{rs} y_h(t) - B_s \tilde{W}^T(t) \sigma\left(x_s(t)\right) + B_s u_{\mathrm{lstm}}(t), \tag{2.40}$$

where the pair (A_{rs}, B_{rs}) defines the modified reference model dynamics for the $x_s(t)$ subsystem. By augmenting (2.40) with the $x_\theta(t)$ dynamics, we recover a closed-loop system structure analogous to (2.7), thereby extending the previous results to this special case involving free integrators.

2.4 LSTM Network

We design a Long Short-Term Memory (LSTM) network [33] to assist the pilot in minimizing the outer loop tracking error defined in (2.24). To this end, the LSTM network receives the inner loop tracking error $e_1(t)$ from (2.8), the human input $y_h(t)$ from (2.22), the plant output $y_2(t)$ from (2.1), and the reference signal $r(t)$ from (2.16), as inputs. Based on these inputs, the LSTM generates the control action $u_{\mathrm{lstm}}(t)$ used in (2.24).

2.4.1 Network Structure

The computation of $u_{\text{lstm}}(t)$ in (2.24) involves several steps, beginning with the internal "gate operations" of the LSTM network

$$out_f^n = \sigma_g(W_f e_{norm}^n + R_f h^{n-1} + b_f), \tag{2.41a}$$

$$out_o^n = \sigma_g(W_o e_{norm}^n + R_o h^{n-1} + b_o), \tag{2.41b}$$

$$out_i^n = \sigma_g(W_i e_{norm}^n + R_i h^{n-1} + b_i), \tag{2.41c}$$

$$\bar{c}^n = \sigma_c(W_g e_{norm}^n + R_g h^{n-1} + b_g). \tag{2.41d}$$

Here, e_{norm}^n denotes the normalized version of the input sequence e^n, which consists of sampled instances of the input signal. The variables h^n and c^n represent the hidden and cell states at time step n, respectively. The matrices W and R denote the input and recurrent weights, while b denotes the bias vectors. The gate activation function, sigmoid, is denoted by σ_g, and the state activation function, tanh, is denoted by σ_c. The subscripts f, i, o, and g correspond to the forget gate, input gate, output gate, and cell candidate, respectively. Superscript n refers to the current time step, and $n-1$ to the previous one.

The purpose of these gate operations is to enable the LSTM network to propagate relevant information over extended time horizons by regulating the flow of past and present input data. This allows the network to effectively model long-term dependencies in the input sequence. The cell and hidden states are updated at each time step using

$$c^n = out_f^n \odot c^{n-1} + out_i^n \odot \bar{c}^n, \tag{2.42a}$$

$$h^n = out_o^n \odot \sigma_c(c^n). \tag{2.42b}$$

The LSTM output is then computed using a fully connected layer

$$u_{\text{lstm}}^n = W_{fc} h^n + b_{fc}, \tag{2.43}$$

where W_{fc} and b_{fc} are the weight matrix and bias vector of the output layer. Finally, the continuous-time control input $u_{\text{lstm}}(t)$ is generated from the discrete LSTM output u_{lstm}^n using a zero-order hold (ZOH) mechanism.

2.4.2 Network Training

We train the LSTM network to predict and compensate for the estimation error of the human pilot model. The outer loop tracking error dynamics from (2.24) can be rewritten as

$$\dot{e}_2(t) = A_m e_2(t) - B_m M \Delta y(t) + B\left(L_{\mathrm{r}} \Delta y(t) - \tilde{H}^T(t)\sigma(x_p(t)) + u_{\mathrm{lstm}}(t)\right). \quad (2.44)$$

To enable the LSTM network to cancel the error terms in parentheses, we define the target function, *truth*, to be estimated as $y(t) = \tilde{H}^T(t)\sigma\left(x_p(t)\right) - L_{\mathrm{r}}\Delta y(t)$. Accordingly, the LSTM is trained to estimate this quantity, i.e., $u_{\mathrm{lstm}}(t)$ serves as the estimate $\hat{y}(t)$ of the true signal $y(t)$.

It is important to note that the true signal $y(t)$ is available during the training phase, allowing for supervised learning of the LSTM parameters. During the testing phase, however, $y(t)$ is no longer accessible, and the effectiveness of the LSTM is assessed indirectly through the tracking performance of the overall closed-loop system, which includes the human operator.

2.5 Simulations

In this section, we analyze a flight control task in which the pilot aims to steer the aircraft according to a set of desired attitude angles. To achieve this, the pilot generates angular rate commands that serve as reference inputs for the inner loop controller, which is responsible for tracking these commands.

The plant dynamics (2.1) is taken as a linearized model of a large transport aircraft at Mach 0.3 and altitude 1000 m [14, 34], whose state vector is denoted by $x_s(t) = \begin{bmatrix} \alpha & q & \beta & p & r \end{bmatrix}^T$, while $\dot{x}_\theta(t) = \begin{bmatrix} p & q & r \end{bmatrix}^T$ (see (2.39)). Here, α (rad) represents the angle of attack, β (rad) represents the side-slip angle, and p, q and r (rad/s) correspond to the roll, pitch, and yaw rates, respectively. The

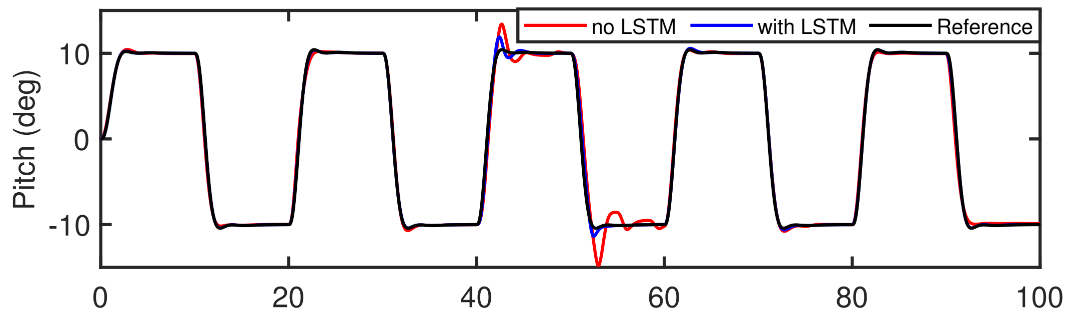

Figure 2.2: Pitch angle tracking performance.

control input vector is defined as $u_p(t) = \begin{bmatrix} \delta_e & \delta_a & \delta_r \end{bmatrix}^T$, which comprises the deflections (rad) of elevators δ_e, ailerons δ_a, and rudders δ_r. The minimum and maximum deflection angles and rates are set to $-23, +17$ deg and $-37, +37$ deg/s for the elevators, $-20, +20$ deg and $-40, +45$ deg/s for the ailerons, and $-25, +25$ deg and $-50, +50$ deg/s for the rudders [35].

The feedback gain L_x in (2.3) is computed using the Linear Quadratic Regulator (LQR) method with the weighting matrices $Q_{\text{LQR}} = \text{diag}\left([1,\ 10,\ 1,\ 10,\ 10]\right)$ and $R_{\text{LQR}} = 100\, I_{3\times3}$. For the inner loop adaptive controller, the Lyapunov matrix in (2.10) is selected as $Q_1 = I_{5\times5}$, the learning rate in (2.9) is set to $\gamma_W = 10$, and the gain κ is chosen as 0.1. To construct the crossover model (2.18), the gain θ_x in Remark 2.2 (see (2.21)) is obtained using an LQR formulation with the cost matrices $Q_{\text{LQR}} = \text{diag}([0_{1\times8},\ 1,\ 1,\ 1])$ and $R_{\text{LQR}} = I_{3\times3}$. This results in the decomposition $\theta_x = [\phi \quad \psi]$, where $\psi = I_{3\times3}$ (see (2.20)). To satisfy Assumption 2.1 and ensure that $M\psi$ is Hurwitz, the matrix M is set to $M = -I_{3\times3}$. For the pilot model, the Lyapunov matrix in (2.28) is selected as $Q_2 = 2\, I_{11\times11}$, and the learning rate in the adaptive law (2.27) is set to $\gamma_H = 10$. The bounds for the projection operator in (2.27) are chosen as ±10 for each element of $\hat{H}(t)$. Finally, the human operator's command is constrained by a saturation limit of $y_{o_i} = 10$ deg/s for $i = 1, \ldots, 3$, as described in (2.22b).

The LSTM network is designed with a single hidden layer consisting of 256 neurons. Since the inner loop error dynamics has five states, and there are three human inputs, three tracked angles, and three reference angles, the input layer contains 14 neurons. The output layer consists of three neurons, corresponding to the three actuators. The LSTM network weights are initialized using the standard Xavier initialization scheme. For training, stochastic gradient descent is used in conjunction with the Adam optimizer, with the following hyperparameters: L2 regularization set to 0.0001, a gradient decay factor of 0.9, and a squared gradient decay factor of 0.999. The simulation time step is chosen as 0.01 s, and the minibatch size is set to 1. The training is performed over 2500 iterations. During training, the elements of the unknown matrix $W \in \mathbb{R}^{s\times m}$ in (2.1) are randomized by assigning values uniformly between -0.5 and 0.5, for each 20 s-long training iteration. The basis functions $\sigma(x)$ used in (2.1) for training are defined as

$$\begin{aligned}
\sigma_{train,1}(x_p) &= [qp, q, 1-\cos(p), r]^T \\
\sigma_{train,2}(x_p) &= [\sin(q), 0.5(q+p), 0.5r, 0.5\sin(p)]^T \\
\sigma_{train,3}(x_p) &= [0.4e^r, r^3, \sqrt{|qp|}, \|[p,q,r]\|]^T \\
\sigma_{train,4}(x_p) &= [0.1qr, 2(q-p), p^2, 1-\cos(q+p)]^T,
\end{aligned} \tag{2.45}$$

each of which is applied sequentially for successive training iterations. Additionally, during training, the reference input $r(t)$ generated by the human operator is randomly varied as a step signal with amplitudes uniformly selected between -10 and 10 degrees.

During testing, the uncertainty matrix W is configured such that its elements are randomly assigned values in the range $[-0.5,\ 0.5]$, and updated every 10 s of simulation time. For testing, the nonlinear function $\sigma(x)$ in (2.1) is defined as

$$\sigma_{\text{test}}(x_p) = \begin{cases} [0.1e^{qr}, 1-\cos(q), \|[p,q,r]\|/2, pr]^T & \text{if } 0 \le t < 20, \\ [r, q(1+r)^2, \sin(q), p/(\cos(q))]^T & \text{if } 20 \le t < 40, \\ [\sin(pq), 2(r-p), \sqrt{|q|}, 1-\cos(r)]^T & \text{if } 40 \le t < 60, \\ [\sqrt[3]{p}, 0.1e^q, 0.2q, r\log(|p|)]^T & \text{if } 60 \le t < 80, \\ [\sin(0.1\sqrt{q}), 0.2\sin(\cos(p)), \|p^2\|, r]^T & \text{if } 80 \le t < 100. \end{cases} \tag{2.46}$$

The loss function used to train the LSTM network is defined as $L(y,\hat{y}) = \frac{1}{N}\sum_{i=0}^{N}(y-\hat{y})^2$, where y denotes the target (or truth) value, $\hat{y}$ is the predicted output of the network, and N is the number of samples in the dataset.

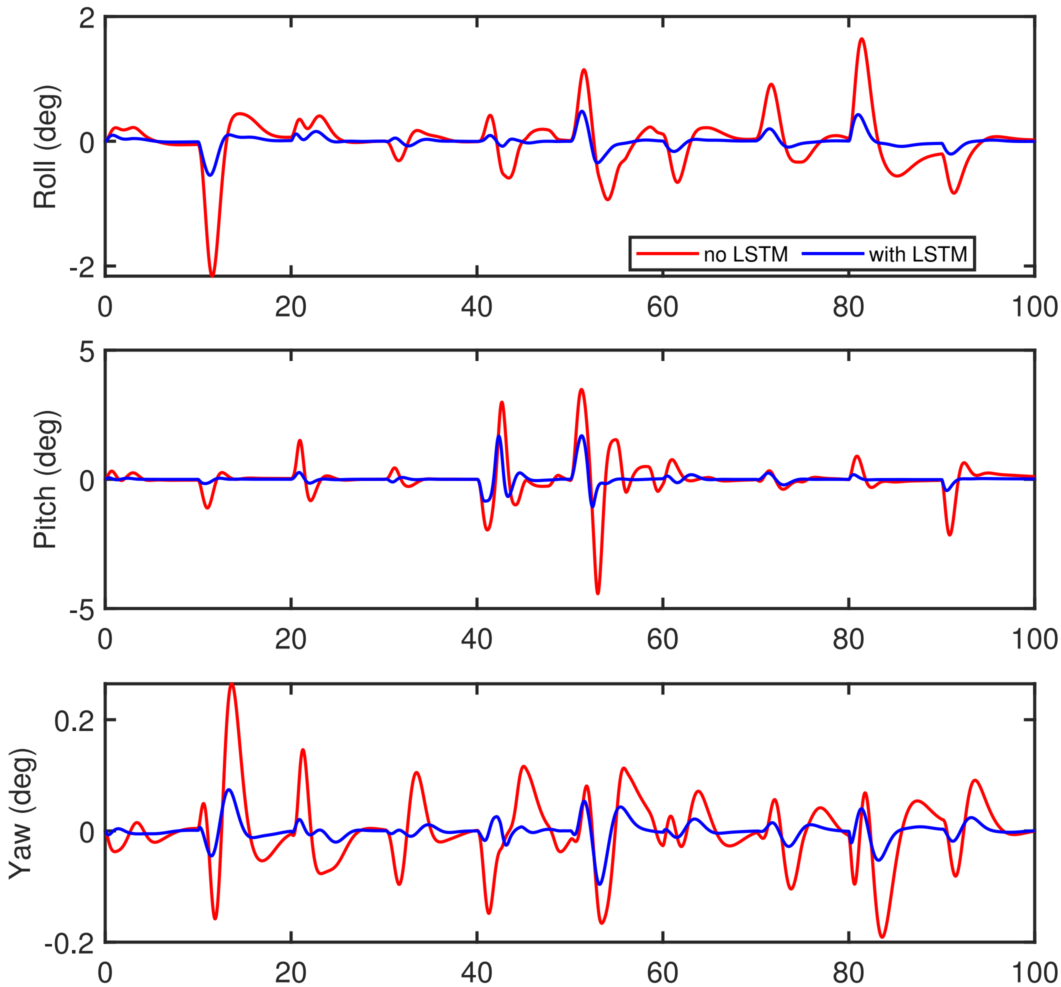

Figure 2.3: Tracking errors of roll, pitch, and yaw angles.

The pilot reference commands consist of zero roll and yaw angles, and a square wave for the pitch angle. The pitch angle tracking performance and corresponding reference tracking errors are shown in Figures 2.2 and 2.3, respectively. A comparison of the tracking errors with and without LSTM augmentation reveals that the inclusion of the LSTM network significantly reduces the tracking error. Figure 2.4 displays the rate commands $y_h(t)$ issued by the pilot to the inner loop in both cases—with and without LSTM assistance. The results indicate that pilot effort is substantially reduced when the LSTM network provides assistance. Finally, Figure 2.5 shows the control surface deflections. It is observed that the total control input magnitude is reduced when the LSTM network is active. This suggests that the system becomes more efficient in handling larger uncertainties with LSTM augmentation.

2.6 Summary

This chapter introduced a control framework based on a Long Short-Term Memory (LSTM) network to enhance the performance of human operators in adaptively controlled systems. The proposed approach supports the human by augmenting their control decisions, thereby enabling more effective task execution. Simulation results from a flight control application demonstrate that the inclusion of the LSTM network improves overall closed-loop system performance and significantly reduces pilot workload.

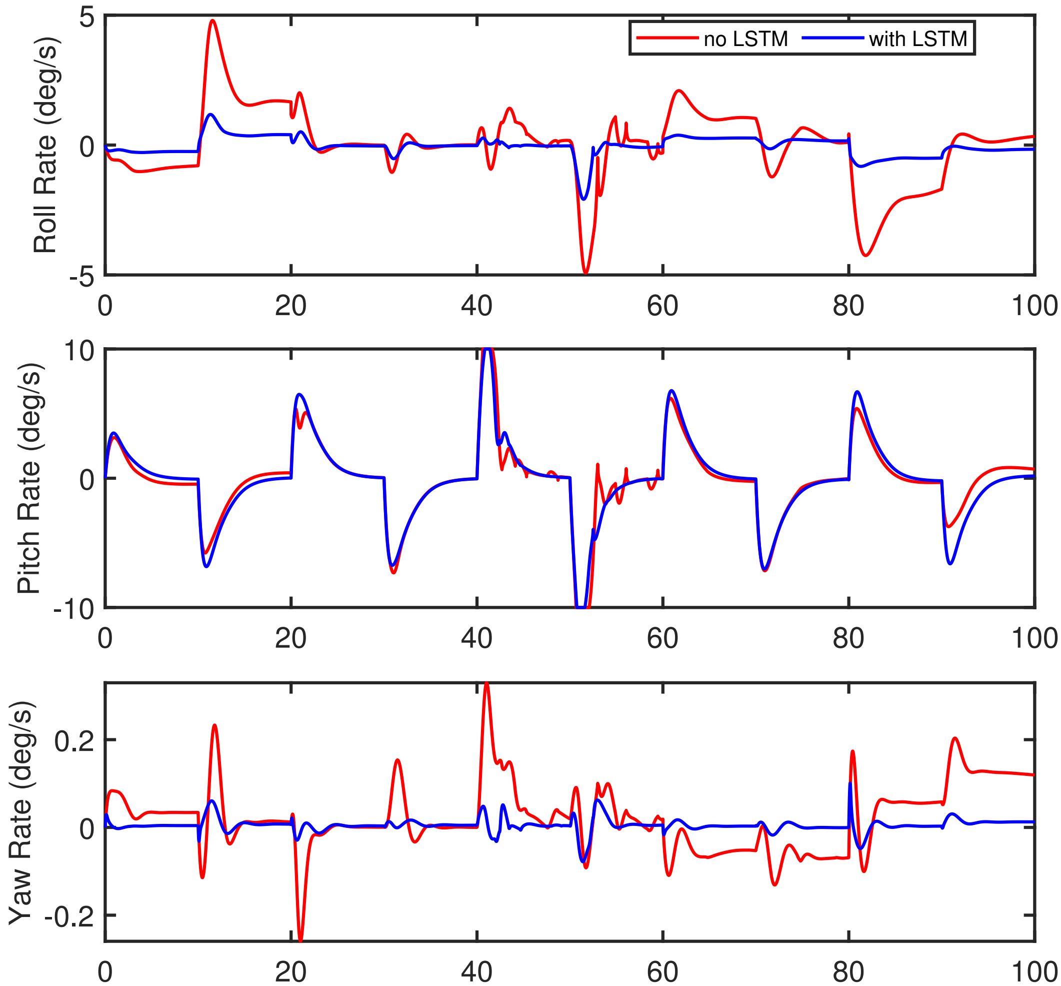

Figure 2.4: Human rate commands for two cases: no LSTM and with LSTM augmentation.

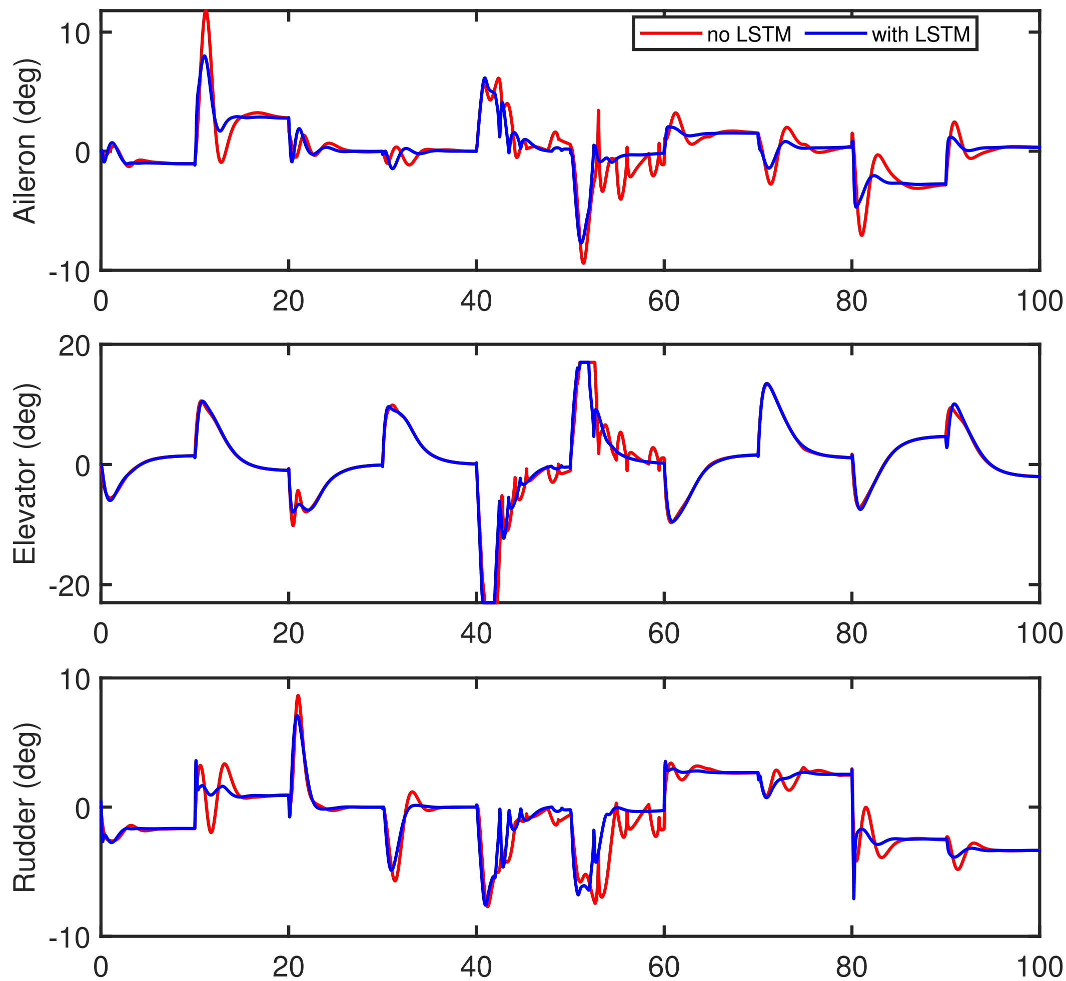

Figure 2.5: Total control input given to the system for two cases: no LSTM and with LSTM augmentation.

Chapter 3

A Robust Human-Autonomy Collaboration Framework with Experimental Validation

It's crucial that automation systems permit human intervention in critical scenarios, ensuring that the system does not override human decision-making [8]. In shared control, there is always a danger of human-machine conflict, and for safety reasons, we want humans to have a higher authority than the machine [3]. To avoid undesired human-machine conflicts, researchers have proposed collaboration schemes based on arbitration mechanisms. These mechanisms adjust the timing and scale of human and automation control inputs, according to the requirements of the system [5, 9, 10].

Accurate modeling of human operators is essential for the development of shared control frameworks, as it enables designers to anticipate pilot responses during the design phase. In the context of flight control, numerous control-theoretic pilot models have been proposed in the literature [12]. In the previous chapter [1], we introduced an adaptive model to simulate pilot behavior operating in collaboration with a LSTM network. This approach demonstrated strong performance in simulation studies, in terms of increased precision and

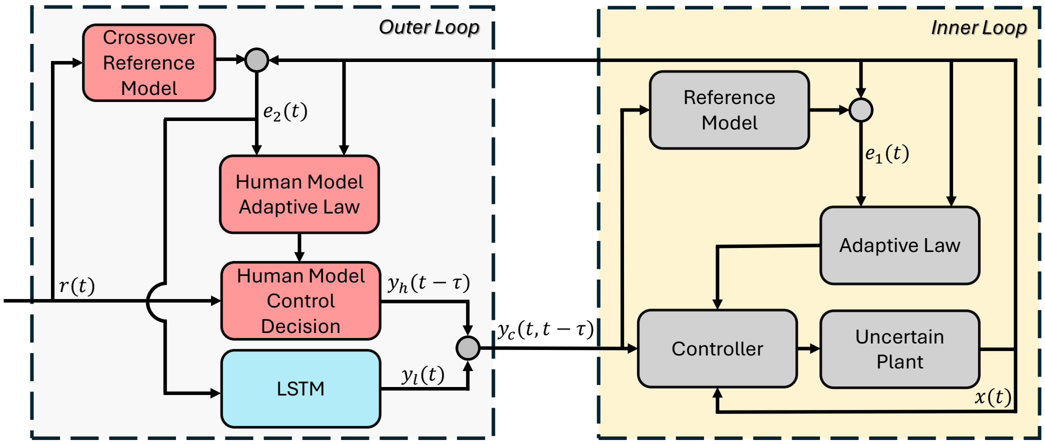

Figure 3.1: Control architecture that has the adaptive pilot model for LSTM network training.

reduced workload. We extend this line of research by introducing a resilient human-autonomy collaboration framework, in which an LSTM network learns from an adaptive pilot model [15] to anticipate and correct operator-induced errors [25]. Unlike conventional approaches, the proposed method remains effective even when the operator's intent is not precisely estimated. This is achieved by scaling the LSTM's contribution based on how strongly the pilot's intentions correlate with the system's overall control effort. To assess the effectiveness of the framework, we conduct human-in-the-loop experiments. The results show clear performance gains in scenarios where operator intent is reliably inferred. Notably, the system also maintains stable behavior when intent estimation is inaccurate, allowing the human operator to retain full control without interference. These findings demonstrate that the framework enhances control performance while safeguarding human authority within the loop.

3.1 Training Setup

To effectively train the LSTM network without relying on extensive experimental trials, it is advantageous to use human behavior models that simulate operator responses. Given that human operators are capable of adapting to dynamic environments, adaptive modeling approaches are particularly well-suited for this purpose. In this work, we employ the adaptive human model proposed by Habboush and Yildiz [15].

The control architecture used during training is illustrated in Figure 3.1, and is structured into inner and outer control loops. The inner loop consists of an adaptive controller that drives the plant states to follow those of a reference model despite the presence of uncertainties. The outer loop contains both the adaptive human model and the LSTM network, which jointly generate rate commands to track a reference trajectory. The LSTM is trained specifically to support the human model's performance, as detailed in Section 3.1.3.

3.1.1 Model Reference Adaptive Controller

The plant dynamics are described by

$$\begin{aligned} \dot{x}_p(t) &= A_p x_p(t) + B_p \Lambda u_p(t), \\ y_1(t) &= C_1^T x_p(t), \;\; y_2(t) = C_2^T x_p(t), \end{aligned} \tag{3.1}$$

where $x_p(t) \in \mathbb{R}^{n_p}$ denotes the accessible state vector, and $u_p(t) \in \mathbb{R}^m$ is the control input. The matrix $\Lambda \in \mathbb{R}_+^{m\times m} \cap \mathbb{D}^{m\times m}$ represents an unknown diagonal control effectiveness matrix, with elements $\lambda_{i,i} \in (0,1]$. The system matrix $A_p \in \mathbb{R}^{n_p\times n_p}$ is unknown, while the input matrix $B_p \in \mathbb{R}^{n_p\times m}$ is known. It is assumed that the pair $(A_p, B_p\Lambda)$ is controllable. In addition, $C_1 \in \mathbb{R}^{n_p\times m}$ and $C_2 \in \mathbb{R}^{n_p\times m}$ are known output matrices, and the outputs $y_1(t) \in \mathbb{R}^m$ and $y_2(t) \in \mathbb{R}^m$ correspond to the inner and outer control loops, respectively.

The inner loop adaptive controller is designed to ensure that the plant state

$x_p(t)$ tracks the reference model state $x_r(t) \in \mathbb{R}^{n_p}$, defined by

$$\dot{x}_r(t) = A_r x_r(t) + B_r y_c(t, t-\tau), \;\; x_r(t_0) = 0, \tag{3.2}$$

where $A_r \in \mathbb{R}^{n_p \times n_p}$ is a Hurwitz matrix, $B_r \in \mathbb{R}^{n_p \times m}$, and $y_c(t, t-\tau) \in \mathbb{R}^m$ is the input command to the inner loop, defined as

$$y_c(t, t-\tau) = y_h(t-\tau) + y_l(t), \tag{3.3}$$

with $y_h \in \mathbb{R}^m$ representing the delayed human command and $\tau \in \mathbb{R}^+$ denoting the input delay. The term $y_l \in \mathbb{R}^m$ corresponds to the command generated by the LSTM network.

To achieve reference state tracking, the nominal dynamics and control input are defined as

$$\dot{x}_n(t) = A_n x_n(t) + B_p u_n(t), \tag{3.4a}$$

$$u_n(t) = -L_x x_n(t) + L_r y_c(t, t-\tau), \tag{3.4b}$$

where $L_x \in \mathbb{R}^{m \times n_p}$ is selected such that $A_r = A_n - B_p L_x$, and $L_r \in \mathbb{R}^{m \times m}$ is chosen to satisfy $\lim_{t\to\infty} y_1(t) = y_c$. At steady state, the reference model (3.2) gives $x_r(\infty) = -A_r^{-1} B_p L_r y_c$. This implies that, once reference tracking is achieved, i.e., $\lim_{t\to\infty} x_p(t) = x_r(t)$, the plant output becomes $y_1(\infty) = -C_1^T A_r^{-1} B_p L_r y_c$ (see (3.1)). Therefore, the feedforward gain is selected as

$$L_r = -(C_1^T A_r^{-1} B_p)^{-1}. \tag{3.5}$$

Considering (3.1), we assume the existence of ideal, unknown control gains $K_x^* \in \mathbb{R}^{m \times n_p}$ and $K_r^* \in \mathbb{R}^{m \times m}$ such that the following matching conditions hold

$$A_p - B_p \Lambda K_x^* = A_r, \quad B_p \Lambda K_r^* = B_r. \tag{3.6}$$

Since Λ is a diagonal positive-definite matrix, its inverse can be expressed as $\operatorname{diag}(\lambda^*) = \Lambda^{-1}$, leading to $K_r^* = \Lambda^{-1} L_r = \operatorname{diag}(\lambda^*) L_r$. Letting $\hat{K}_x(t) \in \mathbb{R}^{m \times n_p}$ and $\hat{\lambda}(t) \in \mathbb{R}^m$, denote the adaptive estimates of the ideal parameters, the plant control law is defined as

$$u_p(t) = -\hat{K}_x(t) x_p(t) + \operatorname{diag}(\hat{\lambda}(t)) L_r y_c(t, t-\tau). \tag{3.7}$$

Substituting this control law into the plant dynamics (3.1) yields

$$\begin{aligned}\dot{x}_p(t) &= A_r x_p(t) + B_r y_c(t, t-\tau) \\ &\quad + B_p \Lambda \mathrm{diag}(\tilde{\lambda}(t)) L_r y_c(t, t-\tau) - B_p \Lambda \tilde{K}_x(t) x_p(t),\end{aligned} \tag{3.8}$$

where $\tilde{K}_x(t) \triangleq \hat{K}_x(t) - K_x^*$ and $\tilde{\lambda}(t) \triangleq \hat{\lambda}(t) - \lambda^*$ denote the parameter estimation errors.

Subtracting the reference model dynamics (3.2) from the plant dynamics (3.8), and using the identity $\Lambda \mathrm{diag}(\tilde{\lambda}(t)) L_r y_c(t, t-\tau) = \mathrm{diag}(L_r y_c(t, t-\tau)) \Lambda \tilde{\lambda}(t)$, the inner-loop tracking error is defined as $e_1(t) \triangleq x_p(t) - x_r(t)$, and the error dynamics become

$$\dot{e}_1(t) = A_r e_1(t) + B_p \mathrm{diag}(L_r y_c(t, t-\tau)) \Lambda \tilde{\lambda}(t) - B_p \Lambda \tilde{K}_x(t) x_p(t). \tag{3.9}$$

Then, the inner loop adaptive laws are chosen as

$$\dot{\hat{K}}_x^T(t) = \gamma_x x_p(t) e_1(t)^T P_1 B_p, \tag{3.10a}$$

$$\dot{\hat{\lambda}}(t) = \gamma_\lambda \mathrm{Proj}\left(\hat{\lambda}(t),\ -\mathrm{diag}(L_r y_c(t, t-\tau)) B_p^T P_1 e_1(t)\right), \tag{3.10b}$$

where the projection operator $\mathrm{Proj}(\cdot, \cdot)$ [36] enforces positive bounds on each $\hat{\lambda}_i(t)$, ensuring that $\hat{\lambda}_{max_i} > \hat{\lambda}_{min_i} > 0$ for all $i = 1, \ldots, m$. The constants $\gamma_x, \gamma_\lambda \in \mathbb{R}_+$ are learning rates, and $P_1 \in \mathbb{R}_+^{n_p \times n_p} \cap \mathbb{S}^{n_p \times n_p}$ is the solution to the following Lyapunov equation

$$A_r^T P_1 + P_1 A_r = -Q_1, \tag{3.11}$$

for some positive definite matrix $Q_1 \in \mathbb{R}_+^{n_p \times n_p} \cap \mathbb{S}^{n_p \times n_p}$.

Lemma 3.1. *Consider the uncertain plant dynamics given in (3.1), the reference model (3.2), and the adaptive control laws in (3.7) and (3.10). Then the closed-loop solution* $(e_1(t), \tilde{K}_x(t), \tilde{\lambda}(t))$ *is globally Lyapunov stable. Moreover,* $\lim_{t\to\infty} e_1(t) = 0$, *and the parameter estimation rates* $\dot{\tilde{K}}_x(t)$ *and* $\dot{\tilde{\lambda}}(t)$ *are bounded, as are all signals in the inner-loop.*

Proof: The command $y_c(t)$ in (3.3) is bounded because y_h is bounded due to the saturation limits of the human operator (physical manipulator), and y_l is bounded since the LSTM network is trained offline. As a result, its weights remain fixed during testing, and its output is constrained by the bounded activation functions. Therefore, the stability result follows directly from Lemma 1 in [14]. ■

3.1.2 Human Model

In the outer loop, control is shared between the human pilot, modeled using the adaptive framework introduced in [15], and an LSTM network. The pilot's objective is to guide the plant states to match those of a unity-feedback system known as the *crossover-reference model*, illustrated in Fig. 3.1, which is given by

$$\dot{x}_m(t) = A_m x_m(t) + B_m r(t-\tau). \tag{3.12}$$

Here, $x_m(t) \in \mathbb{R}^{n_p}$ denotes the state of the reference model, and $r(t) \in \mathbb{R}^m$ is a bounded reference. The matrix $A_m \in \mathbb{R}^{n_p \times n_p}$ is Hurwitz, and $B_m \triangleq B_r\theta_r \in \mathbb{R}^{n_p \times m}$. Following a similar reasoning as in the design of L_r in (3.5), the nominal feedforward gain $\theta_r \in \mathbb{R}^{m \times m}$ is chosen as

$$\theta_r = -(C_2^T A_m^{-1} B_r)^{-1}, \tag{3.13}$$

so that the steady-state output satisfies $\lim_{t\to\infty} y_2(t) = r$, provided that $\lim_{t\to\infty} x_p(t) = x_m(t)$. In addition, we assume the existence of a matrix $\theta_x \in \mathbb{R}^{m \times n_p}$ satisfying $A_m = A_r - B_p L_r \theta_x$.

The human control input formulation proposed in [15] is given by

$$\mathcal{G}(t) = \hat{\Phi}_1(t)x_p(t) + \theta_r r(t) + \int_{-\tau}^{0} \hat{\Phi}_2(t,\eta) L_r y_h(t+\eta)\,\mathrm{d}\eta, \tag{3.14a}$$

$$v(t) = L_r^{-1}\mathrm{diag}(\hat{\lambda}_2(t)) L_r \mathcal{G}(t), \tag{3.14b}$$

$$y_{h_i}(t) = \begin{cases} v_i(t), & \text{if } |v_i(t)| \le y_{o_i}, \\ y_{o_i}\mathrm{sgn}(v_i(t)), & \text{if } |v_i(t)| > y_{o_i}, \end{cases} \tag{3.14c}$$

where $\hat{\Phi}_1(t) \in \mathbb{R}^{m \times n_p}$, $\hat{\Phi}_2(t,\eta) \in \mathbb{R}^{m \times m}$, and $\hat{\lambda}_2(t) \in \mathbb{R}^m$ denote the adaptive estimates of the ideal quantities

$$\begin{aligned} \Phi_1^*(t) &= \bar{H}(t)\Phi(t+\tau, t), \\ \Phi_2^*(t,\eta) &= \bar{H}(t)\Phi(t+\tau, t+\eta+\tau) B_p \Lambda_2(t+\eta+\tau), \end{aligned} \tag{3.15}$$

and $\lambda_2^*(t)$, respectively, with $\bar{H}(t) \triangleq -(\theta_x + L_r^{-1}H^T(t+\tau))$, $H^T(t) \triangleq -\Lambda\tilde{K}_x(t)$, and $\Lambda_2(t) \triangleq \Lambda\mathrm{diag}(\hat{\lambda}(t))$. Due to the use of a projection operator in (3.10b), which enforces positive lower bounds on $\hat{\lambda}(t)$, we ensure that $\Lambda_2(t) \in \mathbb{D}_+$, so that $\mathrm{diag}(\lambda_2^*(t)) = \Lambda_2^{-1}(t+\tau)$ exists for all $t \geq 0$. Finally, (3.14c) represents an element-wise saturation function, where $y_{o_i} \in \mathbb{R}_+$ denotes the saturation limit for the i^{th} component of the human input vector $y_{h_i}(t)$.

The outer-loop tracking error dynamics for the overall system can be expressed as

$$\begin{aligned}\dot{e}_2(t) =& A_m e_2(t) + B_p \Lambda_2(t) L_r y_l(t) + B_p \Lambda_2(t) L_r \Delta y(t-\tau) \\ &+ B_p \mathrm{diag}(L_r \mathcal{G}(t-\tau)) \Lambda_2(t) \tilde{\lambda}_2(t-\tau) \\ &+ B_p L_r \tilde{\Phi}_1(t-\tau) x_p(t-\tau) \\ &+ B_p L_r \int_{-\tau}^{0} \tilde{\Phi}_2(t-\tau,\eta) L_r y_h(t+\eta-\tau) \mathrm{d}\eta,\end{aligned} \tag{3.16}$$

where $e_2(t) \triangleq x_p(t) - x_m(t)$ denotes the outer-loop tracking error, and the adaptive estimation errors are defined as $\tilde{\Phi}_1(t) \triangleq \hat{\Phi}_1(t) - \Phi_1^*(t)$, $\tilde{\Phi}_2(t,\eta) \triangleq \hat{\Phi}_2(t,\eta) - \Phi_2^*(t,\eta)$ and $\tilde{\lambda}_2(t) \triangleq \hat{\lambda}_2(t) - \lambda_2^*(t)$. Additionally, $\Delta y(t) \triangleq y_h(t) - v(t)$ represents the control deficiency introduced by saturation in the human command input.

An auxiliary signal $e_\Delta(t)$ is introduced to account for constraints on human input, following the approach in [15], and is governed by

$$\dot{e}_\Delta(t) = A_m e_\Delta(t) + B_p \mathrm{diag}(\hat{\lambda}_3(t)) L_r \Delta y(t-\tau), e_\Delta(t_0) = 0, \tag{3.17}$$

where $\mathrm{diag}(\lambda_3^*(t)) = \Lambda_2(t)$, and $\hat{\lambda}_3(t) \in \mathbb{R}^m$ denotes the adaptive estimate of the ideal value $\lambda_3^*(t)$. The augmented error signal for the outer-loop is then defined as $e_y(t) \triangleq e_2(t) - e_\Delta(t)$. To adapt the model parameters, update laws are employed as

$$\dot{\hat{\lambda}}_2(t) = \gamma_2 \mathrm{Proj}\left(\hat{\lambda}_2(t),\ -\mathrm{diag}(L_r \mathcal{G}(t-\tau)) B_p^T P_2 e_y(t)\right), \tag{3.18a}$$

$$\dot{\hat{\lambda}}_3(t) = \gamma_3 \mathrm{Proj}\left(\hat{\lambda}_3(t),\ \mathrm{diag}(L_r \Delta y(t-\tau)) B_p^T P_2 e_y(t)\right), \tag{3.18b}$$

$$\dot{\hat{\Phi}}_1^T(t) = \gamma_{\phi_1} \mathrm{Proj}\left(\hat{\Phi}_1^T(t),\ -x_p(t-\tau) e_y^T(t) P_2 B_p L_r\right), \tag{3.18c}$$

$$\dot{\hat{\Phi}}_2^T(t,\eta) = \gamma_{\phi_2} \mathrm{Proj}\left(\hat{\Phi}_2^T(t,\eta),\ -L_r y_h(t+\eta-\tau) e_y^T(t) P_2 B_p L_r\right), \tag{3.18d}$$

where $\gamma_2, \gamma_3, \gamma_{\phi_1}, \gamma_{\phi_2} \in \mathbb{R}_+$ are learning rates, and $P_2 \in \mathbb{R}_+^{n_p \times n_p} \cap \mathbb{S}^{n_p \times n_p}$ is the positive definite solution to the Lyapunov equation $A_m^T P_2 + P_2 A_m = -Q_2$, for some $Q_2 \in \mathbb{R}_+^{n_p \times n_p} \cap \mathbb{S}^{n_p \times n_p}$.

While the stability of an adaptive human model controlling an adaptive system was analyzed in [15], the current study requires a stability analysis for the complete system, which includes the integrated LSTM network. To this end, we first present Lemma 3.2, which builds upon the analysis provided in [15], and then proceed to state the main stability theorem of this work.

Lemma 3.2. *There exist $\phi \in \mathbb{R}_+$ and $\dot{\phi} \in \mathbb{R}_+$ such that*

$$\|\Phi(t+\tau, t)\|_F \leq \phi, \text{ for all } t \geq t_0, \tag{3.19a}$$

$$\|\Phi(t+\tau, t+\eta+\tau)\|_F \leq \phi, \text{ for all } t \geq t_0, -\tau \leq \eta \leq 0 \tag{3.19b}$$

$$\|\dot{\Phi}(t+\tau, t)\|_F \leq \dot{\phi}, \text{ for all } t \geq t_0, \tag{3.19c}$$

$$\|\dot{\Phi}(t+\tau, t+\eta+\tau)\|_F \leq \dot{\phi}, \text{ for all } t \geq t_0, -\tau \leq \eta \leq 0. \tag{3.19d}$$

Proof Summary (A detailed proof can be found in [15]): Considering the homogeneous part of (3.2), we obtain that $x_r(t) = 0$, and consequently $x_p(t) = e_1(t) + x_r(t) = e_1(t)$ for all $t > t_0$. Thus, the origin $x_p(t) = 0$ of the homogeneous part of the outer loop dynamics is uniformly stable (Lemma 3.1). The state transition matrix associated with the homogeneous component of the outer loop dynamics admits the bound specified in (3.19a) and (3.19b). Furthermore, the transition matrix $\Phi(t+\tau, t)$ can be expressed as $\Phi(t+\tau, t) = X(t+\tau)X^{-1}(t)$, where $X(t)$ denotes a *fundamental matrix* [37] that satisfies the differential equation $\dot{X}(t) = A(t)X(t)$, with $A(t) \triangleq A_r + B_p H^T(t)$. The time derivative of $\Phi(t+\tau, t)$ then takes the form $\dot{\Phi}(t+\tau, t) = A(t+\tau)\Phi(t+\tau, t) - \Phi(t+\tau, t)A(t)$. Given that $H^T(t) \triangleq -\Lambda\tilde{K}x(t)$ is bounded as established in Lemma 3.1, it follows that $A(t)$ is also bounded. Consequently, there exists a positive constant $\dot{\phi} \in \mathbb{R}_+$ such that the bounds in (3.19c) and (3.19d) hold. ∎

Theorem 3.1. *Given the dynamical system in (3.1), along with the adaptive controller specified by (3.2), (3.7), and (3.10), as well as the adaptive human pilot model described by (3.12), (3.14), and (3.18), there exists a positive scalar $\tau^* \in \mathbb{R}_+$ such that for any $\tau \in [0, \tau^*]$, the trajectory $\left(e_y(t), \tilde{\lambda}_2(t), \tilde{\lambda}_3(t), \tilde{\Phi}_1(t), \tilde{\Phi}_2(t,\eta)\right)$ remains bounded for all $t \geq t_0$ and asymptotically approaches a compact subset of the state space. Moreover, all signals in the closed-loop system remain bounded.*

Proof: The stability proof begins with the construction of a Lyapunov–Krasovskii candidate function

$$
\begin{aligned}
V_2 &= e_y^T(t)P_2e_y(t) + \gamma_3^{-1}\tilde{\lambda}_3^T(t)\tilde{\lambda}_3(t) \\
&+ \gamma_2^{-1}\tilde{\lambda}_2^T(t)\Lambda_2(t)\tilde{\lambda}_2(t) + \int_{-\tau}^{0}\int_{t+\nu}^{t}\dot{\tilde{\lambda}}_2^T(\xi)\dot{\tilde{\lambda}}_2(\xi)\mathrm{d}\xi\mathrm{d}\nu \\
&+ \gamma_{\phi_1}^{-1}\mathrm{Tr}\{\tilde{\Phi}_1^T(t)\tilde{\Phi}_1(t)\} + \int_{-\tau}^{0}\int_{t+\nu}^{t}\mathrm{Tr}\{\dot{\tilde{\Phi}}_1^T(\xi)\dot{\tilde{\Phi}}_1(\xi)\}\mathrm{d}\xi\mathrm{d}\nu \\
&+ \gamma_{\phi_2}^{-1}\int_{-\tau}^{0}\mathrm{Tr}\{\tilde{\Phi}_2^T(t,\eta)\tilde{\Phi}_2(t,\eta)\}\mathrm{d}\eta \\
&+ \int_{-\tau}^{0}\int_{t+\nu}^{t}\int_{-\tau}^{0}\mathrm{Tr}\{\dot{\tilde{\Phi}}_2^T(\xi,\eta)\dot{\tilde{\Phi}}_2(\xi,\eta)\}\mathrm{d}\eta\mathrm{d}\xi\mathrm{d}\nu.
\end{aligned}
\tag{3.20}
$$

Using the same analytical approach presented in [15], we compute the time derivative of (3.20) as

$$
\begin{aligned}
\dot{V}_2 &\leq -\lambda_{min}(Q_2)\left\|e_y(t)\right\|^2 \\
&+ \tau(\mu + 2\gamma_2^2)\left\|\mathrm{diag}(L_r\mathcal{G}(t-\tau))\right\|^2\left\|P_2B_p\right\|^2\left\|e_y(t)\right\|^2 \\
&+ \tau(1 + 2\gamma_{\phi_1}^2)\left\|x_p(t-\tau)\right\|^2\left\|e_y(t)\right\|^2\left\|P_2B_pL_r\right\|_F^2 \\
&+ \tau(1 + 2\gamma_{\phi_2}^2)\int_{-\tau}^{0}\|L_ry_h(t+\eta-\tau)\|^2\|e_y(t)\|^2\|P_2B_pL_r\|_F^2\,\mathrm{d}\eta \\
&+ 2\tau\dot{\lambda}_2^{*T}(t)\dot{\lambda}_2^*(t) + 2\tau\mathrm{Tr}\{\dot{\Phi}_1^{*T}(t)\dot{\Phi}_1^*(t)\} \\
&+ 2\tau\int_{-\tau}^{0}\mathrm{Tr}\{\dot{\Phi}_2^{*T}(t,\eta)\dot{\Phi}_2^*(t,\eta)\}\mathrm{d}\eta + 2e_y^T(t)P_2B_p\Lambda_2(t)L_ry_l(t) \\
&+ \gamma_2^{-1}\tilde{\lambda}_2^T(t)\dot{\Lambda}_2(t)\tilde{\lambda}_2(t) - 2\gamma_2^{-1}\tilde{\lambda}_2^T(t)\Lambda_2(t)\dot{\lambda}_2^*(t) \\
&- 2\gamma_3^{-1}\tilde{\lambda}_3^T(t)\dot{\lambda}_3^*(t) - 2\gamma_{\phi_1}^{-1}\mathrm{Tr}\{\dot{\Phi}_1^{*T}(t)\tilde{\Phi}_1(t)\} \\
&- 2\gamma_{\phi_2}^{-1}\int_{-\tau}^{0}\mathrm{Tr}\{\dot{\Phi}_2^{*T}(t,\eta)\tilde{\Phi}_2(t,\eta)\}\mathrm{d}\eta.
\end{aligned}
\tag{3.21}
$$

Since Lemma 3.1 guarantees that $H(t), \Lambda_2(t)$ and their time derivatives $\dot{H}(t), \dot{\Lambda}_2(t)$ are bounded, it follows that there exist constants $h, \dot{h}, \beta_3, \dot{\beta}_3 \in \mathbb{R}_+$ such that $\|H(t)\| \leq h$, $\|\dot{H}(t)\| \leq \dot{h}$, $\|\Lambda_2(t)\|_F \leq \beta_3$, and $\|\dot{\Lambda}_2(t)\|_F \leq \dot{\beta}_3$ for all $t \geq t_0$. Consequently, the corresponding ideal signals satisfy $\|\lambda_3^*(t)\| \leq \beta_3$ and $\|\dot{\lambda}_3^*(t)\| \leq \dot{\beta}_3$. Moreover, because the projection operator ensures $\hat{\lambda}_{\min_i} > 0$ for $i = 1, \ldots, m$, there exists a constant $\beta_2 \in \mathbb{R}_+$ such that $\|\Lambda_2^{-1}(t)\|_F \leq \beta_2$.

Using the identity $\frac{\mathrm{d}\Lambda_2^{-1}}{\mathrm{d}t} = -\Lambda_2^{-1}\dot{\Lambda}_2\Lambda_2^{-1}$, and the boundedness of $\dot{\Lambda}_2(t)$, it follows that there exists $\dot{\beta}_2 \in \mathbb{R}_+$ such that $\left\|\frac{\mathrm{d}\Lambda_2^{-1}}{\mathrm{d}t}\right\|_F \leq \dot{\beta}_2$. Hence, for $t \geq t_0$, $\|\lambda_2^*(t)\| \leq \beta_2$ and $\|\dot{\lambda}_2^*(t)\| \leq \dot{\beta}_2$. Finally, applying Lemma 3.2 and the definitions in (3.15), we deduce the existence of constants $\phi_1, \dot{\phi}_1, \phi_2, \dot{\phi}_2 \in \mathbb{R}_+$ such that $\|\Phi_1^*(t)\|_F \leq \phi_1$, $\left\|\dot{\Phi}_1^*(t)\right\|_F \leq \dot{\phi}_1$, $\|\Phi_2^*(t,\eta)\|_F \leq \phi_2$, $\left\|\dot{\Phi}_2^*(t,\eta)\right\|_F \leq \dot{\phi}_2$ for all $t \geq t_0$, $-\tau \leq \eta \leq 0$.

By incorporating the established upper bounds and introducing the constant $p \triangleq \max(\|P_2B_p\|^2, \|P_2B_pL_r\|_F^2)$, (3.21) can be rewritten as

$$\begin{aligned}
\dot{V}_2 \leq & -\lambda_{min}(Q_2)\|e_y(t)\|^2 \\
& + \tau p(\mu + 2\gamma_2^2)\|\mathrm{diag}(L_r\mathcal{G}(t-\tau))\|^2\|e_y(t)\|^2 \\
& + \tau p(1 + 2\gamma_{\phi_1}^2)\|x_p(t-\tau)\|^2\|e_y(t)\|^2 \\
& + \tau p(1 + 2\gamma_{\phi_2}^2)\int_{-\tau}^{0}\|L_ry_h(t+\eta-\tau)\|^2\|e_y(t)\|^2\,\mathrm{d}\eta \\
& + 2\tau\dot{\beta}_2^2 + 2\tau\dot{\phi}_1^2 + 2\tau^2\dot{\phi}_2^2 + 2\sqrt{p}\beta_3\|e_y(t)\|\|L_ry_l(t)\| \\
& + \gamma_2^{-1}\tilde{\beta}_2^2\dot{\beta}_3 + 2\gamma_2^{-1}\tilde{\beta}_2\beta_3\dot{\beta}_2 \\
& + 2\gamma_3^{-1}\tilde{\beta}_3\dot{\beta}_3 + 2\gamma_{\phi_1}^{-1}\dot{\phi}_1\tilde{\phi}_1 + 2\gamma_{\phi_2}^{-1}\tau\dot{\phi}_2\tilde{\phi}_2,
\end{aligned} \tag{3.22}$$

where $\tilde{\beta}_2 \triangleq \left\|\hat{\lambda}_{2max}\right\| + \beta_2$, $\tilde{\beta}_3 \triangleq \left\|\hat{\lambda}_{3max}\right\| + \beta_3$, $\tilde{\phi}_1 \triangleq \left\|\hat{\Phi}_{1max}\right\| + \phi_1$, and $\tilde{\phi}_2 \triangleq \left\|\hat{\Phi}_{2max}\right\| + \phi_2$. Defining $q \triangleq \lambda_{min}(Q_2)/p$, (3.22) can be rewritten as

$$\begin{aligned}
\dot{V}_2 \leq \ & p\|e_y(t)\|^2\Big(-q + \tau\Big\{(\mu + 2\gamma_2^2)\|\mathrm{diag}(L_r\mathcal{G}(t-\tau))\|^2 \\
& + (1 + 2\gamma_{\phi_1}^2)\|x_p(t-\tau)\|^2 + 2\sqrt{p}\beta_3\|e_y(t)\|\|L_ry_l(t)\| \\
& + (1 + 2\gamma_{\phi_2}^2)\int_{-\tau}^{0}\|L_ry_h(t+\eta-\tau)\|^2\,\mathrm{d}\eta\Big\}\Big) \\
& + 2\tau(\dot{\beta}_2^2 + \dot{\phi}_1^2 + \tau\dot{\phi}_2^2) + \gamma_2^{-1}\tilde{\beta}_2^2\dot{\beta}_3 + 2\gamma_2^{-1}\tilde{\beta}_2\beta_3\dot{\beta}_2 \\
& + 2\gamma_3^{-1}\tilde{\beta}_3\dot{\beta}_3 + 2\gamma_{\phi_1}^{-1}\dot{\phi}_1\tilde{\phi}_1 + 2\gamma_{\phi_2}^{-1}\tau\dot{\phi}_2\tilde{\phi}_2.
\end{aligned} \tag{3.23}$$

From Lemma 3.1 and the saturation condition in (3.14c), it follows that there exist constants $\alpha_1 \in \mathbb{R}_+$ and $\alpha_2 \in \mathbb{R}_+$ such that $\|x_p(t)\|^2 \leq \alpha_1$ and $\|L_ry_h(t)\|^2 \leq \alpha_2$ for all $t \geq t_0$. Furthermore, since the reference signal $r(t)$ is bounded and the projection operator is employed in the adaptive laws (3.18), the signal $\mathcal{G}(t)$

defined in (3.14a) is also bounded. Consequently, there exists $\alpha_3 \in \mathbb{R}_+$ such that $\|\mathrm{diag}(L_r\mathcal{G}(t))\|^2 \leq \alpha_3$ for all $t \geq t_0$. In addition, as the LSTM network uses bounded activation functions and is trained offline, there exists $\alpha_4 \in \mathbb{R}_+$ such that $\|L_r y_l(t)\| \leq \alpha_4$ for all $t \geq t_0$. Therefore, equation (3.23) can be rewritten as

$$\begin{aligned}\dot{V}_2 \leq\ & p\,\|e_y(t)\|^2 \Big(-q + \tau\Big\{(\mu + 2\gamma_2^2)\alpha_3 + (1 + 2\gamma_{\phi_1}^2)\alpha_1 \\ & + (1 + 2\gamma_{\phi_2}^2)\tau\alpha_2\Big\}\Big) + 2\sqrt{p}\beta_3\alpha_4\,\|e_y(t)\| \\ & + 2\tau(\dot{\beta}_2^2 + \dot{\phi}_1^2 + \tau\dot{\phi}_2^2) + \gamma_2^{-1}\tilde{\beta}_2^2\dot{\beta}_3 + 2\gamma_2^{-1}\tilde{\beta}_2\beta_3\dot{\beta}_2 \\ & + 2\gamma_3^{-1}\tilde{\beta}_3\dot{\beta}_3 + 2\gamma_{\phi_1}^{-1}\dot{\phi}_1\tilde{\phi}_1 + 2\gamma_{\phi_2}^{-1}\tau\dot{\phi}_2\tilde{\phi}_2.\end{aligned} \tag{3.24}$$

Accordingly, there exists a sufficiently small $\tau^* \in \mathbb{R}_+$ such that the condition

$$\tau^*\Big\{(\mu + 2\gamma_2^2)\alpha_3 + (1 + 2\gamma_{\phi_1}^2)\alpha_1 + (1 + 2\gamma_{\phi_2}^2)\tau^*\alpha_2\Big\} < q. \tag{3.25}$$

is satisfied. Therefore, for any $\tau \in [0, \tau^*]$, the inequality in (3.24) guarantees that $\dot{V}_2 < 0$ whenever the condition

$$\|e_y(t)\| > \frac{\sqrt{4z_2z_1 + z_3^2} + z_3}{2z_2}, \tag{3.26}$$

holds, where

$$\begin{aligned} z_1 \triangleq\ & 2\tau(\dot{\beta}_2^2 + \dot{\phi}_1^2 + \tau\dot{\phi}_2^2) + \gamma_2^{-1}\tilde{\beta}_2^2\dot{\beta}_3 + 2\gamma_2^{-1}\tilde{\beta}_2\beta_3\dot{\beta}_2 \\ & + 2\gamma_3^{-1}\tilde{\beta}_3\dot{\beta}_3 + 2\gamma_{\phi_1}^{-1}\dot{\phi}_1\tilde{\phi}_1 + 2\gamma_{\phi_2}^{-1}\tau\dot{\phi}_2\tilde{\phi}_2,\end{aligned}$$

$$z_2 \triangleq p\Big(q - \tau\Big\{(\mu + 2\gamma_2^2)\alpha_3 + (1 + 2\gamma_{\phi_1}^2)\alpha_1 + (1 + 2\gamma_{\phi_2}^2)\tau\alpha_2\Big\}\Big), \tag{3.27}$$

$$z_3 \triangleq 2\sqrt{p}\beta_3\alpha_4. \tag{3.28}$$

Hence, the solution $(e_y(t), \tilde{\lambda}_2(t), \tilde{\lambda}_3(t), \tilde{\Phi}_1(t), \tilde{\Phi}_2(t,\eta))$ remains bounded and converges to the compact set defined as

$$\begin{aligned} E \triangleq \Big\{ & (e_y(t), \tilde{\lambda}_2(t), \tilde{\lambda}_3(t), \tilde{\Phi}_1(t), \tilde{\Phi}_2(t,\eta)) : \\ & \|e_y(t)\| > \frac{\sqrt{4z_2z_1 + z_3^2} + z_3}{2z_2}, \left\|\tilde{\lambda}_2(t)\right\| \leq \tilde{\beta}_2, \\ & \left\|\tilde{\lambda}_3(t)\right\| \leq \tilde{\beta}_3, \left\|\tilde{\Phi}_1(t)\right\| \leq \tilde{\phi}_1, \left\|\tilde{\Phi}_2(t,\eta)\right\| \leq \tilde{\phi}_2 \Big\},\end{aligned} \tag{3.29}$$

for all $\tau \in [0, \tau^*]$. Since the reference input $r(t)$ is bounded, it follows from (3.12) that the reference model state $x_m(t)$ is bounded. Given the boundedness of $x_p(t)$, the outer-loop tracking error $e_2(t)$ is also bounded. Moreover, the auxiliary error $e_\Delta(t)$ is bounded because the augmented error $e_y(t) = e_2(t) - e_\Delta(t)$ is bounded. This, in turn, implies that $\Delta y(t)$ is bounded from (3.17). ■

3.1.3 LSTM Network

To assist the pilot in reducing the reference tracking error defined in (3.16), we utilize an LSTM network [33]. Ideally, the network would receive the reference tracking error $e_2(t)$ as input. However, in experimental scenarios where the system is under human control, $e_2(t)$ is not directly observable by the network. To address this, we generate a reference input $r_l(t)$ obtained from a predefined mission target, such as one derived from visual sensor data (see Figure 3.2) and feed it into the reference model to estimate the pilot's intention. This yields the signal $x_{m,l}(t) = A_m x_{m,l}(t) + B_m r_l(t-\tau)$. The LSTM network then receives the estimated error $e_{2,l}(t) \triangleq x_p(t) - x_{m,l}(t)$ as input and outputs a preliminary control signal denoted as $y_l'(t)$.

To achieve ideal tracking performance, the LSTM network is trained to cancel out the components responsible for deviations from the reference model. Specifically, if the LSTM output is set to

$$\begin{aligned} y_l'(t) = &-\Delta y(t-\tau) - L_r^{-1}\Lambda_2^{-1}(t) L_r \tilde{\Phi}_1(t-\tau) x_p(t-\tau) \\ &- L_r^{-1}\Lambda_2^{-1}(t)\mathrm{diag}(L_r \mathcal{G}(t-\tau))\Lambda_2(t)\tilde{\lambda}_2(t-\tau) \\ &- L_r^{-1}\Lambda_2^{-1}(t) L_r \int_{-\tau}^{0} \tilde{\Phi}_2(t-\tau,\eta) L_r y_h(t+\eta-\tau)\mathrm{d}\eta, \end{aligned} \tag{3.30}$$

then the resulting outer-loop error dynamics reduce to $\dot{e}_2(t) = A_m e_2(t)$. Thus, the expression in 3.30 is treated as the *truth* function that the LSTM is trained to approximate in order to recover the desired error behavior.

3.1.4 Modulation of LSTM

The LSTM network's input is $e_{2,l}(t)$, an estimate of $e_2(t)$, which may be inaccurate due to various reasons. For example, sensors may incorrectly identify the target location, the estimation of $r_l(t)$ may be erroneous, or the pilot may choose to deviate from a predetermined mission. In such cases, the LSTM network, operating with an incorrect estimate, may interfere with the pilot's control actions since the LSTM's and the pilot's control objectives are misaligned. To detect such discrepancies, we analyze the correlation between the inner-loop reference model responses to the pilot input alone and to the combined inputs from the pilot and the LSTM. These responses are computed as $\dot{x}_{r,h}(t) = A_r x_{r,h}(t) + B_r y_h(t-\tau)$, $y_{2,h} = C_2^T x_{r,h}(t)$, for the pilot input only, and $\dot{x}_{r,l}(t) = A_r x_{r,l}(t) + B_r(y_l'(t) + y_h(t-\tau))$, $y_{2,l} = C_2^T x_{r,l}(t)$ for the combined pilot and LSTM inputs.

We propose that when the correlation between the outputs $y_{2,h}$ and $y_{2,l}$, which represent the intended tracking objectives, decreases, the influence of the LSTM should be reduced. This is achieved by scaling the LSTM signal $y_l'(t)$ with a modulation coefficient ρ_m, such that the effective LSTM input becomes $y_l(t) = \rho_m y_l'(t)$. This modulation mechanism is based on the following shared control principles: (a) the human operator must be able to reclaim control authority when necessary, (b) the actions of the LSTM must remain interpretable and predictable to the human operator, and (c) if the estimation of human intent is substantially inaccurate, or if the reference signal followed by the operator changes, the LSTM's contribution to the control should diminish.

The modulation coefficient ρ_m is computed by first evaluating the Pearson correlation coefficient between $y_{2,h}$ and $y_{2,l}$, which can be written as

$$\rho(t_k) = \frac{\sum_{i=k-n}^{k} \left(y_{2,h}(t_i) - \bar{y}_{2,h}\right)\left(y_{2,l}(t_i) - \bar{y}_{2,l}\right)}{\sqrt{\sum\limits_{i=k-n}^{k} \left(y_{2,h}(t_i) - \bar{y}_{2,h}\right)^2}\sqrt{\sum\limits_{i=k-n}^{k} \left(y_{2,l}(t_i) - \bar{y}_{2,l}\right)^2}}, \tag{3.31}$$

where the bar notation indicates the mean over the sample window, t_k denotes the current sampling instant, and n is the number of samples collected over a time

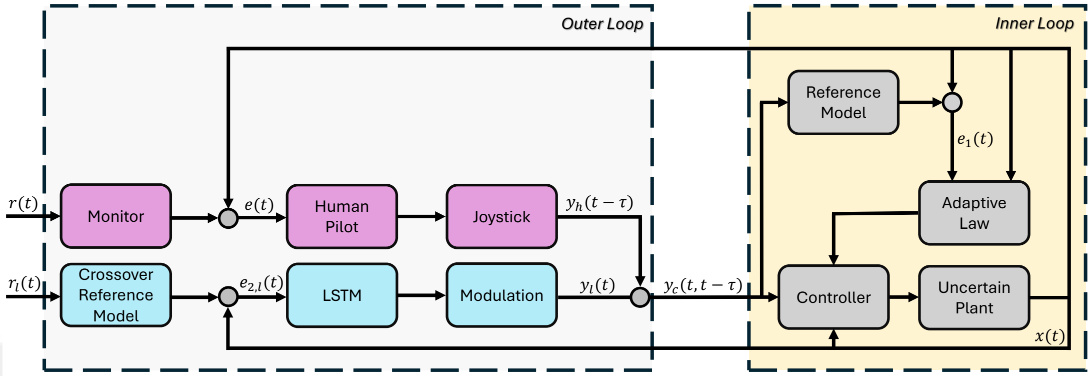

Figure 3.2: Control architecture used in the experiments.

window of duration τ_p. Using the correlation value, the modulation coefficient is then computed as $\rho_m = \exp(-(\rho-1)^2/(0.1^2))$ which yields values within the range $[0,1]$. According to user feedback, the time window τ_p is set to 10 seconds.

3.2 Experiments

A flight control experiment is conducted using the setup illustrated in Figure 3.2. A total of 16 participants from the Department of Mechanical Engineering at Bilkent University took part in the study, which was approved by the Bilkent University Ethics Committee. Informed consent was obtained from all participants. During the experiment, the subjects observed both the aircraft's pitch angle and a reference signal displayed on a screen. They then issued pitch rate commands to the inner loop via a Logitech Extreme 3D Pro joystick. The joystick's saturation limits were configured as $y_{o_i} = 10$ deg/s for $i = 1, \ldots, 3$ (see (3.14c)).

3.2.1 Inner Loop Parameters

For the plant described in (3.1), a linearized model of a Boeing 747 in level cruise flight at 774 ft/s and an altitude of 40 kft is employed [15]. The state

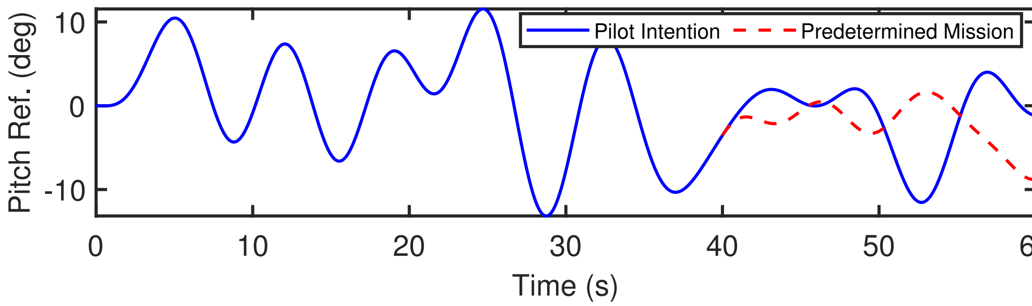

Figure 3.3: Pitch references used in experiments.

vector of the plant includes the velocities along the x- and z-axes, the pitch rate, and the pitch angle. The control input $u_p(t)$ corresponds to the elevator deflection, with magnitude and rate saturation limits specified as $+17/-23$ deg and $+37/-37$ deg/s, respectively [38].

For the inner-loop controller, the feedback gain L_x in (3.4b) is determined using an LQR design with weighting matrices $Q_{\text{LQR}} = \text{diag}([0.001,\ 0.01,\ 10,\ 0.1])$ and $R_{\text{LQR}} = 10$. The Lyapunov matrix P_1 in (3.11) is obtained by selecting $Q_1 = 10^{-3} I_{4\times4}$. The learning rates in the adaptive update laws (3.10) are set to $\gamma_x = 2$ and $\gamma_\lambda = 4$. Due to the presence of a zero at the origin in the transfer function $x_{p3}(s)/u_p(s)$, the inner-loop feedforward gain L_r is designed based on the short-period approximation of the aircraft dynamics [39]. Specifically, it is computed as $L_r = -\left(C_{\text{sp}}^T A_{\text{sp}}^{-1} B_{\text{sp}}\right)^{-1}$, where A_{sp}, B_{sp}, and C_{sp} represent the short-period model matrices.

3.2.2 Network Training

The LSTM network is trained through simulations employing the human pilot model described in Section 3.1.2. The parameters of the pilot model are identified using experimental participant data and nonlinear optimization routines available in MATLAB. Based on the optimization results, the parameter vector is set as $\theta_x = [0.0115, 0.0009, -0.3535, 0.7673]$, and the human response delay is determined to be $\tau = 0.18$ s. The feedforward gain θ_r is selected according to (3.13). For the Lyapunov equation of the outer loop, the matrix is chosen as

$Q_2 = 10^{-3} I_{4\times4}$. The implementation of the finite integral term in (3.14a) and the update law in (3.18d) is carried out by discretizing the integral, following the approach in [40]. The learning rates are selected as $\gamma_2 = 1.2, \gamma_3 = 1.3, \gamma_{\phi_1} = 0.8 I_{4\times4}$ and $\gamma_{\phi_2} = 2.2$.

The LSTM network is designed with a single hidden layer comprising 64 neurons. Its input layer contains 4 neurons, corresponding to the four states involved in the outer-loop error dynamics. Since the system includes a single actuator, the output layer consists of one neuron connected through a fully connected layer. Network weights are initialized using the Xavier initialization method, and training is performed using the Adam optimizer. The simulation runs with a fixed time step of 0.01 s. Training is conducted for 1500 iterations, employing an L2 regularization factor of 0.0001, a gradient decay rate of 0.9, and a squared gradient decay rate of 0.999.

To enhance robustness, training incorporates varying plant and actuator uncertainties. Specifically, for $t \leq 20$, the system parameters are set as $A_p = A_n$ and $\Lambda = I$. For $t > 20$, the plant model is perturbed using $A_p = A_n - B_p W_{\text{unc}}$, and control effectiveness is modified as $\Lambda = \Lambda_v$. The uncertainty matrix $W_{\text{unc}} \in \mathbb{R}^{n_p \times m}$ contains entries randomly selected from the interval $[-0.2, 0.2]$ for each training episode. Similarly, the diagonal elements of $\Lambda_v \in \mathbb{R}_+^{m\times m}$ are randomly chosen between 0.5 and 1.

3.2.3 Results

During the experiments, the first 20 seconds are conducted under nominal conditions, where the pilot tracks the reference signal without any uncertainty. After this period, system uncertainties are introduced: the plant dynamics in (3.1) are modified as $A_p = A_n - B_p[0.16,\ 0.10,\ 0.07,\ -0.07]$, and the control effectiveness matrix is set to $\Lambda = 0.8$. At $t = 40$ seconds, the reference signal is split into two separate trajectories, as illustrated in Figure 3.3. Participants are instructed to follow the reference labeled as the pilot's intention, whereas the LSTM network continues to follow the predetermined mission defined by $r_l(t)$, as shown in

Table 3.1: Workload (W), Precision (P), and TPX for Pitch Angle (θ) Tracking for Pilot Only and LSTM Aided Cases, by Intervals.

Interval	Case	W	P	TPX
$0 \leq t < 20$	Pilot Only	1.03	79.39	0.33
	with LSTM	1.05	93.81	0.44
$20 \leq t < 40$	Pilot Only	1.14	58.16	0.20
	with LSTM	1.02	72.85	0.30
$40 \leq t < 60$	Pilot Only	1.14	77.40	0.32
	with LSTM	1.08	74.06	0.30

Figure 3.2.

The system is evaluated under two conditions: the pilot assisted by the LSTM network, and the pilot operating the system without assistance. Evaluation is based on two performance criteria: reference tracking precision P, and workload W. Workload is quantified as the average number of direction changes per second in the human control input [41]. Precision is computed as the percentage of task duration in which the system output remains within predefined performance bounds. These bounds are defined by the maximum positive and negative tracking errors observed when the crossover model (3.12) controls the uncertainty-free plant. The overall task performance index (TPX) is then calculated as [41]:

$$\text{TPX} = \frac{P^2\sqrt{W_{\text{min}}}}{100^2\sqrt{W}},$$

where W_{min} denotes the workload associated with the ideal human input.

Table 3.1 presents the averaged performance metrics collected from 16 participants. A key observation from the data is that the inclusion of the LSTM network significantly enhances both tracking precision and TPX during the first two time intervals. This improvement is particularly notable in the second interval, where the system operates under model and control effectiveness uncertainties. In the third interval, although both TPX and precision experience a slight reduction in the LSTM-assisted case compared to the pilot-only scenario, the decrease is very limited. This indicates that the LSTM's influence is appropriately reduced when the pilot deviates from the predefined mission, thereby preserving the pilot's control authority. This outcome is attributed to the effectiveness of the modulation

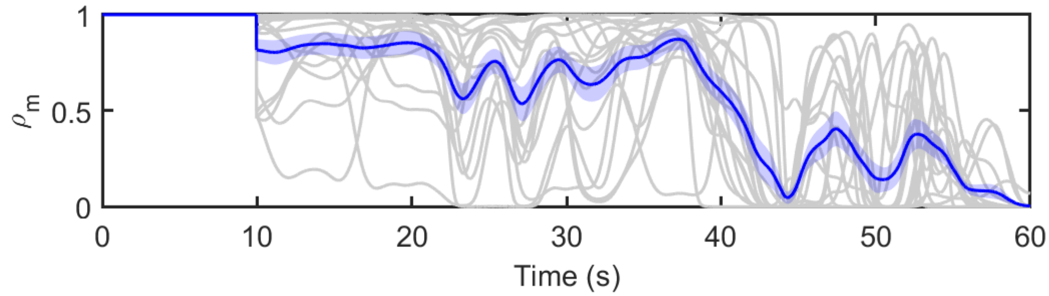

Figure 3.4: Modulation coefficient across multiple experiments. Gray lines: individual trajectories, and the blue line: mean values with the standard error of the mean.

mechanism. As illustrated in Figure 3.4, the modulation coefficient is relatively high during the first two intervals (indicating strong LSTM influence) and decreases in the third interval (indicating reduced LSTM influence). Additionally, the workload remains relatively stable throughout the experiment, demonstrating that the assistive system does not impose an additional burden on the pilot, even in situations where the reference signals for the pilot and the LSTM diverge.

Remark 3.1. *When the LSTM network operates at full capacity without modulation, a reduction in workload is observed during the first two intervals with LSTM augmentation, followed by an increase in the third interval. However, as discussed in Section 3.1.4, introducing a modulation coefficient provides distinct advantages. It ensures a more consistent workload across varying scenarios by dynamically adjusting the LSTM's influence based on the agreement with pilot intention.*

To summarize, the results demonstrate the LSTM network's ability to substantially enhance system performance, particularly under uncertain conditions. Furthermore, they highlight the system's capability to regulate the LSTM's influence, thereby preserving the human operator's control authority in cases of potential conflict between human input and automated assistance.

Chapter 4

Arbitration with Control Barrier Functions for Safe Shared Control

While certain tasks can be fully automated, human input is beneficial for most high-risk applications because humans and machines have complementary strengths of performance, flexibility, judgment, and resilience [3]. For instance, humans can act as overseers while automation performs routine tasks until anomalies arise, or both may act simultaneously, where each of them is capable of independently executing the task under ideal conditions [5]. The latter is usually referred to as *shared control*, and given the limitations of automation, it is most beneficial in safety-critical scenarios.

Establishing shared control frameworks requires careful consideration. One of the guiding design principles is human-centeredness, where the human operator has the final authority [42]. It is also important that the human operator can complete the task independently, particularly when a failure and/or degradation of the automation occurs [43]. Furthermore, the transparency and predictability of automation behavior are central, providing increased situational awareness to the human operator. This awareness enables the operator to override the system when necessary. Accordingly, these aspects must also be reflected in how shared control frameworks are validated. Validation should not only consider nominal conditions, but also the system performance beyond the design specifications of the automation. Evaluation in such a way ensures robustness and resiliency, especially in dynamic and safety-critical emergencies where human flexibility and responsiveness are essential [44].

A key requirement in shared control is arbitration: determining how to blend human and automation inputs. Numerous arbitration schemes have been developed for domains such as manufacturing [9], flight control [25], and driving [45]. These schemes are often based on adaptive weightings that depend on safety metrics, distraction, workload, or trust [46, 47, 48, 49]. They frequently rely on complex, scenario-specific functional forms tailored to particular system architectures [26, 27].

Rather than tuning parameter-dependent arbitration schemes, we propose a Control Barrier Function (CBF) based approach, where constraints are enforced through computationally efficient quadratic programming (QP) formulations [19]. By expressing shared control metrics as inequality constraints, we enable a principled automation intervention. For example, rising cognitive workload may lead to increased attention and reduce trust simultaneously. These multiple metrics can be given their own bounds within the optimization, resulting in a far richer arbitration than a single mixing coefficient. The formulation is modular, as new barriers or metrics can be added without changing the structure of the arbitration approach. It is also scalable since the inclusion of additional metrics or barriers requires only augmenting the safe set with negligible cost in computation. Moreover, the method is interpretable, meaning changes in behavior can be analytically linked to specific barriers or metrics, making it possible to understand which factor altered the automation assistance in a given instance. This also makes it easier to modify the barrier conditions for each metric, enabling targeted adjustments of the system's response to attention, fatigue, or workload.

In most shared control applications, CBFs have been employed to prevent human operators from driving systems into unsafe regions [20, 21, 22], or see humans

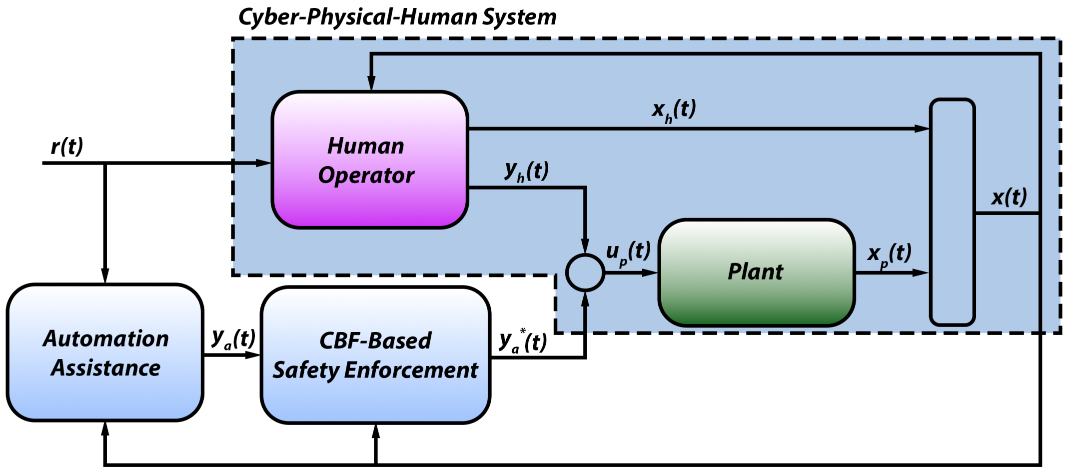

Figure 4.1: Block diagram of the proposed architecture.

as passive entities that must not be harmed [23, 24]. A recent effort introduces a performance-based trust metric as a CBF constraint that supports adaptive cruise control in a traded control setting, where either the human or the automation is in control at any given moment [50]. While these works do address shared control, they use CBFs primarily to ensure safety with respect to the physical system's states. To the best of our knowledge, there is no prior work that employs CBFs to regulate automation assistance based on shared control metrics such as workload, attention, or trust within a simultaneous control framework. Our approach formulates an arbitration problem, where CBFs enforce collaborative constraints that directly reflect human-system interaction quality.

4.1 Framework

In this section, we introduce a safe shared-control framework, where safety is guaranteed via a careful control barrier function (CBF) design, as seen in Fig. 4.1. It is noted that although the overall architecture contains an automation assistant, we do not focus on its design, since this framework is independent of the specific assistance employed. Rather, we are interested in investigating how the nominal output of an assistive automation needs to be altered to ensure a safe shared control scheme.

4.1.1 Plant Dynamics

We consider a generic control-affine plant dynamics

$$\dot{x}_p = f_p(x_p) + g_p(x_p)u_p, \tag{4.1}$$

where $x_p \in \mathbb{R}^{n_p}$, $f_p : \mathbb{R}^{n_p} \to \mathbb{R}^{n_p}$ and $g_p : \mathbb{R}^{n_p} \to \mathbb{R}^{n_p \times q_p}$ are locally Lipschitz, and $u_p \in \mathbb{R}^{q_p}$. In the absence of a "safety enforcer" (see Fig. 4.1), the plant control input u_p can be written as

$$u_p = y_h + y_a, \tag{4.2}$$

where $y_h \in \mathbb{R}^{q_p}$ is the human operator command, and $y_a \in \mathbb{R}^{q_p}$ is the assistance signal produced by the automation.

Remark 4.1. *In this work, we do not commit to a specific type of assistance strategy. As one illustrative example, the automation can act to minimize the deviation between the human's input y_h and the nominally ideal behavior y_h^* (assuming it's known). This would imply the classical input mixing strategy studied in the literature [51], $u_p = (1-\lambda)y_h + \lambda y_h^\star = y_h + \lambda(y_h^\star - y_h)$. Here, the assistance signal becomes $y_a = y_h^\star - y_h$, and the control authority parameter λ is a design parameter that is usually time-varying.*

4.1.2 Human Operator Model

For the human operator, we consider a dynamic system given as

$$\dot{x}_h = f_h(x_h, x_p) + g_h(x_h, x_p)r \tag{4.3a}$$

$$y_h = h_h(x_h) + j_h(x_h)r, \tag{4.3b}$$

where $x_h \in \mathbb{R}^{n_h}$ is the human operator state vector, $x_p \in R^{n_p}$ is given in (4.1), and $f_h : \mathbb{R}^{n_h+n_p} \to \mathbb{R}^{n_h}$, $g_h : \mathbb{R}^{n_h+n_p} \to \mathbb{R}^{n_h \times q_h}$, $h_h : \mathbb{R}^{n_h} \to \mathbb{R}^{q_p}$ and $j_h : \mathbb{R}^{n_h} \to \mathbb{R}^{q_p \times q_h}$

are locally Lipschitz functions, and $r \in \mathbb{R}^{q_h}$ is the reference signal vector. Using (4.1)-(4.3), we obtain that

$$\dot{x} = \begin{bmatrix} \dot{x}_p \\ \dot{x}_h \end{bmatrix} = \begin{bmatrix} f_p(x_p) + g_p(x_p)h_h(x_h) \\ f_h(x_h, x_p) \end{bmatrix} + \begin{bmatrix} g_p(x_p) & g_p(x_p)j_h(x_h) \\ 0_{n_h \times q_p} & g_h(x_h, x_p) \end{bmatrix} \begin{bmatrix} y_a \\ r \end{bmatrix}, \quad (4.4)$$

where $x = [x_p^T \quad x_h^T]^T \in \mathbb{R}^n$ with $n = n_p + n_h$. The vector fields in (4.4) can be identified as

$$f(x) = \begin{bmatrix} f_p(x_p) + g_p(x_p)h_h(x_h) \\ f_h(x_h, x_p) \end{bmatrix}, \quad (4.5)$$

$$g_a(x) = \begin{bmatrix} g_p(x_p) \\ 0_{n_h \times q_p} \end{bmatrix}, \quad (4.6)$$

$$g_r(x) = \begin{bmatrix} g_p(x_p)j_h(x_h) \\ g_h(x_h, x_p) \end{bmatrix}, \quad (4.7)$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$, $g_a : \mathbb{R}^n \to \mathbb{R}^{n \times q_p}$ and $g_r : \mathbb{R}^n \to \mathbb{R}^{n \times q_h}$ are locally Lipschitz with respect to their arguments. Taking the automation assistance signal y_a in (4.2) and the reference signal r in (4.3) as its inputs, the overall shared control system dynamics can then be obtained using (4.1)-(4.7) as

$$\dot{x} = f(x) + g_a(x)y_a + g_r(x)r. \quad (4.8)$$

4.1.3 Arbitration Control Barrier Functions (ACBF)

We begin by supplying the definition of the Control Barrier Function from [52].

Definition 4.1. *Consider the control-affine system*

$$\dot{x} = f(x) + g(x)u, \quad (4.9)$$

with $x \in X \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times q}$ are locally Lipschitz, and $u \in U \subset \mathbb{R}^q$. A continuously differentiable function $b : \mathbb{R}^n \to \mathbb{R}$ defined over a safe set

$$C \triangleq \{x \in \mathbb{R}^n \mid b(x) \geq 0\} \quad (4.10)$$

is called a Control Barrier Function (CBF) if there exists a class $\mathcal{K}$ function α such that

$$\sup_{u \in U} [L_f b(x) + L_g b(x) u + \alpha(b(x))] \geq 0, \quad \forall x \in C, \tag{4.11}$$

where L_f and L_g are the Lie derivatives, which show how CBF evolve along the trajectories of the dynamical system. Lie derivatives of b along the vector fields f and g, respectively, are given by

$$L_f b(x) = \nabla b(x)^\top f(x), \quad L_g b(x) = \nabla b(x)^\top g(x). \tag{4.12}$$

It is assumed that $L_g b(x) \neq 0$ when $b(x) = 0$.

Considering the shared control system dynamics (4.8), where the reference signal r is a known exogenous input, the condition (4.11) becomes

$$\sup_{y_a \in Y_a} [L_f b(x) + L_{g_r} b(x) r + L_{g_a} b(x) y_a + \alpha(b(x))] \geq 0$$
$$\forall x \in C, \tag{4.13}$$

where Y_a denotes a closed set of allowable automation assistance signals. In the context of shared control, the CBF $b(x)$ in (4.13) can represent the attention of the human operator to the task, so the safe set C given in (4.10) can describe the region where the operator has enough attention to the task.

As described in Definition 4.1, the function $b(x)$ in the inequality (4.13) being a CBF ensures that there exists at least one automation assistance input $y_a \in Y_a$ such that the inequality (e.g., safety condition) can be satisfied. However, the conditions when regulating the automation assistance in the case of shared control need to be stricter: Even before defining the barriers regarding workload, trust, or attention, we require the safe set C in (4.10) to be characterized by a region where the human operator can keep the system within C without any help from the assistance. Moreover, shared control principles also dictate that automation disengagement, i.e., $y_a = 0$, must always be feasible in terms of shared control safety. In other words, the human must be able to keep the system within the safe limits even if the automation is switched off, since such instances can lead to unsafe situations [53]. This sets our foundation for regulating automation assistance. Considering the shared control system dynamics (4.8), we define *Arbitration Control Barrier Functions* as follows:

Definition 4.2. *Given a safe set $C \triangleq \{x \in \mathbb{R}^n \mid b(x) \geq 0\}$, $b(x)$ is an Arbitration Control Barrier Function (ACBF) for system (4.8) if there exists a class $\mathcal{K}$ function α such that*

$$L_f b(x) + L_{g_r} b(x) r + \alpha(b(x)) \geq 0, \quad \forall x \in C. \tag{4.14}$$

where L_f and L_{g_r} denote Lie derivatives along f and g_r, respectively.

It is noted that (4.14) is the same as (4.13) for $y_a = 0$. Unlike (4.13), the existence of any y_a in the allowable automation assistance set Y_a is not enough, but $y_a = 0$ must always be feasible to satisfy (4.14).

Remark 4.2. *Definition 4.2 allows independent development and deployment of different ACBFs since they all share a common feasible solution at $y_a = 0$. This allows for the addition of new ACBFs and adjustment of the existing ones in a practical way.*

Once the ACBFs $b_1(x), b_2(x), \ldots, b_n(x)$ are designed to capture different requirements of the shared control scheme, such as keeping a high human operator awareness and trust, and limiting workload, the safe set is defined as

$$C \triangleq \{x \in \mathbb{R}^n \,|\, b_i(x) \geq 0,\ \forall i \in \{1, 2, \ldots, n\}\}, \tag{4.15}$$

and the inequality (4.14) must be satisfied for all $x \in C$.

Remark 4.3. *The safe set C in (4.15) needs to be defined carefully: If it is too "loose" or overly permissive, the system may be labeled as "safe" under unrealistic conditions. For instance, using a single barrier function to enforce high human attention might classify highly oscillatory and potentially unstable behaviors as safe, simply because the barrier condition is met. Therefore, the safe set should be specified tightly by various ACBFs to reflect required constraints and ensure that the system's behavior remains acceptable across all relevant dimensions.*

Following the design of ACBFs, a safe automation intervention can be found by minimizing the norm between nominal automation assistance $y_a(t)$ and a safe one. Let $\tilde{y}_a \in Y_a$ be the optimization variable, representing a candidate safe version of $y_a(t)$. The optimal safe automation automation assistance input is then obtained as

$$
\begin{aligned}
y_a^*(t) = \arg\min & \|\tilde{y}_a - y_a(t)\|^2 \\
& \text{s.t.} \\
& \tilde{y}_a \in Y_a \\
& L_f b_1(x) + L_{g_r} b_1(x) r + L_{g_a} b_1(x) \tilde{y}_a + \alpha_1(b_1(x)) \geq 0 \\
& L_f b_2(x) + L_{g_r} b_2(x) r + L_{g_a} b_2(x) \tilde{y}_a + \alpha_2(b_2(x)) \geq 0 \\
& \qquad \vdots \\
& L_f b_n(x) + L_{g_r} b_n(x) r + L_{g_a} b_n(x) \tilde{y}_a + \alpha_n(b_n(x)) \geq 0,
\end{aligned}
\tag{4.16}
$$

where $y_a^*(t)$ is the signal of automation assistance closest to the nominal automation assistance $y_a(t)$ that ensures the forward invariance of C, that is, all solutions starting at any $x(0) \in C$ satisfy $x(t) \in C$ for $\forall t \geq 0$.

A possible concern with using the safety enforcement scheme (4.16) is the uncertain nature of the human operator. The variability of the human operator's behavior (the deviation from the deterministic dynamics (4.3)) may violate the forward invariance property of the safe set C in (4.15), i.e., the shared control system may exit the safe set. Nonetheless, constraining the automation assistance using Control Barrier Functions (CBFs) still benefits real operators in practice: As an example, even if a non-ideal user becomes overly distracted, the assistance pushes the interaction toward the attention profile of a well-trained operator. However, if staying within the exact boundaries of the safety set C is critical, despite uncertain human inputs, then the approach presented in this paper can be extended to include robust barrier methods. These robust barrier methods follow similar principles with the developments presented here but incorporate uncertainty models [54].

4.2 Case Study

4.2.1 Shared Control System Description

Consider a pitch tracking task, where the pilot makes the aircraft follow a pitch angle reference. The linearized plant dynamics representing the aircraft, where input is the pitch rate command and output is the pitch angle, is given by

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t), \tag{4.17a}$$

$$y_p(t) = C_p x_p(t), \tag{4.17b}$$

where $x_p \in \mathbb{R}^{n_p}$ is the plant state vector, $u_p \in \mathbb{R}$ is the pitch rate command, $y_p \in \mathbb{R}$ is the plant output (pitch angle), $A_p \in \mathbb{R}^{n_p \times n_p}$ is the known system matrix, $B \in \mathbb{R}^{n_p}$ is the known control input matrix, and $C \in \mathbb{R}^{1 \times n_p}$ is the known output matrix.

The human pilot is described by a linear model, given as [55]

$$k_p \frac{T_p s + 1}{T_z s + 1}, \tag{4.18}$$

where k_p, T_p and T_z are positive scalars. The model in (4.18) can be written in state space form as

$$\dot{x}_h(t) = a_h x_h(t) + b_h(r(t) - y_p(t)), \tag{4.19a}$$

$$y_h(t) = c_h x_h(t) + d_h(r(t) - y_p(t)), \tag{4.19b}$$

$x_h \in \mathbb{R}$ is the human state, $y_h \in \mathbb{R}$ is the human output, and $r \in \mathbb{R}$ is the reference signal. Parameters a_h, b_h, c_h, and d_h are real valued scalars. The plant input u_p in (4.17) is given as

$$u_p(t) = y_h(t) + y_a(t), \tag{4.20}$$

where the human command y_h is defined in (4.19b), and y_a is the automation assistant's signal (see Fig. 4.1). Using (4.17)-(4.20), the overall shared control

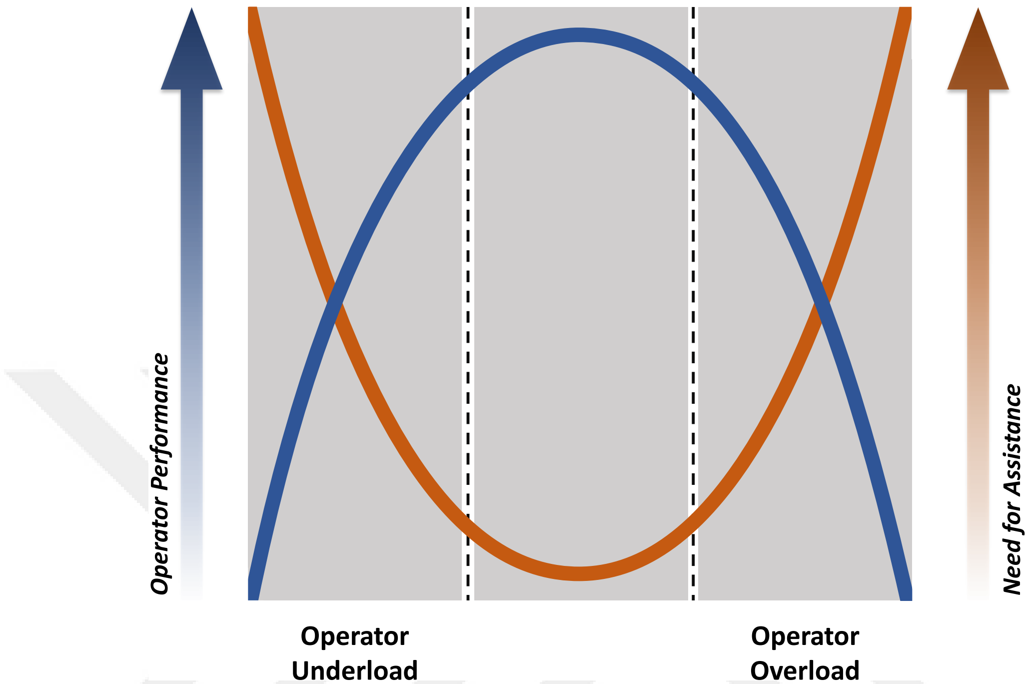

Figure 4.2: Relation between operator workload, performance, and the need for assistance [2].

system dynamics can be written as

$$\dot{x} = \begin{bmatrix} \dot{x}_p \\ \dot{x}_h \end{bmatrix} = \begin{bmatrix} A_p - B_p d_h C_p & B_p c_h \\ -b_h C_p & a_h \end{bmatrix} \begin{bmatrix} x_p \\ x_h \end{bmatrix} + \begin{bmatrix} B_p & B_p d_h \\ 0 & b_h \end{bmatrix} \begin{bmatrix} y_a \\ r \end{bmatrix}, \quad (4.21)$$

where $x = [x_p^T \quad x_h]^T \in \mathbb{R}^{n_p+1}$.

4.2.2 Arbitration Control Barrier Function Design

We design a "safety enforcement" mechanism (see Fig. 4.1) to ensure that the pilot workload stays within a prescribed interval in the presence of the automation assistance. The workload should not be reduced too much to eliminate attention loss, and increased unnecessarily to prevent fatigue (see Fig. 4.2 as an example for operator performance vs. workload curve) [56].

For simplicity, we define the pilot workload as summation of the pilot command and its time derivative as

$$w_p(t) = |y_h(t)| + |\dot{y}_h(t)|. \tag{4.22}$$

We define two barrier functions that limit the pilot workload from above and below as

$$b_1(t) = w_{max} - w_p(t), \tag{4.23a}$$

$$b_2(t) = w_p(t) - w_{min}, \tag{4.23b}$$

respectively, where w_{max} and w_{min} are upper and lower limits for the workload. Using (4.21), the inequality (4.14) can be written for b_1 and b_2 as

$$\frac{\partial b_1}{\partial x}\begin{bmatrix} A_p - B_p d_h C_p & B_p c_h \\ -b_h C_p & a_h \end{bmatrix}\begin{bmatrix} x_p \\ x_h \end{bmatrix} + \frac{\partial b_1}{\partial x}\begin{bmatrix} B_p d_h \\ b_h \end{bmatrix} r + \frac{\partial b_1}{\partial r}\dot{r} + \frac{\partial b_1}{\partial \dot{r}}\ddot{r} + \alpha_1(b_1) \geq 0, \tag{4.24a}$$

$$\frac{\partial b_2}{\partial x}\begin{bmatrix} A_p - B_p d_h C_p & B_p c_h \\ -b_h C_p & a_h \end{bmatrix}\begin{bmatrix} x_p \\ x_h \end{bmatrix} + \frac{\partial b_2}{\partial x}\begin{bmatrix} B_p d_h \\ b_h \end{bmatrix} r + \frac{\partial b_2}{\partial r}\dot{r} + \frac{\partial b_2}{\partial \dot{r}}\ddot{r} + \alpha_2(b_2) \geq 0, \tag{4.24b}$$

Note that $C_p B_p = 0$, since the plant output (pitch angle) is not affected by the input directly, but through pitch rate, such that the input and output directions are orthogonal. Differentiating (4.22) with respect to x, r, and $\dot{r}$, we obtain that

$$\begin{aligned}\frac{\partial w_p}{\partial x} &= \frac{\partial y_h}{\partial x}\operatorname{sign}(y_h) + \frac{\partial \dot{y}_h}{\partial x}\operatorname{sign}(\dot{y}_h) = \begin{bmatrix} -d_h C_p \\ c_h \end{bmatrix}\operatorname{sign}(y_h) \\ &+ \begin{bmatrix} -c_h b_h C_p - d_h C_p A_p \\ c_h a_h \end{bmatrix}\operatorname{sign}(\dot{y}_h),\end{aligned} \tag{4.25}$$

$$\begin{aligned}\frac{\partial w_p}{\partial r} &= \frac{\partial y_h}{\partial r}\operatorname{sign}(y_h) + \frac{\partial \dot{y}_h}{\partial r}\operatorname{sign}(\dot{y}_h) \\ &= d_h \operatorname{sign}(y_h) + c_h b_h \operatorname{sign}(\dot{y}_h),\end{aligned} \tag{4.26}$$

$$\frac{\partial w_p}{\partial \dot{r}} = \frac{\partial y_h}{\partial \dot{r}}\operatorname{sign}(y_h) + \frac{\partial \dot{y}_h}{\partial \dot{r}}\operatorname{sign}(\dot{y}_h) = d_h \operatorname{sign}(\dot{y}_h). \tag{4.27}$$

Remark 4.4. *As described in Section 4.1, ACBFs should be defined such that whenever $y_a(t) = 0$, the system should be safe, which means the workload should stay between w_{min} and w_{max}. When there is no automation intervention, the system is excited only by the reference signal. For simplicity, we assume that the reference signal is a sine wave. Then, in steady state, a sinusoidal response is expected since the overall shared control system is linear. For such a case, $w_p(t)$ stays in a fixed range (see (4.22)), and we assume that is the optimum workload range for a well trained human operator.*

4.2.3 Simulations

For simulations and experiments, we consider longitudinal flight dynamics of a Boeing 747, cruising in level flight at an altitude of 40kft and a velocity of 774ft/sec. State and control input matrices for this plant is given as [57]

$$A_p = \begin{bmatrix} -0.003 & 0.039 & 0 & -0.322 \\ -0.065 & -0.319 & 7.740 & 0 \\ 0.020 & -0.101 & -0.429 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{4.28a}$$

$$B_p = \begin{bmatrix} 0.010 & -0.180 & -1.160 & 0 \end{bmatrix}^T. \tag{4.28b}$$

The state vector is

$$x_p(t) = \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) & x_4(t) \end{bmatrix}^T, \tag{4.29}$$

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 x and z axes, the pitch rate (crad/sec), and the pitch angle of the aircraft (crad), respectively. The control input $u(t)$ is the elevator deflection (crad). The parameters of the human pilot model in (4.18) are selected as $T_p = 1$, $T_z = 2$, and $k_p = 4$. For barriers in (4.23), $w_{max} = 0.08$ and $w_{min} = 0.04$ values are selected considering the nominal workload range of the aforementioned pilot model for task reference. Moreover, class Kappa functions in (4.24) are selected as $\alpha_1(b(x))) = \alpha_2(b(x)) = b(x)$. Reference signal given is set as $r(t) = 0.1\sin(t)$.

The assistance system (see Fig. 4.1) is selected as a replica of the human model, aiming to enhance pilot performance and reduce workload. This deliberately simple assistance design is not meant to be sophisticated. It only serves to illustrate the CBF-based safety enforcer's behavior. For completeness, we also examine a scenario where automation assistance malfunctions, simulated by scaling the assistance output by -0.3. Safety enforcement is set to give zero output for $t < 10s$ to allow the system to reach sustained tracking conditions.

Fig. 4.3 shows the pilot input to the system, y_h, when the automation assistance functions as intended. In the absence of safety enforcement, there is a substantial reduction in pilot input compared to the pilot-alone scenario. However, when safety enforcement is applied, the pilot input appears as a relaxed version of the pilot-only case, maintaining similar characteristics. In contrast, Fig. 4.4 presents the pilot input in the presence of a malfunction. Here, without safety enforcement, the pilot input becomes noticeably sharper and higher in magnitude, indicating increased pilot effort in response to the faulty automation behavior. With safety enforcement, however, the pilot inputs do not significantly deviate from the pilot-only case.

A similar trend is observed in the tracking error. Fig. 4.5 shows that, in the absence of safety enforcement, the assistance system reduces the pilot's tracking error during nominal operation. When safety enforcement is applied, the improvement is more subtle but still evident compared to the pilot-only case. In the malfunction scenario shown in Fig. 4.6, the assistance signal increases the tracking error when safety enforcement is not in place. However, with safety enforcement implemented, the error remains comparable to that of the pilot-only operation.

By directly examining the value of the workload function defined in (4.22), Fig. 4.7 illustrates the workload profile when the assistance system operates as intended. It is evident that, although the assistance improves performance, the absence of safety enforcement causes the pilot workload to drop below the desired range. In contrast, Fig. 4.8 shows that in the presence of a malfunction, the automation assistance leads to a considerable increase in workload. In both

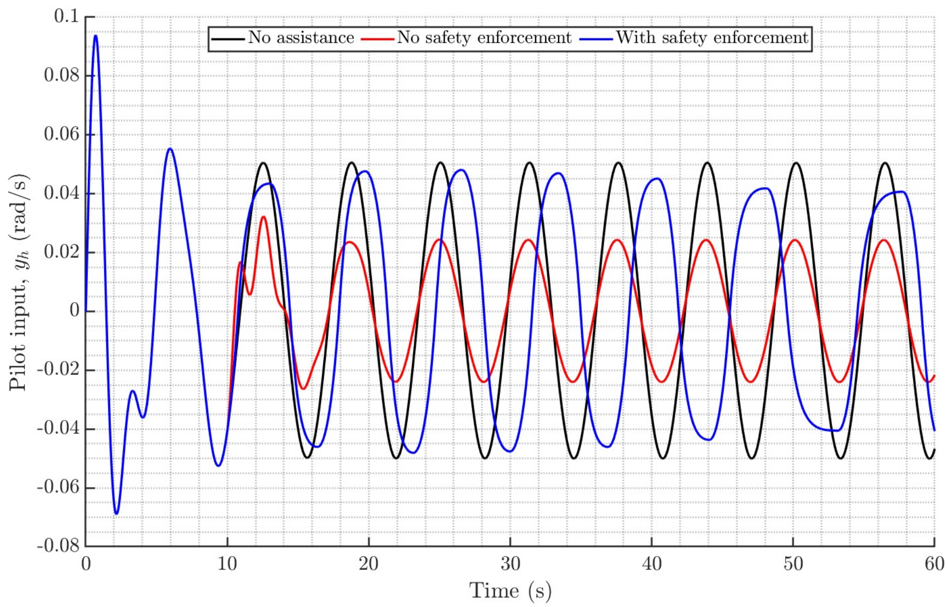

Figure 4.3: Input given to the system by the pilot model, $y_h(t)$, when automation assistance works expectedly.

scenarios, the implementation of safety enforcement successfully maintains the pilot workload within the range observed during pilot-only operation.

Table 4.1 shows the distribution of workload and the root mean square (RMS) of tracking error for each scenario. As expected, when the pilot model operates the aircraft without any assistance (first row), the workload always stays in the nominal range with a mediocre tracking performance. Automation assistance is shown to improve the performance drastically (second row) but decreases the workload beyond the lower limit, which may cause a loss of situational awareness. In the case of a malfunction (third row), the assistance increases the workload considerably and causes performance degradation. With the inclusion of the safety enforcer (see Fig. 4.1), the desired workload range is restored (fourth row), even in the case of an automation assistance malfunction (fifth row), while the tracking performance becomes similar to the case where the aircraft is operated by the pilot only without any assistance.

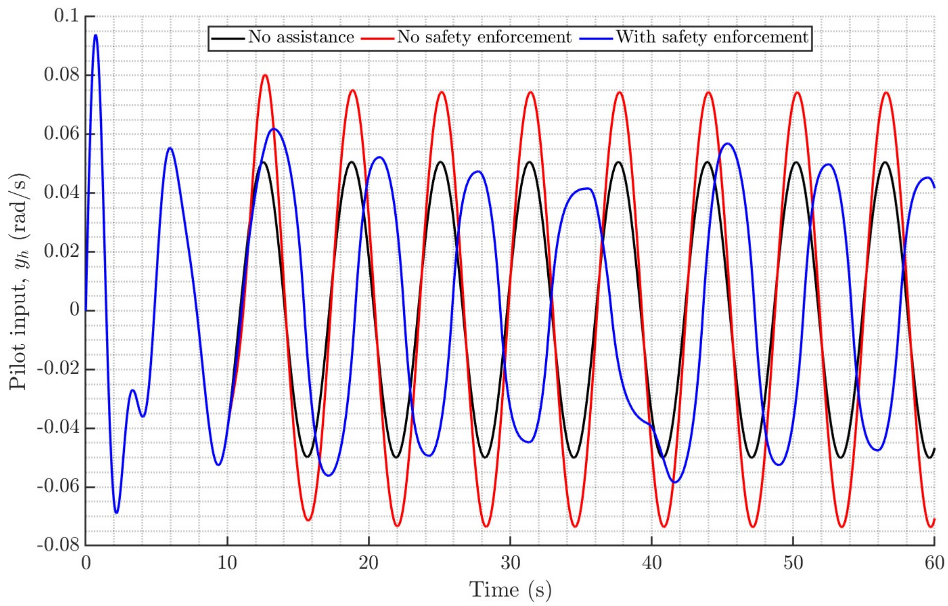

Figure 4.4: Input given to the system by the pilot model, $y_h(t)$, in the event of malfunction.

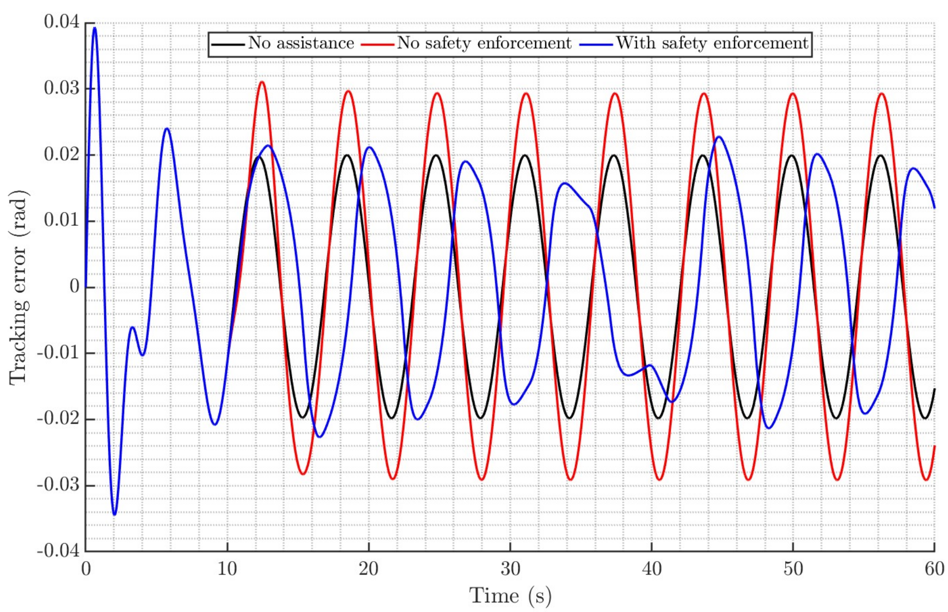

Figure 4.5: Tracking error for simulations, $e(t) = r(t) - y_p(t)$, when automation assistance works expectedly.

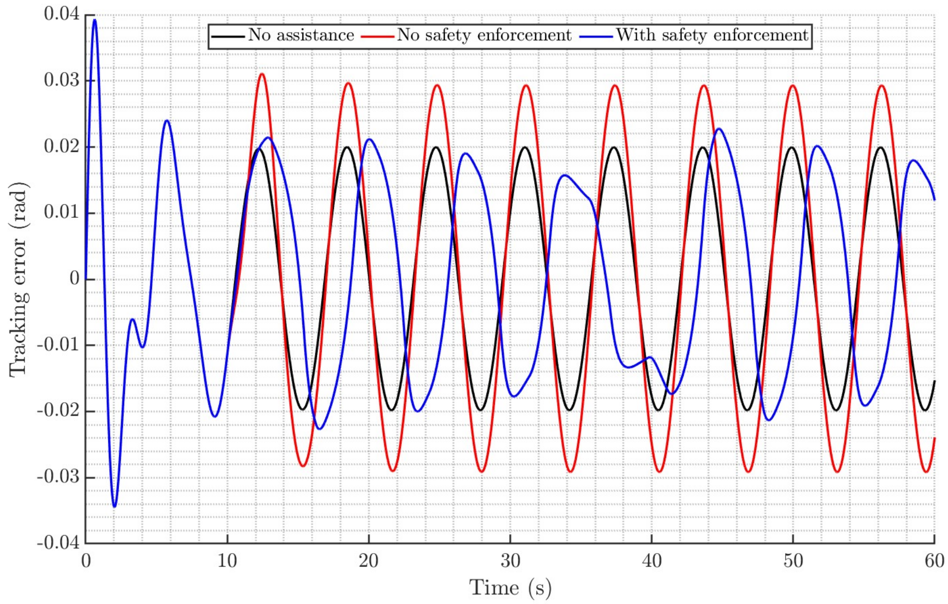

Figure 4.6: Tracking error for simulations, $e(t) = r(t) - y_p(t)$, in the event of malfunction.

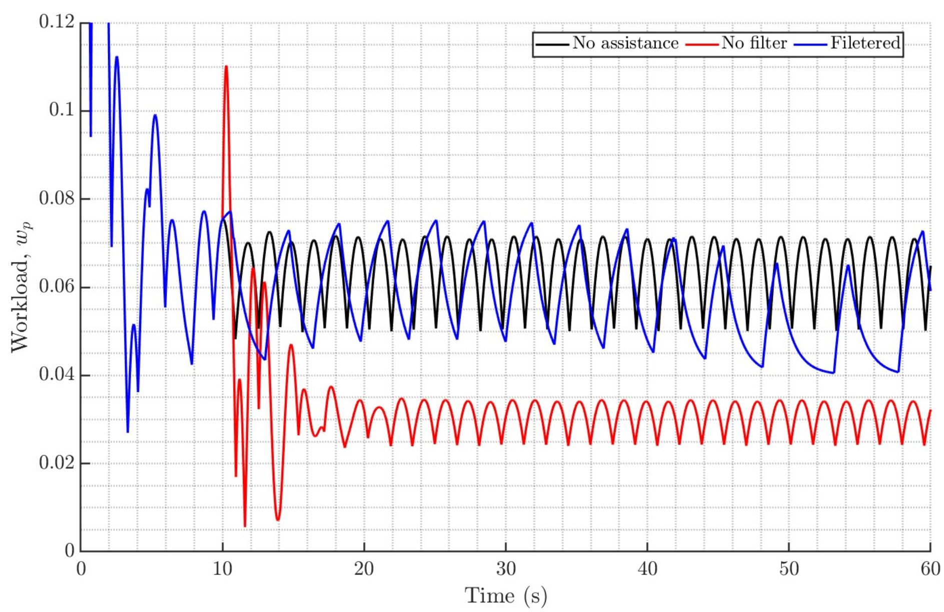

Figure 4.7: Workload of the pilot model, $w_p(t)$, when automation assistance works expectedly.

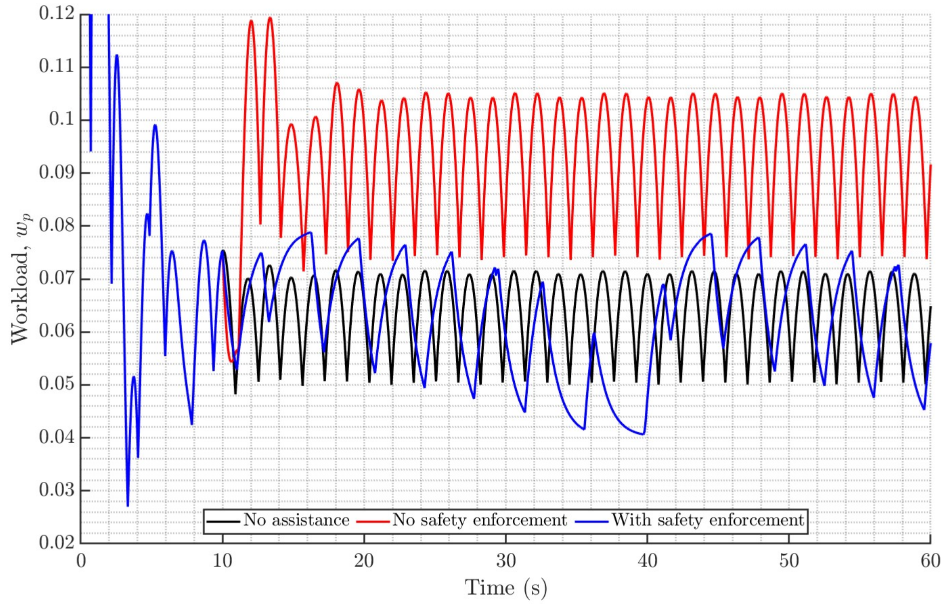

Figure 4.8: Workload of the pilot model, $w_p(t)$, in the event of malfunction.

Table 4.1: Simulations: Distribution of workload (w_p) and RMSE of tracking error ($r - y_p$) for $t > 10\,\text{s}$ (SE denotes safety enforcement).

	$w_p < w_{\text{min}}$	$\leq w_p \leq$	$w_{\text{max}} < w_p$	RMSE
`No assis.`	0.00%	100.00%	0.00%	0.014
`No SE`	94.50%	4.48%	1.02%	0.007
`No SE-Malf.`	0.00%	12.79%	87.21%	0.021
`With SE.`	0.00%	100.00%	0.00%	0.013
`With SE-Malf.`	0.00%	100.00%	0.00%	0.015

4.2.4 Experiments

The flight control task is evaluated over five experimental trials, and the results are presented as averaged values. During the task, the human operator monitors the aircraft's pitch angle and a reference trajectory via a display, aiming to minimize the pitch angle tracking error. In the experiments, the internal state $x_h(t)$ of the human operator is not directly measurable, but it is required for safety enforcement as defined in (4.24). Therefore, it is estimated using the expression in (4.19b). To mitigate the effects of measurement noise, the time derivative $\dot{y}_h(t)$ is computed using a filtered derivative of the human input signal $y_h(t)$, which is

Table 4.2: Experiments: Distribution of workload (w_p) and RMSE of tracking error ($r - y_p$) for $t > 10\,\text{s}$, averaged over participants (SE denotes safety enforcement).

	$w_p < w_{\min}$	$\leq w_p \leq$	$w_{\max} < w_p$	RMSE
`No assist.`	6.23%	31.66%	62.12%	0.022
`No SE`	49.34%	25.27%	25.39%	0.014
`No SE-Malf.`	1.65%	13.81%	84.54%	0.032
`With SE`	6.87%	28.18%	64.95%	0.013
`With SE-Malf.`	5.22%	26.50%	68.28%	0.010

provided through the joystick.

Table 4.2 shows workload distributions during flight control tasks, averaged over five experiments. Although experimental results show a larger spread compared to simulations, when the safety enforcement is not implemented, it causes more operator underload when it works as expected, and causes operator overload when there is a malfunction.

When tracking errors are observed, it is clear that the worst case happens when there is a malfunction of the assistance without filtering, followed by a mediocre performance when the human operator is alone in operation. The other three cases are comparable, with a well-working assistance giving a good performance, but with the safety enforcement, the best performances are achieved.

Fig. 4.9 offers a closer look at the workload distribution. It is clear that this automation assistance causes a significant shift in workload distribution, whether it works in an expected manner or with a malfunction. Former marks a shift to the underload region, and the latter marks a shift to the overload region. With the proposed safety enforcement, however, the workload distribution is restored to the desired region where the pilot is alone in operation, with a performance increase, as indicated in Table 4.2.

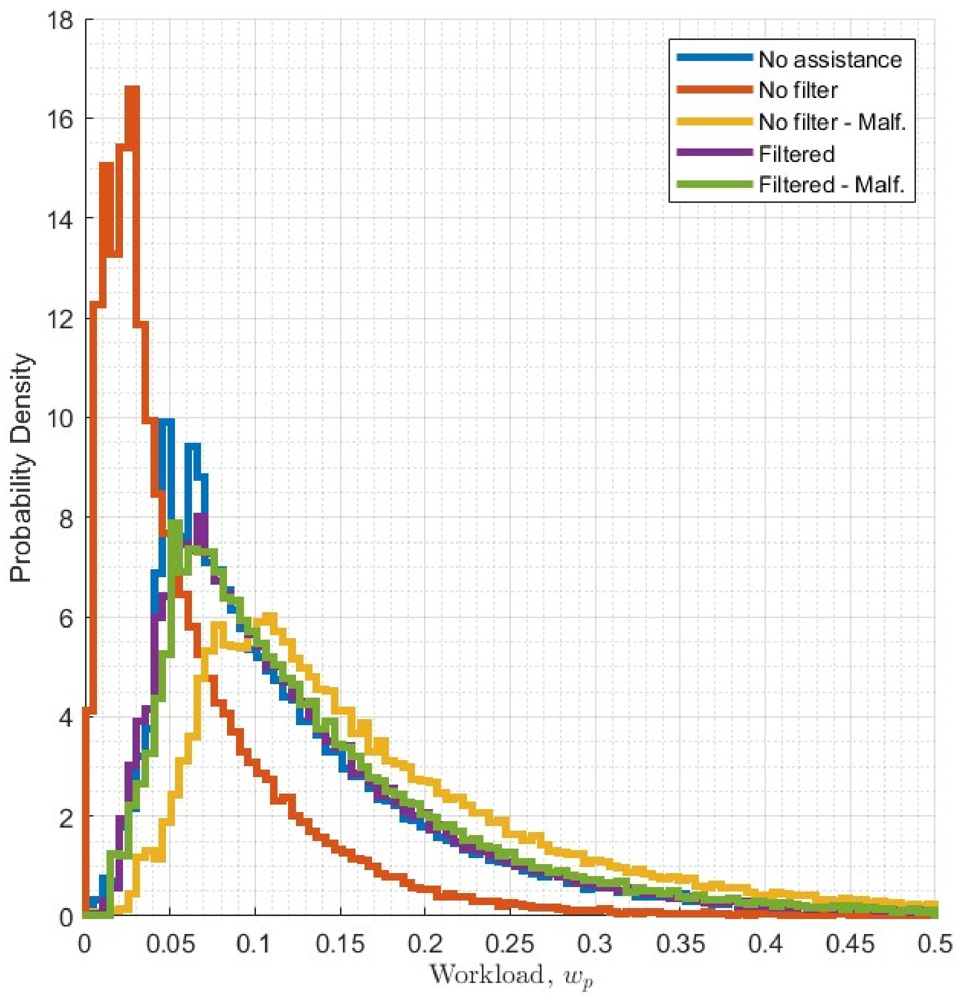

Figure 4.9: Probability density of workload (w_p) for $t > 10$s, averaged over participants for each configuration.

4.3 Conclusion

This work reinterprets CBFs within the broader context of shared control systems by extending their role from safeguarding plant-only dynamics to enforcing shared control metrics. By formulating shared control metrics as CBF constraints, we enable an interpretable and computationally efficient arbitration mechanism. The proposed approach is modular and automation-agnostic, and it opens the door for future integration of dynamic models capturing shared control metrics, such as trust, attention, and workload, thereby strengthening its applicability in diverse real-world scenarios.

Chapter 5

Conclusion

This thesis establishes frameworks for shared control. In the first framework, an automation assistance system is proposed that uses an LSTM network trained on an adaptive pilot model to correct pilot errors while achieving better flight control tracking and reduced pilot workload. Then, a second framework is built upon this framework where a modulation mechanism maintains operator authority and pilot intent alignment even when assistance fails or user intention estimations become faulty. The combination of these features enables the development of operator assistance systems which demonstrate enhanced resilience and adaptability. Moreover, the thesis presents an innovative shared control arbitration method based on Control Barrier Functions (CBFs) which integrates human-related metrics like attention and trust and workload directly into constraint equations. The proposed control framework eliminates system-specific complex arbitration methods by providing an automation-independent solution which represents shared control metrics through inequality constraints. The insights gained from this work pave the way for safe and efficient integration of human and automated control inputs which enhances both theoretical and practical aspects of shared control systems.

Bibliography

[1] M. Y. Uzun, E. Inanc, and Y. Yildiz, "Enhancing human operator performance with long short-term memory networks in adaptively controlled systems," *IEEE Control Systems Letters*, 2023.

[2] R. Hoeger, A. Amditis, M. Kunert, A. Hoess, H.-P. Krueger, A. Bartels, and A. Beutner, "Highly automated vehicles for intelligent transport: Haveit approach,"

[3] W. D. Nothwang, M. J. McCourt, R. M. Robinson, S. A. Burden, and J. W. Curtis, "The human should be part of the control loop?," in *Proc. Resilience Week (RWS)*, pp. 214–220, Aug. 2016.

[4] W. Li, D. Sadigh, S. S. Sastry, and S. A. Seshia, "Synthesis for human-in-the-loop control systems," in *Proc. Int. Conf. Tools Algorithms Construct. Anal. Syst*, pp. 470–484, Cham, Switzerland: Springer, 2014.

[5] E. Eraslan, Y. Yildiz, and A. M. Annaswamy, "Shared control between pilots and autopilots: An illustration of a cyberphysical human system," *IEEE Control Systems Magazine*, vol. 40, no. 6, pp. 77–97, Dec. 2020.

[6] D. W. Van Der Wiel, M. M. van Paassen, M. Mulder, M. Mulder, and D. A. Abbink, "Driver adaptation to driving speed and road width: Exploring parameters for designing adaptive haptic shared control," in *Proc. IEEE Conf. Syst., Man, Cybern.*, pp. 3060–3065, 2015.

[7] B. T. Thomsen, A. M. Annaswamy, and E. Lavretsky, "Shared control between adaptive autopilots and human operators for anomaly mitigation," *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 353–358, 2019.

[8] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?," *Cognition, Technology & Work*, vol. 14, pp. 19–28, 2012.

[9] P. Owan, J. Garbini, and S. Devasia, "Uncertainty-based arbitration of human-machine shared control," *arXiv preprint arXiv:1511.05996*, 2015.

[10] V. Izadi, D. Saraphis, and A. H. Ghasemi, "Dynamic control authority arbitration for resolving a reverse-intent conflict in a haptic shared control paradigm," *IFAC-PapersOnLine*, vol. 55, no. 37, pp. 512–517, 2022.

[11] D. T. McRuer and H. R. Jex, "A review of quasi-linear pilot models," *IEEE Trans. Hum. Factors Electron.*, no. 3, pp. 231–249, Sep. 1967.

[12] S. Xu, W. Tan, A. V. Efremov, L. Sun, and X. Qu, "Review of control models for human pilot behavior," *Annual Reviews in Control*, vol. 44, pp. 274–291, 2017.

[13] S. S. Tohidi and Y. Yildiz, "A control theoretical adaptive human pilot model: Theory and experimental validation," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 6, pp. 2585–2597, 2022.

[14] A. Habboush and Y. Yildiz, "An adaptive human pilot model for adaptively controlled systems," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 1964–1969, 2021.

[15] A. Habboush and Y. Yildiz, "An adaptive pilot model with reaction time-delay," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 38–43, IEEE, 2022.

[16] H. Taghavifar, "Neural network autoregressive with exogenous input assisted multi-constraint nonlinear predictive control of autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6293–6304, Jul. 2019.

[17] S. Patel, A. Sarabakha, D. Kircali, and E. Kayacan, "An intelligent hybrid artificial neural network-based approach for control of aerial robots," *J. Intell. Robot. Syst.*, vol. 97, pp. 387–398, May 2019.

[18] E. Inanc, A. Habboush, Y. Gurses, Y. Yildiz, and A. M. Annaswamy, "Neural network adaptive control with long short-term memory," *International Journal of Adaptive Control and Signal Processing*, 2023.

[19] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*, pp. 3420–3431, Ieee, 2019.

[20] B. He, M. Ghasemi, U. Topcu, and L. Sentis, "A barrier pair method for safe human-robot shared autonomy," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2854–2861, IEEE, 2021.

[21] J. Dallas, J. Talbot, M. Suminaka, M. Thompson, T. Lew, G. Orosz, and J. Subosits, "Control barrier functions for shared control and vehicle safety," *arXiv preprint arXiv:2503.19994*, 2025.

[22] W. Qin, H. Yi, Z. Fan, and J. Zhao, "Haptic shared control framework with interaction force constraint based on control barrier function for teleoperation," *Sensors (Basel, Switzerland)*, vol. 25, no. 2, p. 405, 2025.

[23] K. Shi, J. Chang, S. Feng, Y. Fan, Z. Wei, and G. Hu, "Safe human dual-robot interaction based on control barrier functions and cooperation functions," *IEEE Robotics and Automation Letters*, 2024.

[24] S. Ejaz and M. Inoue, "Trust-aware safe control for autonomous navigation: Estimation of system-to-human trust for trust-adaptive control barrier functions," *IEEE Transactions on Control Systems Technology*, vol. 33, no. 4, pp. 1151–1163, 2025.

[25] M. Yusuf Uzun, E. Inanc, and Y. Yildiz, "A robust human-autonomy collaboration framework with experimental validation," *IEEE Control Systems Letters*, vol. 8, pp. 2313–2318, 2024.

[26] C. Sentouh, A.-T. Nguyen, M. A. Benloucif, and J.-C. Popieul, "Driver-automation cooperation oriented approach for shared control of lane keeping assist systems," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 1962–1978, 2018.

[27] J. Jiang and A. Astolfi, "Shared-control for a rear-wheel drive car: Dynamic environments and disturbance rejection," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 723–734, 2017.

[28] A. M. Annaswamy and Y. Yildiz, "Cyber-physical-human systems," in *Encyclopedia of systems and control*, pp. 497–508, Springer, 2021.

[29] 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 *Eur. Control Conf.*, pp. 1–6, 2023.

[30] H. K. Khalil, *Nonlinear Systems.* Upper Saddle River, NJ, USA: Prentice Hall, 3 ed., 2002.

[31] N. Kapoor, A. R. Teel, and P. Daoutidis, "An anti-windup design for linear systems with input saturation," *Automatica*, vol. 34, no. 5, pp. 559–574, 1998.

[32] S. P. Karason and A. M. Annaswamy, "Adaptive control in the presence of input constraints," in *Proc. Amer. Control Conf.*, pp. 1370–1374, 1993.

[33] A. Graves, *Long Short-Term Memory*, pp. 37–45. Berlin, Germany: Springer, 2012.

[34] M. Schwager, A. M. Annaswamy, and E. Lavretsky, "Adaptation-based reconfiguration in the presence of actuator failures and saturation," in *Proc. Amer. Control Conf.*, pp. 2640–2645, 2005.

[35] C. R. Hanke and D. R. Nordwall, "The simulation of a jumbo jet transport aircraft. Volume II: Modelling data," NASA, Washington, DC, USA, and Boeing Company, Chicago, IL, USA, Tech. Rep. CR-114494/D6-30643-VOL2, 1970.

[36] S. S. Tohidi, Y. Yildiz, and I. Kolmanovsky, "Adaptive control allocation for constrained systems," *Automatica*, vol. 121, p. 109161, 2020.

[37] C.-T. Chen, *Linear System Theory and Design.* USA: Oxford University Press, Inc., 3rd ed., 1998.

[38] 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.

[39] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation: dynamics, controls design, and autonomous systems.* John Wiley & Sons, 2015.

[40] Y. Yildiz, A. M. Annaswamy, D. Yanakiev, and I. Kolmanovsky, "Spark-ignition-engine idle speed control: An adaptive control approach," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 990–1002, 2010.

[41] P. Perfect, M. Jump, and M. D. White, "Handling qualities requirements for future personal aerial vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 12, pp. 2386–2398, 2015.

[42] C. E. Billings, *Human-centered aircraft automation: A concept and guidelines*, vol. 103885. National Aeronautics and Space Administration, Ames Research Center, 1991.

[43] W.-P. Brinkman, M. A. Neerincx, and H. van Oostendorp, "Cognitive ergonomics for situated human–automation collaboration," *Interacting with Computers*, vol. 23, pp. iii–iv, 07 2011.

[44] D. A. Abbink, T. Carlson, M. Mulder, J. C. De Winter, F. Aminravan, T. L. Gibo, and E. R. Boer, "A topology of shared control systems—finding common ground in diversity," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 509–525, 2018.

[45] W. Wang, J. Xi, C. Liu, and X. Li, "Human-centered feed-forward control of a vehicle steering system based on a driver's path-following characteristics," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 6, pp. 1440–1453, 2016.

[46] W. Wang, X. Na, D. Cao, J. Gong, J. Xi, Y. Xing, and F.-Y. Wang, "Decision-making in driver-automation shared control: A review and perspectives," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 5, pp. 1289–1307, 2020.

[47] C. Hu, Y. Shi, S. Ge, H. Hu, J. Zhao, and X. Zhang, "Trust-based shared control of human-vehicle system using model free adaptive dynamic programming," *IEEE Transactions on Intelligent Vehicles*, pp. 1–13, 2024.

[48] M. Benloucif, C. Sentouh, J. Floris, P. Simon, and J.-C. Popieul, "Online adaptation of the level of haptic authority in a lane keeping system considering the driver's state," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 61, pp. 107–119, 2019. Special TRF issue: Driving simulation.

[49] A.-T. Nguyen, C. Sentouh, and J.-C. Popieul, "Sensor reduction for driver-automation shared steering control via an adaptive authority allocation strategy," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 5–16, 2018.

[50] C. Hu and J. Wang, "Trust-based and individualizable adaptive cruise control using control barrier function approach with prescribed performance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6974–6984, 2021.

[51] M. Marcano, S. Díaz, J. Pérez, and E. Irigoyen, "A review of shared control for automated vehicles: Theory and applications," vol. 50, no. 6, pp. 475–491, 2020.

[52] W. Xiao, C. G. Cassandras, and C. Belta, *Safe autonomy with control barrier functions: Theory and applications*. Springer, 2023.

[53] J. C. de Winter and D. Dodou, "Preparing drivers for dangerous situations: A critical reflection on continuous shared control," in *2011 IEEE International conference on systems, man, and cybernetics*, pp. 1050–1056, IEEE, 2011.

[54] S. Wang and S. Wen, "Safe control against uncertainty: A comprehensive review of control barrier function strategies," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 11, no. 1, pp. 34–47, 2025.

[55] 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.

[56] T. Gordon and M. Lidberg, "Automated driving and autonomous functions on road vehicles," *Vehicle System Dynamics*, vol. 53, no. 7, pp. 958–994, 2015.

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