Arbitration With Control Barrier Functions for Safe Shared Control

M. Yusuf Uzun and Yildiray Yildiz, *Senior Member, IEEE*

***Abstract*—By combining automation accuracy with human adaptability, shared control provides enhanced performance and safety in dynamic, complex environments. Traditional arbitration methods for integrating automation and human inputs often rely on system-specific, parameter-dependent functions that are based on shared control metrics such as trust, workload, or attention. Meanwhile, Control Barrier Functions (CBFs) enforce safety constraints on automated systems but are typically limited to safeguarding plant states. This letter introduces a novel arbitration method based on Control Barrier Functions (CBFs), where shared control metrics such as workload, attention, and trust are expressed as real-time inequality constraints. The resulting quadratic-programming formulation determines the automation assistance input that enforces these constraints while preserving feasibility and safety. This CBF-based arbitration provides a systematic, interpretable, and scalable foundation for safe human–autonomy integration.**

***Index Terms*—Shared control, Human in the loop.**

I. Introduction

ONE OF the guiding design principles in *shared control* [1] is human-centeredness, where the human operator has the final authority [2]. It is also important that the human operator can complete the task independently, particularly when a failure and/or degradation of the automation occurs [3]. Furthermore, the transparency of automation is central, providing increased situational awareness, and enabling 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 [4].

A key task in shared control is arbitration: determining how to blend human and automation inputs. Arbitration schemes have been developed for domains such as manufacturing [5], flight control [6], and driving [7]. These schemes are often based on adaptive weightings that depend on safety metrics, distraction, workload, or trust [8], [9], [10], [11], frequently relying on hand-tuned, parameter-heavy functional forms tailored to particular system architectures [12], [13].

Rather than tuning parameter-dependent arbitration schemes, we propose a Control Barrier Function (CBF) based approach, where constraints are enforced through quadratic programming formulations [14]. By expressing shared control metrics as inequality constraints, we enable a systematic automation intervention. For example, rising cognitive workload may lead to increased attention and reduce trust simultaneously. Thanks to the CBF-based approach, these multiple metrics can be given their own bounds, 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, 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.

A related line of work introduces a performance-based trust metric as a CBF constraint for adaptive cruise control in a *traded control* setting, without simultaneous action, where either the human or the automation is in control at any given moment [15]. In most shared control applications, CBFs have been employed to prevent human operators from driving systems into unsafe regions [16], [17], [18], or see humans as passive entities that must not be harmed [19], [20]. While these works do address shared control, they use CBFs primarily to ensure safety with respect to the physical system's states. Unlike earlier work, we propose a CBF-based shared control framework that regulates the automation assistance based on shared control metrics (workload/attention/trust) while human and automation act concurrently. Our approach formulates an arbitration problem, where CBFs enforce collaborative constraints that directly reflect human-system interaction quality.

II. Framework

In this section, we introduce a safe shared-control framework, where safety is guaranteed via a careful control barrier function (CBF) design (see Fig. 1).

Received 12 September 2025; revised 10 November 2025; accepted 26 November 2025. Date of publication 10 December 2025; date of current version 17 December 2025. This work was supported by the Scientific and Technological Research Council of Turkey under Grant 121E384. Recommended by Senior Editor L. Menini. *(Corresponding author: M. Yusuf Uzun.)*

The authors are with the Department of Mechanical Engineering, Bilkent University, 06800 Ankara, Türkiye (e-mail: yusuf.uzun@bilkent.edu.tr; yyildiz@bilkent.edu.tr).

Digital Object Identifier 10.1109/LCSYS.2025.3642534

2475-1456 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on March 03,2026 at 16:11:32 UTC from IEEE Xplore. Restrictions apply.

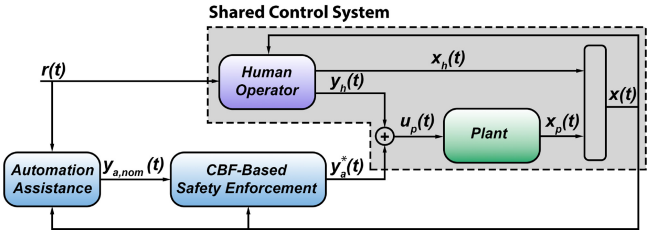

Fig. 1. Block diagram of the proposed architecture.

A. Plant Dynamics

We consider a control-affine plant

$$\dot{x}_p = f_p(x_p) + g_p(x_p)\, u_p, \tag{1}$$

where $x_p \in \mathbb{R}^{n_p}$, $u_p \in \mathbb{R}^{q_p}$, and $f_p : \mathbb{R}^{n_p} \to \mathbb{R}^{n_p}$, $g_p : \mathbb{R}^{n_p} \to \mathbb{R}^{n_p \times q_p}$ are locally Lipschitz. The plant input is taken as the sum of human and automation commands,

$$u_p \;=\; y_h + y_a, \tag{2}$$

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

Remark 1: We do not commit to a specific assistance strategy. As an illustrative example, the automation may aim to minimize the deviation between the human input y_h and a nominal ideal input $\bar{y}_h$. This yields the classical input–mixing strategy [21], $u_p = (1-\lambda)\, y_h + \lambda\, \bar{y}_h \;=\; y_h + \lambda\,(\bar{y}_h - y_h)$, where $\lambda \in [0,\,1]$ is a (possibly time-/state-dependent) mixing coefficient. Comparing with $u_p = y_h + y_a$, the assistance signal is $y_a \;=\; \lambda\,(\bar{y}_h - y_h)$.

B. Human Operator Model

We model the human operator as

$$\dot{x}_h = f_h(x_h, x_p) + g_h(x_h, x_p)\, r, \tag{3a}$$
$$y_h = h_h(x_h) + j_h(x_h)\, r, \tag{3b}$$

where $x_h \in \mathbb{R}^{n_h}$ is the human state, $x_p \in \mathbb{R}^{n_p}$ is the plant state from (1), and $f_h : \mathbb{R}^{n_h+n_p} \to \mathbb{R}^{n_h}$, $g_h : \mathbb{R}^{n_h+n_p} \to \mathbb{R}^{n_h \times q_h}$, $h_h : \mathbb{R}^{n_h} \to \mathbb{R}^{q_p}$, $j_h : \mathbb{R}^{n_h} \to \mathbb{R}^{q_p \times q_h}$ are locally Lipschitz. The reference is $r \in \mathbb{R}^{q_h}$, and thus $y_h \in \mathbb{R}^{q_p}$.

Using (1)–(3), we obtain

$$\dot{x} = \begin{bmatrix} \dot{x}_p \\ \dot{x}_h \end{bmatrix} = \begin{bmatrix} f_p(x_p) + g_p(x_p)\, h_h(x_h) \\ f_h(x_h, x_p) \end{bmatrix} + \begin{bmatrix} g_p(x_p) & g_p(x_p)\, j_h(x_h) \\ 0_{n_h \times q_p} & g_h(x_h, x_p) \end{bmatrix} \begin{bmatrix} y_a \\ r \end{bmatrix}. \tag{4}$$

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) can be identified as

$$f(x) = \begin{bmatrix} f_p(x_p) + g_p(x_p)\, h_h(x_h) \\ f_h(x_h, x_p) \end{bmatrix}, \tag{5a}$$
$$g_a(x) = \begin{bmatrix} g_p(x_p) \\ 0_{n_h \times q_p} \end{bmatrix}, \quad g_r(x) = \begin{bmatrix} g_p(x_p)\, j_h(x_h) \\ g_h(x_h, x_p) \end{bmatrix}. \tag{5b}$$

Taking the automation assistance signal y_a in (2) and the reference signal r in (3) as inputs, the overall shared-control dynamics can be obtained using (1)–(5) as

$$\dot{x} = f(x) + g_a(x) y_a + g_r(x) r. \tag{6}$$

C. Arbitration Control Barrier Functions (ACBF)

Definition 1 [22]: Consider the system $\dot{x} = f(x) + g(x)\, u$, $x \in X \subset \mathbb{R}^n$ and $u \in U \subset \mathbb{R}^q$, where $f : X \to \mathbb{R}^n$ and $g : X \to \mathbb{R}^{n \times q}$ are locally Lipschitz, and U is nonempty and compact. A continuously differentiable function $b : X \to \mathbb{R}$ defines the *safe set*

$$C \triangleq \{\, x \in X \mid b(x) \geq 0 \,\}. \tag{7}$$

The function b is a *control barrier function (CBF)* if there exists a class-$\mathcal{K}$ function α such that

$$\sup_{u \in U} \big[\, L_f b(x) + L_g b(x)\, u + \alpha(b(x)) \,\big] \;\geq\; 0, \qquad \forall x \in C, \tag{8}$$

where $L_f b$ and $L_g b$ are the Lie derivatives of b along f and g. It is assumed that $L_g b(x) \neq 0$ for all $x \in \partial C$, where $\partial C = \{\, x \in X \mid b(x) = 0 \,\}$ is the boundary of C.

Considering the shared-control dynamics (6), with the reference r treated as a known exogenous input, (8) becomes

$$\sup_{y_a \in Y_a} \big[\, L_f b(x) + L_{g_r} b(x)\, r + L_{g_a} b(x)\, y_a + \alpha(b(x)) \,\big] \;\geq\; 0 \quad \forall x \in C, \tag{9}$$

where Y_a is the compact, convex admissible assistance set containing the origin. In the shared-control context, b may encode a metric such as operator attention. Then the safe set C in (7) characterizes states for which the operator's attention is sufficient for safe operation.

As described in Definition 1, the function $b(x)$ in the inequality (9) being a CBF ensures that there exists at least one automation assistance input $y_a \in Y_a$ such that the inequality can be satisfied for each $x \in C$. 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 (7) to be characterized by a region where the human operator can keep the system within C without any help from the assistance. Moreover, 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 [23]. This sets our foundation for regulating automation assistance.

We model the barriers as $b(x, \rho)$ with $\rho \triangleq (r, \dot{r}, \ldots)$ so that the safe set $C(\rho) = \{x \mid b(x, \rho) \geq 0\}$ is *task-specific*, such that its geometry and tightness adapt to the current reference and its local features (e.g., rate, curvature). This makes the constraints reference-aware and allows the same mechanism to enforce safety consistently across different tasks/references. It is assumed that the exogenous task-level reference signal r and its derivatives up to order ν are available to the automation and $r \in C^\nu$, i.e., r is ν-times continuously differentiable.

Using Lie derivatives with respect to the x–argument (holding ρ fixed), we define

$$D_0 b(x, \rho) \;:=\; L_f b(x, \rho) \;+\; L_{g_r} b(x, \rho)\, r + \sum_{i=0}^{\nu-1} \frac{\partial b(x, \rho)}{\partial r^{(i)}}\, r^{(i+1)}, \tag{10a}$$
$$D_a b(x, \rho; y_a) \;:=\; D_0 b(x, \rho) \;+\; L_{g_a} b(x, \rho)\, y_a. \tag{10b}$$

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on March 03,2026 at 16:11:32 UTC from IEEE Xplore. Restrictions apply.

Along the closed loop with assistance y_a,

$$\frac{d}{dt}\, b\big(x(t),\rho(t)\big) \;=\; D_a b(x,\rho;y_a), \tag{11a}$$

$$\frac{d}{dt}\, b\big(x(t),\rho(t)\big)\Big|_{y_a=0} \;=\; D_0 b(x,\rho). \tag{11b}$$

Note that if $b = b(x)$, then $\frac{\partial b}{\partial r^{(i)}} \equiv 0$, $D_0 b(x) = L_f b(x) + L_{g_r} b(x)\, r$, and $D_a b(x;y_a) = D_0 b(x) + L_{g_a} b(x)\, y_a$ such that Lie-derivative expression in (9) is recovered.

Considering the shared-control system (6), we define *Arbitration Control Barrier Functions* as follows.

Definition 2: Let $C(\rho) \triangleq \{x \in \mathbb{R}^n \mid b(x,\rho) \geq 0\}$ be the safe set induced by the barrier $b = b(x,\rho)$. Then b is an *arbitration control barrier function (ACBF)* for (6) if there exists a class-$\mathcal{K}$ function α such that

$$D_0 b(x,\rho) \;+\; \alpha\big(b(x,\rho)\big) \;\geq\; 0, \qquad \forall x \in C(\rho). \tag{12}$$

That is, the ACBF condition is evaluated along the no-assistance dynamics ($y_a = 0$).

Remark 2: Definition 2 allows independent development of different ACBFs since they all share a common feasible solution at $y_a = 0$. This modularity allows independent development and tuning, and deployment of multiple ACBFs, such that adding a new barrier or tightening an existing one preserves feasibility because $y_a = 0$ always satisfies (12).

Let $b_1(x,\rho), \ldots, b_m(x,\rho)$ encode different shared-control requirements (e.g., maintaining operator awareness and trust, limiting workload). The resulting safe set is the intersection

$$C(\rho) \;\triangleq\; \bigcap_{i=1}^{m} \{x \in \mathbb{R}^n \mid b_i(x,\rho) \geq 0\}, \tag{13}$$

and the ACBF condition (12) must hold for all $x \in C(\rho)$.

Remark 3: The safe set $C(\rho)$ in (13) needs to be defined carefully: If it is 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 oscillatory behaviors as safe 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.

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

$$y_a^\star(t) = \arg\min_{y_a \in Y_a} \;\| y_a - y_{a,nom}(t) \|^2 \tag{14}$$

$$\text{s.t.} \quad D_a b_i\big(x,\rho;y_a\big) + \alpha_i\big(b_i(x,\rho)\big) \;\geq\; 0, \quad i = 1,\ldots,m,$$

where b_i are the ACBFs and α_i are class-$\mathcal{K}$ functions. Each constraint is affine in y_a (through $L_{g_a} b_i\, y_a$ in (10b)), so (14) is a convex quadratic program with a nonempty feasible set that always contains $y_a = 0$ by Definition 2. The optimizer $y_a^\star(t)$ is thus the assistance signal closest to the nominal $y_{a,nom}(t)$ that enforces the barriers. Under the ACBF conditions, applying $y_a := y_a^\star(t)$ in (6) renders the safe set $C(\rho)$ forward invariant such that for any $x(0) \in C(\rho(0))$, the trajectory satisfies $x(t) \in C(\rho(t))$ for all $t \geq 0$.

Unlike classical CBF formulations (as in Definition 1) that require $L_g b(x) \neq 0$ on the boundary of the safe set, ∂C, the ACBF condition (12) is evaluated along the no-assistance dynamics ($y_a = 0$). Therefore, if $L_{g_a} b(x,\rho) = 0$, the full constraint $D_a b(x,\rho;y_a) + \alpha(b(x,\rho)) \geq 0$ used in the QP (14) reduces to (12) since $D_a b = D_0 b$ with $L_{g_a} b(x,\rho) = 0$ (see (10b)), which is feasible by definition.

III. Case Study: A Theoretical Analysis

A. 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 elevator deflection and output is the pitch angle, is given by

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t), \tag{15a}$$

$$y_p(t) = C_p x_p(t), \tag{15b}$$

where $x_p \in \mathbb{R}^{n_p}$ is the plant state vector, $u_p \in \mathbb{R}$ is the input, $y_p \in \mathbb{R}$ is the output, $A_p \in \mathbb{R}^{n_p \times n_p}$ is the known system matrix, $B_p \in \mathbb{R}^{n_p \times 1}$ is the known control input matrix, and $C_p \in \mathbb{R}^{1 \times n_p}$ is the known output matrix. The plant input is $u_p = y_h + y_a$ as in (2), with $q_p = 1$ here.

If the original plant dynamics, denoted A_p', are not stable, or if one wishes to modify its dynamics, it is common to apply state feedback of the form $u_p = u_p' - K_p x_p$, yielding a closed-loop system with effective dynamics

$$A_p = A_p' - B_p K_p, \tag{16}$$

where $K_p \in \mathbb{R}^{1 \times n_p}$ is the feedback gain. We use A_p to denote such a modified, stable system matrix.

The human pilot is described by [24]

$$y_h(s) \;=\; k_p \frac{T_p s + 1}{T_z s + 1}\big(r(s) - y_p(s)\big), \tag{17}$$

where k_p, T_p and T_z are positive scalars. The model in (17) can be written in state space form as

$$\dot{x}_h(t) = a_h x_h(t) + b_h\big(r(t) - y_p(t)\big), \tag{18a}$$

$$y_h(t) = c_h x_h(t) + d_h\big(r(t) - y_p(t)\big), \tag{18b}$$

where $x_h \in \mathbb{R}$ 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. Using (15)–(18), the overall shared control system dynamics can be written as

$$\dot{x} = \begin{bmatrix} \dot{x}_p \\ \dot{x}_h \end{bmatrix} = \begin{bmatrix} A_p - B_p d_h C_p & B_p c_h \\ -b_h C_p & a_h \end{bmatrix} \begin{bmatrix} x_p \\ x_h \end{bmatrix} + \begin{bmatrix} B_p & B_p d_h \\ 0 & b_h \end{bmatrix} \begin{bmatrix} y_a \\ r \end{bmatrix}, \tag{19}$$

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

B. Arbitration Control Barrier Function Design

We design a "safety enforcement" mechanism (see Fig. 1) to ensure that the pilot workload stays within reasonable limits in the presence of automation assistance. The workload should neither be reduced to a level that causes attention loss, nor increased unnecessarily, which can lead to fatigue.

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on March 03,2026 at 16:11:32 UTC from IEEE Xplore. Restrictions apply.

While various workload metrics can be defined based on pilot command, command rate, tracking error, or error rate depending on the task, in this case study the workload is represented as a simple, differentiable measure of pilot control activity that captures both the effort and rate of response, defined as the sum of the squared pilot command and its time derivative as

$$w_p(x,\rho) \;=\; y_h(x,\rho)^2 \;+\; \dot{y}_h(x,\rho)^2. \tag{20}$$

We define two barriers

$$b_1(x,\rho) = w_p(x,\rho) - w_L(t), \tag{21a}$$
$$b_2(x,\rho) = w_U(t) - w_p(x,\rho), \tag{21b}$$

so that $b_1 \geq 0$ and $b_2 \geq 0$ are equivalent to $w_L \leq w_p \leq w_U$. The signals $w_U(t)$ and $w_L(t)$ are included in the state vector for notational simplicity. We keep writing $b_1 = b_1(x,\rho)$ and $b_2 = b_2(x,\rho)$, and we keep the symbol x for the augmented state, i.e., $x = [x_p^\top \quad x_h \quad w_L \quad w_U]^\top$ and $n = n_p + 3$. These states are updated as

$$\dot{w}_L(t) = D_0 w_p(x,\rho) + \gamma_1\big(w_p(x,\rho) - w_L(t)\big) - \phi\big(d(t)\big), \tag{22a}$$
$$\dot{w}_U(t) = D_0 w_p(x,\rho) - \gamma_2\big(w_U(t) - w_p(x,\rho)\big) + \phi\big(d(t)\big), \tag{22b}$$

where $d(t) \;\triangleq\; w_U(t) - w_L(t)$ is the separation, and $\gamma_1, \gamma_2 > 0$. Here $D_0 w_p(x,\rho)$ is the no-assistance workload rate (see (10a)–(10b)). The function

$$\phi(d) \;=\; \frac{c_p}{s_p}\log\Big(1 + e^{s_p\,(d_{\min} - d)}\Big), \tag{23}$$

for some $c_p, s_p, d_{\min} > 0$. Here $\phi(d)$ is a continuously differentiable surrogate of $c_p \max(0, d_{\min} - d)$ such that it is strictly decreasing in d, vanishes for large d, and activates when the separation d approaches $d_{\min}$. The constant c_p scales the effect and s_p controls the sharpness of activation. We assume an initially ordered pair of limits, $d(0) \geq 0$.

By explicitly referencing the pilot-only workload rate $D_0 w_p(x,\rho)$ in the limit dynamics (22), the ACBF constraints become adaptive to task phase and difficulty. In contrast to fixed envelopes that can be overly tight during aggressive maneuvers and overly loose in calm segments, the bounds implicitly track the human-only slope, yielding instantaneous-rate limits centered on $D_0 w_p(x,\rho)$. This baseline-referenced formulation preserves feasibility with $y_a = 0$ while preventing assistance from inducing abrupt under- or over-loading relative to what the pilot would naturally expect.

Lemma 1: Let $\gamma_1 = \gamma_2 = \gamma > 0$ and $d_{\min} \geq 0$. The separation d evolves as

$$\dot{d} \;=\; F(d) \;\coloneqq\; -\gamma\, d \;+\; 2\,\phi(d), \tag{24}$$

which admits a unique equilibrium $d^\star > 0$ with $F(d^\star) = 0$, and $d^\star$ is exponentially stable.

Proof: Subtracting (22a) from (22b) with $\gamma_1 = \gamma_2 = \gamma$ gives (24). Since ϕ is C^1, F is locally Lipschitz and solutions are unique. Because ϕ is strictly decreasing, $F'(d) = -\gamma + 2\phi'(d) < -\gamma < 0$, so F is strictly decreasing with $F(0) = 2\phi(0) > 0$ and $F(d) \to -\infty$ as $d \to \infty$; hence there is a unique $d^\star > 0$ with $F(d^\star) = 0$.

Let $e \coloneqq d - d^\star$ and choose the Lyapunov function $V(e) = \frac{1}{2}e^2$. Using $F(d^\star) = 0$, we get $\dot{V} = e\,F(d) = e(-\gamma e + 2(\phi(d) - \phi(d^\star)))$. Since ϕ is strictly decreasing, $e(\phi(d) - \phi(d^\star)) \leq 0$, with equality only at $e = 0$. Therefore $\dot{V} \leq -\gamma e^2 = -2\gamma V$, which implies $V(t) \leq e^{-2\gamma t}V(0)$ and hence $e(t) \to 0$, and (24) is exponentially stable. ∎

Theorem 1: Consider the shared-control dynamics (19), the workload definition (20), the barriers (21), and the update laws (22). Then b_1 and b_2 are valid ACBFs (see Definition 2) for the shared-control system and there exist class–$\mathcal{K}$ functions $\alpha_1(s) = \gamma_1 s$ and $\alpha_2(s) = \gamma_2 s$ such that, for all states x in the safe set, (12) holds, i.e., $D_0 b_i(x,\rho) + \alpha_i(b_i(x,\rho)) \geq 0, \quad \forall x \in C(\rho)$, for $i = 1, 2$.

Proof: By Definition 2, the ACBF condition is evaluated with assistance disengaged, i.e., along the no-assistance dynamics ($y_a = 0$), so $D_a(\cdot) = D_0(\cdot)$. From (21) and the chain rule applied to functions of (x, ρ),

$$D_0 b_1 \;=\; D_0 w_p \;-\; \dot{w}_L, \quad D_0 b_2 \;=\; \dot{w}_U \;-\; D_0 w_p. \tag{25}$$

Evaluating (22) with $D_a \to D_0$ gives

$$\dot{w}_L = D_0 w_p + \gamma_1\big(w_p - w_L\big) - \phi\big(d\big), \tag{26}$$
$$\dot{w}_U = D_0 w_p - \gamma_2\big(w_U - w_p\big) + \phi\big(d\big). \tag{27}$$

Subtracting (26) from $D_0 w_p$ and substituting $b_1 = w_p - w_L$ yields

$$D_0 b_1 \;=\; D_0 w_p - \dot{w}_L \;=\; -\,\gamma_1\, b_1 \;+\; \phi(d). \tag{28}$$

Similarly, subtracting $D_0 w_p$ from (27) and using $b_2 = w_U - w_p$ gives

$$D_0 b_2 \;=\; \dot{w}_U - D_0 w_p \;=\; -\,\gamma_2\, b_2 \;+\; \phi(d). \tag{29}$$

Let $\alpha_1(s) = \gamma_1 s$ and $\alpha_2(s) = \gamma_2 s$. Then, for $i = 1, 2$,

$$D_0 b_i + \alpha_i(b_i) \;=\; \phi\big(d\big) \;\geq\; 0, \tag{30}$$

since $\phi(\cdot) \geq 0$ by construction (see (23)). ∎

Remark 4: At each instant, the safety filter solves (14). From the barriers (21) with (22), $D_a b_1 = D_a w_p - \dot{w}_L$ and $D_a b_2 = \dot{w}_U - D_a w_p$, so the constraints in (14) are equivalent to the instantaneous workload–rate band $-\phi(d) \leq D_a w_p(x,\rho; y_a) - D_0 w_p(x,\rho) \leq \phi(d)$, or equivalently, $D_0 w_p(x,\rho) - \phi(d(t)) \leq \frac{d}{dt} w_p(x,\rho) \leq D_0 w_p(x,\rho) + \phi(d(t))$. By Lemma 1, d converges exponentially to $d^\star$, so most operation occurs with $d(t) \approx d^\star$ and the band tightens to $|\dot{w}_p - D_0 w_p(x,\rho)| \leq \phi(d^\star)$.

The parameter $d_{\min}$ in (23) shapes ϕ, which in turn determines the equilibrium separation $d^\star$. Larger $d_{\min}$ results in larger $d^\star$, which increases the available margin, enlarging the set of states where nonzero assistance remains feasible under (12). Conversely, $d_{\min} = 0$ imposes smallest buffer, producing the tightest margins and thus the most conservative assistance. As discussed in Remark 3, such a choice would restrict the automation from providing sufficient support, whereas selecting a large d_{min} may cause $|\dot{w}_p - D_0 w_p(x,\rho)| \ll \phi(d^\star)$, which allows excessively high workload levels. In extreme cases, this could lead the automation assistance to override the human operator's authority during malfunction conditions. Since the human operator's workload is inherently limited at the manipulator level during task execution, d_{min}

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on March 03,2026 at 16:11:32 UTC from IEEE Xplore. Restrictions apply.

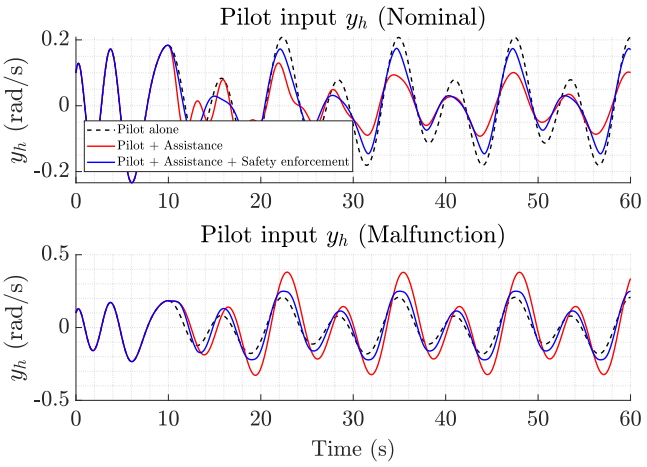

Fig. 2. Input given to the system by the pilot model, $y_h(t)$.

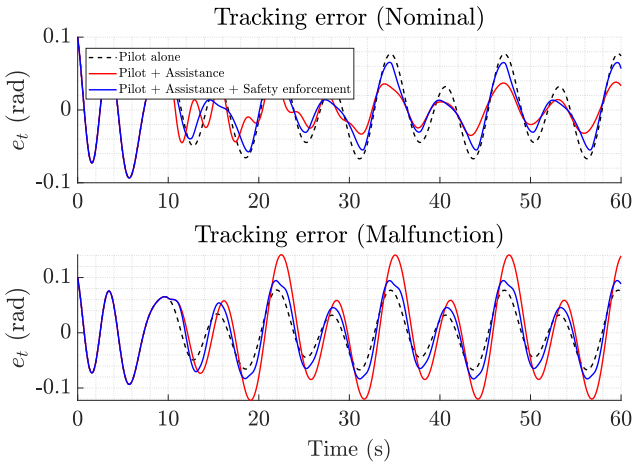

Fig. 3. Tracking error, $e_t(t) = r(t) - y_p(t)$. RMS(e_t) [rad], Nominal: P= 0.045, P+A= 0.031, P+A+S= 0.037. Malfunction: P= 0.045, P+A= 0.072, P+A+S= 0.056.

can be selected to represent a limited fraction of the operator's maximum feasible workload, thereby defining a realistic and safe buffer.

C. Simulations

For simulations, we consider longitudinal flight dynamics of a Boeing 747, cruising in level flight at an altitude of 40kft and a velocity of 774ft/s. State and control input matrices for this plant are given as [25]

$$A_p = \begin{bmatrix} -0.003 & 0.039 & 0 & -0.322 \\ -0.065 & -0.319 & 7.740 & 0 \\ 0.020 & -0.101 & -0.429 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (31a)$$

$$B_p = \begin{bmatrix} -0.010 & 0.180 & 1.160 & 0 \end{bmatrix}^T. \quad (31b)$$

States $x_1(t)$, $x_2(t)$, $x_3(t) = q(t)$, and $x_4(t) = y_p(t) = \theta(t)$ are the components of the aircraft's velocity (ft/s) along x and z axes, the pitch rate (crad/s), and the pitch angle of the aircraft (crad), respectively. The control input $u_p(t)$ is the elevator deflection (crad). The feedback gain in (16) is selected with LQR gains $Q_{LQR} = \text{diag}([0.001, 0.01, 10, 0.1])$, and $R_{LQR} = 10$. The parameters of the human pilot model in (17) are selected as $T_p = 0.2$, $T_z = 0.6$, and $k_p = 3$. For updates of lower and upper workload limits in (22), $\gamma_1 = \gamma_2 = 0.5$ is selected, and w_L and w_U are initialized to 0 and 0.1, respectively. Moreover, d_{min} in (23) is set to 0.02, and shape parameters are set to $c_p = 1$ and $s_p = 1000$. The reference signal is set to $r(t) = 0.1(\cos(0.5t) - \sin(t))$.

The assistance system (see Fig. 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, and it only serves to illustrate the CBF-based safety enforcement's behavior. For completeness, we also examine a scenario where automation assistance malfunctions, simulated by scaling the assistance output by -0.5. Safety enforcement is set to give zero output for $t < 10s$ to allow the system to reach sustained tracking.

Fig. 2 shows the pilot input to the system, y_h, during simulations for nominal assistance and malfunctioning assistance cases. In the absence of safety enforcement, there is a substantial reduction in pilot input compared to the pilot-alone scenario for nominal case. However, when safety enforcement is applied, the pilot input appears as a relaxed version of the pilot-only case without any considerable drop. In contrast, for the malfunction scenario, without safety enforcement, the pilot input becomes 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. 3 shows that, in the nominal case, when the safety enforcement is absent, the assistance reduces the pilot's tracking error. When safety enforcement is applied, the error reduction effect decreases as expected, since the response is more balanced. In the malfunction scenario, the assistance becomes detrimental to the system and increases the tracking error. However, with safety enforcement implemented, the performance degradation due to assistance malfunction is confined to a tighter region.

Fig. 4 illustrates the workload profiles calculated using (20). The absence of safety enforcement diminishes the pilot workload significantly throughout the simulation, which may cause loss of situational awareness. In contrast, in the presence of a malfunction, the automation assistance leads to a considerable increase in workload. In both scenarios, the safety enforcement maintains the pilot workload around pilot-only case.

In Fig. 5, the pilot's workload w_p is plotted together with the adaptive bounds w_L and w_U governed by (22). These bounds are not static envelopes. Although they are anchored to the pilot-only slope $D_0 w_p(x, \rho)$, they also include feedback terms in w_p and $\phi(d)$. Hence, they change indirectly with the automation signal y_a through its effect on w_p over time. Consequently, assistance can nudge the envelope through the designed limit dynamics, preventing abrupt tightening/loosening. The ACBF inequalities (see (14)) then keep w_p confined between w_L and w_U while enforcing the instantaneous-rate band given in Remark 4. The result is an

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on March 03,2026 at 16:11:32 UTC from IEEE Xplore. Restrictions apply.

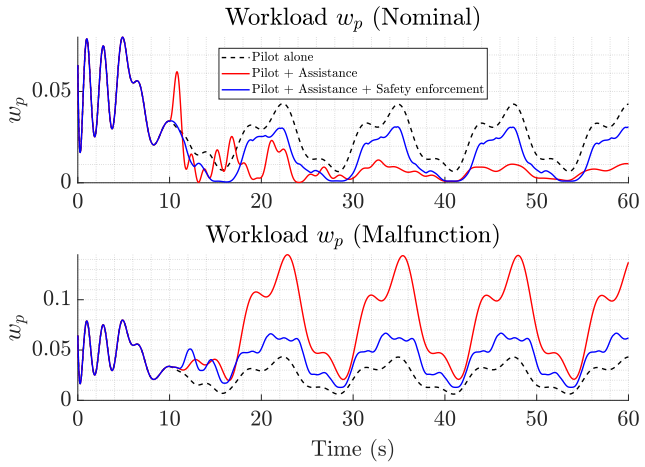

Fig. 4. Workload of the pilot model, $w_p(t)$. Nominal: RMS($|w_p^{\mathrm{P}} - w_p^{\mathrm{P+A}}|$) = 0.018, RMS($|w_p^{\mathrm{P}} - w_p^{\mathrm{P+A+S}}|$) = 0.010. Malfunction: RMS($|w_p^{\mathrm{P}} - w_p^{\mathrm{P+A}}|$) = 0.053, RMS($|w_p^{\mathrm{P}} - w_p^{\mathrm{P+A+S}}|$) = 0.018.

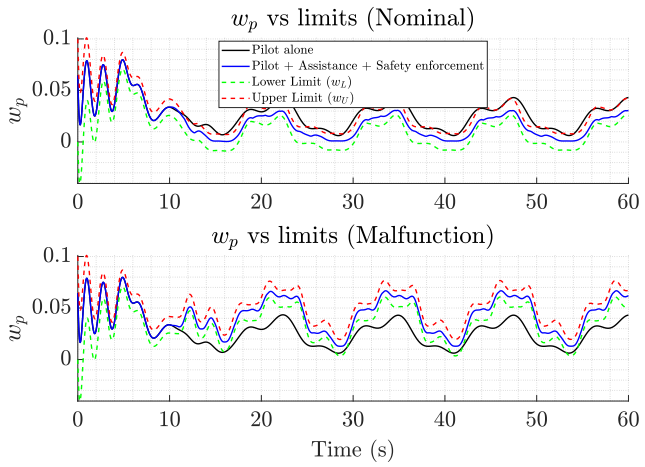

Fig. 5. Workload, $w_p(t)$, with upper and lower limits in (22).

envelope that adapts to task phase and assistance-induced trends, and guards against sudden under- or over-loading.

IV. Conclusion

This letter presented an Arbitration Control Barrier Function (ACBF) framework for safe shared human–automation control, enabling workload-aware assistance with guaranteed feasibility under concurrent human and automation actions. In this formulation, it is assumed that task-level reference $r(t)$ is known. Although this is correct when the task is predefined, such as refueling or landing, there may be cases where this assumption can be violated. Such cases may require intention estimation or incorporating robust CBFs. Extending the framework in this direction is considered for future work.

References

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

[2] C. E. Billings, "Human-centered aircraft automation: A concept and guidelines," AMES Research Center, Nat. Aeronaut. Space Admin., Washington, DC, USA, Rep. 103885, 1991.

[3] W.-P. Brinkman, M. A. Neerincx, and H. van Oostendorp, "Cognitive ergonomics for situated human–automation collaboration," *Interact. Comput.*, vol. 23, no. 4, pp. 3–4, Jul. 2011. [Online]. Available: https://doi.org/10.1016/S0953-5438(11)00060-9

[4] D. A. Abbink et al., "A topology of shared control systems—Finding common ground in diversity," *IEEE Trans. Human-Mach. Syst.*, vol. 48, no. 5, pp. 509–525, Oct. 2018.

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

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

[7] 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 Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1440–1453, Jun. 2017.

[8] W. Wang et al., "Decision-making in driver-automation shared control: A review and perspectives," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 5, pp. 1289–1307, Sep. 2020.

[9] 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 Trans. Intell. Veh.*, vol. 10, no. 7, pp. 4103–4115, Jul. 2025.

[10] 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," *Transp. Res. F Traffic Psychol. Behav.*, vol. 61, pp. 107–119, Feb. 2019.

[11] 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 Trans. Mechatronics*, vol. 23, no. 1, pp. 5–16, Feb. 2018.

[12] 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 Trans. Control Syst. Technol.*, vol. 27, no. 5, pp. 1962–1978, Sep. 2019.

[13] J. Jiang and A. Astolfi, "Shared-control for a rear-wheel drive car: Dynamic environments and disturbance rejection," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 5, pp. 723–734, Oct. 2017.

[14] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. IEEE 18th Eur. Contr. Conf. (ECC)*, 2019, pp. 3420–3431.

[15] C. Hu and J. Wang, "Trust-based and individualizable adaptive cruise control using control barrier function approach with prescribed performance," *IEEE Trans. Int. Trans. Syst.*, vol. 23, no. 7, pp. 6974–6984, Jul. 2022.

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

[17] J. Dallas et al., "Control barrier functions for shared control and vehicle safety," 2025, *arXiv:2503.19994*.

[18] 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*, vol. 25, no. 2, p. 405, 2025.

[19] 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 Robot. Autom. Lett.*, vol. 9, no. 11, pp. 9581–9588, Nov. 2024.

[20] 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 Trans. Control Syst. Technol.*, vol. 33, no. 4, pp. 1151–1163, Jul. 2025.

[21] M. Marcano, S. Díaz, J. Pérez, and E. Irigoyen, "A review of shared control for automated vehicles: Theory and applications," *IEEE Trans. Human-Mach. Syst.*, vol. 50, no. 6, pp. 475–491, Dec. 2020.

[22] W. Xiao, C. G. Cassandras, and C. Belta, *Safe Autonomy With Control Barrier Functions: Theory and Applications*. Cham, Switzerland: Springer, 2023.

[23] J. C. de Winter and D. Dodou, "Preparing drivers for dangerous situations: A critical reflection on continuous shared control," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2011, pp. 1050–1056.

[24] B. J. Bacon and D. K. Schmidt, "An optimal control approach to pilot/vehicle analysis and the Neal-smith criteria," *J. Guid. Control Dyn.*, vol. 6, no. 5, pp. 339–347, 1983.

[25] A. E. Bryson, *Control of Spacecraft and Aircraft*. Princeton, NJ, USA: Princeton Univ. Press, 1994.

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on March 03,2026 at 16:11:32 UTC from IEEE Xplore. Restrictions apply.