Safe Adaptive Output Tracking Controller for Robotic Manipulators

Ritirupa Dey[1], Avimanyu Sahoo[2], *Senior Member, IEEE*, and Vignesh Narayanan[1], *Senior Member, IEEE*

***Abstract*—Accurate tracking control of robotic manipulators often requires reliable velocity measurements and precise knowledge of system dynamics. However, in practice, joint velocity sensors are either unavailable or prone to noise, and exact robot dynamics may not always be known. To address these challenges, especially when the robot trajectories are required to adhere to prescribed safety constraints, in this paper, we propose a neural network (NN)-based adaptive observer and controller framework that relies solely on joint position feedback for an n-link robotic manipulator. To this end, we develop an adaptive observer with a custom online weight-tuning law to estimate the joint velocities of the robotic system. Using the estimated states, we design a tracking controller that learns the robot dynamics online and captures the model nonlinearities. Using the proposed closed-loop observer–controller system, we derive sufficient conditions based on control-barrier function (CBF) that guarantee system trajectories respect prescribed safety constraints, thereby ensuring safe robotic operation. We include numerical simulations on a two-link robotic manipulator system to validate the effectiveness of the proposed controller.**

I. Introduction

The importance of safety in robotic manipulators has grown significantly with their deployment in human-centric environments such as collaborative manufacturing, medical assistance, and service robotics [1]. Various different controllers schemes, including adaptive NN-based controllers, have been proposed in the literature for both regulation and tracking tasks [2], [3]. Beyond achieving precise trajectory tracking, manipulators must operate within prescribed safe envelopes of joint position and velocity limits to prevent accidents in such human-robot interaction scenarios. To address these challenges, safety-critical control frameworks have been developed in the literature, among which control barrier functions (CBFs) have emerged as an effective tool for guaranteeing forward invariance of safe sets in safety-critical systems [4]–[6].

CBFs define safety through inequality constraints on the system state [7], [8]. Standard CBF formulations, however, rely on accurate system dynamics. To address system uncertainties and safety, several extensions have been proposed, including robust, learning-based, and adaptive CBF methods [9]–[11]. Robust CBFs handle uncertainty via worst-case bounds and are often conservative [9]. Learning-based approaches relax modeling requirements by approximating CBF dynamics from data, however typically require large offline datasets and face scalability issues [10], [12]. Adaptive CBF methods enforce safety online [13], but commonly assume linear-in-the-parameters (LIP) uncertainty structures and are mainly demonstrated on regulation tasks. Moreover, most existing approaches require full state measurements, which limits their applicability in practical robotic systems.

In robotic manipulators, accurate velocity information is essential for feedback control. When direct velocity measurements are unavailable, finite-difference estimators based on position data are commonly used, but they amplify sensor noise and are unsuitable for safety-critical applications [14]. This has motivated observer-based approaches for enforcing safety under partial or noisy state measurements. In such frameworks, observers reconstruct unmeasured states and the CBF constraints are modified to account for estimation errors, ensuring system safety despite imperfect information. For instance, [15] combines an input-to-state stable (ISS) observer with CBFs to guarantee forward invariance under known bounded estimation error assumptions.

Several observer-based CBF frameworks have been proposed to address safety under state estimation uncertainty. In [16], function approximation is used to model observer-induced uncertainties, combined with an adaptive CBF for control design. An extended state observer (ESO) is employed in [17] to estimate both states and disturbances, which are incorporated into a robust CBF-based quadratic program. Disturbance-observer-based approaches in [18], [19] further reduce conservatism by parameterizing CBF constraints using observer estimates. More recently, [11] integrates an adaptive deep neural network with CBFs and a state-derivative estimator to handle system uncertainties; however, the resulting CBF depends on known bounds of the drift dynamics and is limited to regulation tasks. While these methods demonstrate the feasibility of observer–CBF designs, they often rely on specific uncertainty structures or LIP assumptions, leaving open challenges for the task in robotic manipulators.

To address this gap, we introduced a safe human-in-the-loop neural network controller trained offline for robotic manipulators [20] and an adaptive state-constrained online neural network controller [21] with safety guarantees established through CBF–based torques, while assuming full state measurements. In this paper, we build on them and propose a combined observer-controller architecture for controlling robot manipulators to follow a desired trajectory while adaptively satisfying safety requirements represented as state-constraints. The primary contributions of this paper are threefold : (i) Design of a lightweight functional link neural network (FLNN)-based observer with a novel weight tuning law, to estimate the unknown system velocities. Unlike recent approaches [15]–[17], the FLNN formulation avoids stringent assumptions, like LIP or prior bounds on drift terms. (ii) Development

This work was supported by Air Force Office of Scientific Research under grant FA9550-24-1-0228, and National Science Foundation under grants 2337998 and 2337999.

[1]Ritirupa Dey and Vignesh Narayanan are with the AI Institute, University of South Carolina, Columbia, SC, USA (deyr@email.sc.edu and vignar@sc.edu). [2]Avimanyu Sahoo is with the Department of Electrical Engineering, University of Alabama in Huntsville (UAH), AL, USA (avimanyu.sahoo@uah.edu).

of an online NN-based controller that captures the complex nonlinearities of the robot dynamics and ensures trajectory tracking while simultaneously enforcing both joint position and velocity constraints through a unified CBF. (iii) Analytical safety guarantees via safe-set invariance with adaptive torque control of the novel combined observer-controller framework.

Notations: We use standard mathematical notations. We use the Euclidean norm for vectors and for a matrix $A = [a_{ij}]$, $\|A\|_F$ denotes its Frobenius norm. Throughout, 'hat', "$\hat{(\cdot)}$", denotes estimated quantities, 'tilde', "$\tilde{(\cdot)}$", denotes estimation errors, and the subscripts "o" and "c" refer to observer- and controller-related variables, respectively.

II. Problem Formulation

Consider an n-link robot manipulator with dynamics expressed in the Euler-Lagrange form [3] as

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

with $q(t), \dot{q}(t) \in \mathbb{R}^n$ representing the joint position and velocity vectors, $M(q(t))$, $V_m(q(t),\dot{q}(t))$, $G(q(t))$, and $F(\dot{q}(t))$ representing the inertia matrix, the coriolis/centripetal force, the effect of gravity, and the friction force acting on the robotic arm, respectively. Bounded unknown disturbances, e.g., unstructured/unmodeled dynamics, are denoted by $\tau_d(t)$ and $\tau(t)$ is the control torque. Note that only $q(t)$ can be measured in this scenario.

It is well known that (i) $M(q(t))$ is a positive definite symmetric matrix bounded by $m_1 I \leq M(q(t)) \leq m_2 I$, where m_1, m_2 are known positive constants, (ii) $\|V_m(q(t),\dot{q}(t))\| \leq v_b(q(t))\|\dot{q}(t)\|$ with $v_b(q(t)) \in \mathcal{C}^1(S)$, where S is a compact simply connected set of $\mathbb{R}^n$ and $\mathcal{C}^1$ is the set of all continuously differentiable functions from $\mathbb{R}^n$ to $\mathbb{R}$, and (iii) the matrix $\dot{M}(q(t)) - 2V_m(q(t),\dot{q}(t))$ is skew-symmetric [3].

Assumption 1. *The desired trajectory $q_d(t)$ and its derivatives are bounded by a known real constant Q_d, i.e.,*

$$\|[q_d^T(t) \;\; \dot{q}_d^T(t) \;\; \ddot{q}_d^T(t)]^T\| \leq Q_d. \quad (2)$$

Furthermore, the desired trajectories should be a feasible trajectory for the given robotic system, i.e., $(q_d(t), \dot{q}_d(t))$ should be a solution to the dynamics given in (1).

Assumption 2. *The disturbance term τ_d is bounded by a known real constant, i.e., $\|\tau_d\| \leq b_d$.*

These standard assumptions are essential and guarantee existence of a tracking control torque for the robotic system.

Assumption 3. *The inertia matrix $M(q(t))$ is known.*

Control Objective: Let $q_d(t) \in \mathbb{R}^n$ be a desired joint trajectory. The control objective is to design $\tau(t)$ such that $q(t)$ satisfies $\lim_{t\to\infty} \|q(t)-q_d(t)\| = 0$, while simultaneously respecting the safety constraints $\|q_s(t) - q(t)\| \leq \epsilon_M$ and $\|\dot{q}_s(t)-\dot{q}(t)\| \leq \epsilon_M$ for all $t \geq 0$, where $q_s(t), \dot{q}_s(t) \in \mathbb{R}^n$ are the prescribed *safe* joint position and velocity trajectories, and $\epsilon_M > 0$ is a tolerance margin. The variable $q_s(t)$ represents the prescribed safe joint trajectory around which a tolerance band of width $2\epsilon_M$ is defined. Note that the threshold ϵ_M may differ between the joint position and velocity constraints. However, for simplicity of exposition, a common threshold is adopted throughout this work.

Remark 1. *The defined safety constraints spans two representative cases as detailed in [21]: (Case i) When $q_s(t) = q_d(t)$. (Case ii) When $q_s(t)$ is constant.*

Next, we present our NN-based solution to the control design problem that meets all the objectives.

III. Main Results

The proposed observer–controller framework is illustrated in Fig. 1. It uses two neural networks, both adaptively updated, one as the observer and the other for designing the feedback controller. The detailed designs of the NN-based observer and controller are presented in the following subsections.

A. Design of FLNN-based Observer

Denoting state variables $x_1(t) = q(t)$, $x_2(t) = \dot{q}(t)$, the robot dynamics (1) can be written as

$$\begin{aligned}&\dot{x}_1(t) = x_2(t),\\ &\dot{x}_2(t) = h_o(x_1(t), x_2(t)) - M^{-1}(x_1(t))\tau_d(t)+\\ &M^{-1}(x_1(t))\tau(t),\end{aligned} \quad (3)$$

where y(t) = $x_1(t)$ is the measured output and $h_o(x_1(t), x_2(t))$ is given by

$$h_o(x_1(t), x_2(t)) = -M^{-1}(x_1(t))\{V_m(x_1(t), x_2(t))x_2(t) + F(x_2(t))+G(x_1(t))\}. \quad (4)$$

To design an observer, under the standard smoothness and boundedness assumptions commonly applied to the robot dynamics, the unknown nonlinear function $h_o(x_1(t), x_2(t))$ can be approximated with an FLNN. Alternatively, there exists some constant ideal weight matrix $W_o \in \mathbb{R}^{N_o \times n}$ and sufficient number N_o of input basis function [3] such that

$$h_o(x_1(t), x_2(t)) = W_o^T(t)\sigma_o(x_1(t), x_2(t)) + \epsilon_o(x_1(t), x_2(t)), \quad (5)$$

where $\|\epsilon_o(x_1(t), x_2(t))\| \leq \epsilon_{o,M}$ is the approximation error and $\sigma_o(x_1(t), x_2(t)) \in \mathbb{R}^{N_o}$ is the basis or activation function.

Assumption 4. *The ideal weight matrix of the FLNN, W_o, is bounded by known positive constants, i.e., $\|W\|_o \leq Z_{W_o}$.*

Defining the the estimated states $\hat{x}_1(t), \hat{x}_2(t)$, the functional estimate of (5) is given by

$$\hat{h}_o(\hat{x}_1(t), \hat{x}_2(t)) = \hat{W}_o^T(t)\sigma_o(\hat{x}_1(t), \hat{x}_2(t)), \quad (6)$$

where $\hat{W}_o$ is the estimated NN weight matrix tuned online (described later).

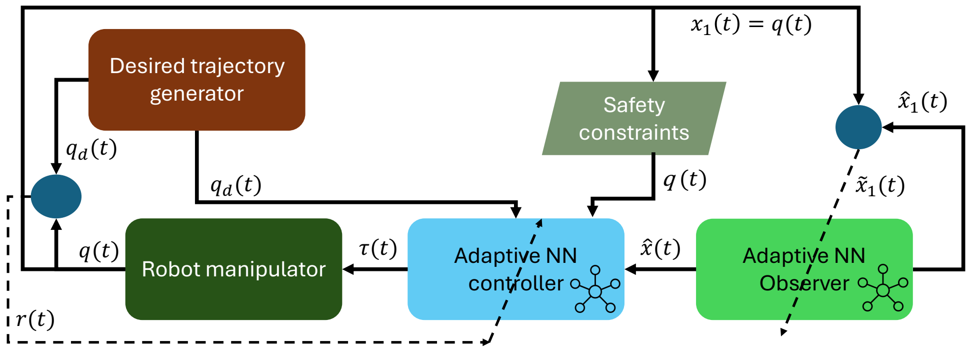

Fig. 1. Block Diagram of the proposed observer-controller framework.

Using the estimated function (6), the proposed FLNN-based observer has the form

$$\begin{aligned}\dot{\hat{z}}_1(t) =& \hat{x}_2(t) + k_D\tilde{x}_1(t)\\ \dot{\hat{z}}_2(t) =& \hat{W}_o\sigma_o(\hat{x}_1(t), \hat{x}_2(t)) - M^{-1}\tau_d(t)+\\ & k_0 I\tilde{x}_1(t) + M^{-1}\tau(t),\end{aligned} \tag{7}$$

where $\tilde{x}_1 = x_1 - \hat{x}_1$ is the output estimation error, $k_p > 0$ and $k_D > 0$ are design parameters (observer gains), and

$$k_0 = \left(\frac{kpk_D\|\tilde{x}_1\| + \sqrt{N_o}(Z_{W_o} + \|\hat{W}_o\|) + (\epsilon_{o,M} + \zeta_{o,M}) + \|e\|}{2\sqrt{k_pk_D(\|\tilde{x}_1\|^2+\nu)}}\right)^2, \tag{8}$$

$\zeta_{o,M} > 0$ is a disturbance bound. Using the observed states from (8), the system state estimates are can be computed as

$$\begin{aligned}\hat{x}_1(t) &= \hat{z}_1(t),\\ \hat{x}_2(t) &= \hat{z}_2(t) + k_p\tilde{x}_1(t),\end{aligned} \tag{9}$$

Using (7), the state estimation dynamics can be rewritten as

$$\begin{aligned}&\dot{\hat{x}}_1(t) = \hat{x}_2(t) + k_D\tilde{x}_1(t),\\ &\dot{\hat{x}}_2(t) = \dot{\hat{z}}_2(t) + k_p\dot{\tilde{x}}_1(t) = \hat{W}_o\sigma_o(\hat{x}_1(t), \hat{x}_2(t)) - M^{-1}\tau_d(t)\\ &+ k_0I\tilde{x}_1(t) + M^{-1}\tau(t) + k_p\dot{\tilde{x}}_1(t).\end{aligned} \tag{10}$$

Defining the observer errors $\tilde{x}_1 = x_1 - \hat{x}_1$ and $\tilde{x}_2 = x_2 - \hat{x}_2$, by subtracting (10) from (3), the observer error dynamics can be rewritten as

$$\begin{aligned}\dot{\tilde{x}}_1(t) =& \tilde{x}_2(t) - k_D\tilde{x}_1(t),\\ \dot{\tilde{x}}_2(t) =& W_o^T\sigma_o(x_1(t), x_2(t)) - \hat{W}_o^T\sigma_o(\hat{x}_1(t), \hat{x}_2(t))-\\ & k_0I\tilde{x}_1(t) - k_p\dot{\tilde{x}}_1(t) + \epsilon_o(x(t)).\end{aligned} \tag{11}$$

Adding and subtracting $W_o^T\sigma_o(\hat{x})$ yields $\dot{\tilde{x}}_1(t) = \tilde{x}_2(t) - k_D\tilde{x}_1(t)$, $\dot{\tilde{x}}_2(t) = \tilde{W}_o^T\sigma_o(\hat{x}_1(t), \hat{x}_2(t)) - k_0I\tilde{x}_1(t) - k_p\dot{\tilde{x}}_1(t) + w_o(x(t), \hat{x}(t)) + \epsilon_o(x(t))$, where $\tilde{W} = W - \hat{W}$ is the FLNN weight estimation error and $w_o(x, \hat{x}) = W_o^T\{\sigma_o(x_1, x_2) - \sigma_o(\hat{x}_1, \hat{x}_2)\}$. With simple algebraic manipulation, the observer dynamics can be expressed as

$$\begin{aligned}\dot{\tilde{x}}_1 &= \tilde{x}_2 - k_D\tilde{x}_1,\\ \dot{\tilde{x}}_2 &= -k_p\tilde{x}_2 - (k_0 - k_pk_D)\tilde{x}_1 + \tilde{W}_o^T\sigma_o(\hat{x}_1, \hat{x}_2)+\\ &\quad \epsilon_o(x) + w_o(x, \hat{x}).\end{aligned} \tag{12}$$

Remark 2. *The disturbance function* $w_o(x, \hat{x})$ *is bounded by* $\|w_o(x, \hat{x})\| \le \zeta_{o,M}$, *with* $\zeta_{o,M} > 0$.

The design of the safe set is presented below.

B. Design of Safe Sets

Given the desired trajectory $q_d(t)$, the tracking error and the filtered tracking error [3] are respectively defined as $e(t) = q_d(t) - x_1(t)$ and $\hat{r}(t) = \dot{\hat{e}}(t) + \Lambda e(t)$, where $\Lambda = \alpha I, \alpha > 0$ is a positive definite design matrix.

First, we design the safe set in terms of the joint position and velocity constraints as

$$\begin{aligned}S_{q,\dot{q}} = \{(x_1(t), \hat{x}_2(t)) : \|q_s(t) - x_1(t)\| \le \alpha_{max},\\ \|\dot{q}_s(t) - \hat{x}_2(t)\| \le \alpha_{max}\}.\end{aligned} \tag{13}$$

Based on this, we define two auxiliary sets in the error space and the filtered-error space, as

$$\begin{aligned}S_{e,\dot{\hat{e}}} = \{(e(t), \dot{\hat{e}}(t)) : \|e(t)\| \le |\alpha_{max} - \|q_s(t) - q_d(t)\||_{L_\infty}\\ = \mathcal{E}_1, \|\dot{\hat{e}}(t)\| \le |\alpha_{max} - \|\dot{q}_s(t) - \dot{q}_d(t)\||_{L_\infty} = \mathcal{E}_2\},\end{aligned} \tag{14}$$

$$S_{\hat{r}} = \{\hat{r}(t) = \dot{\hat{e}}(t) + \Lambda e(t) : \|\hat{r}(t)\| \le \mathcal{M}_R - \mathcal{M}_r\}, \tag{15}$$

where $\mathcal{R} = [\mathcal{M}_R - \mathcal{M}_r]$, with $\mathcal{M}_R = min\{\sigma_{max}(\Lambda)\mathcal{E}_1, \mathcal{E}_2\}$ and $\mathcal{M}_r = min\{\|\Lambda e(t)\|, \|\dot{\hat{e}}(t)\|\}$. Next, we derive the relationships between these safe sets in the following theorem.

Theorem 1. *(i) Consider the sets* $S_{q,\dot{q}}$, $S_{e,\dot{\hat{e}}}$, *and* $S_{\hat{r}}$ *defined by (13)-(15). Let* δ_v, δ_q, *and* ϵ_M *be positive constants such that* $\|\dot{q}_s(t) - \dot{q}_d(t)\| < \delta_v$ *and* $\|q_s(t) - q_d(t)\| < \delta_q$ *for all* $t \in \mathbb{R}_{\ge t_0}$. *Choose* $\delta_v < \alpha_{max}$ *and* $\delta_q < \alpha_{max}$, *then if* $\hat{r}(t) \in S_{\hat{r}}$, *then it follows that* $(e(t), \dot{\hat{e}}(t)) \in S_{e,\dot{\hat{e}}}$ *and* $(x_1(t), \hat{x}_2(t)) \in S_{q,\dot{q}}$ *for all* $t \in \mathbb{R}_{\ge t_0}$. *(ii) If a point* $(x_1(t), \hat{x}_2(t)) \in \partial S_{q,\dot{q}}$, *then the corresponding* $\hat{r}(t) \in \partial S_{\hat{r}}$, *where* $\partial(\cdot)$ *denotes the boundary of a set.*

Proof. Refer to [21].

Remark 3. *Theorem 1 shows that safety with respect to* $S_{\hat{r}}$ *implies safety with respect to* $S_{e,\dot{\hat{e}}}$ *and, consequently,* $S_{q,\dot{q}}$ *[21]. Through the boundedness of both the estimated velocity states and the weight errors, it follows that the actual velocities are also bounded.*

We enforce safety constraints on the joint position and velocity estimation errors. These constraints are given by

$$S_{\tilde{x}_1} = \{\tilde{x}_1 : \|\tilde{x}_1\| \leq \epsilon_{max}\}, S_{\tilde{x}_2} = \{\tilde{x}_2 : \|\tilde{x}_2\| \leq \epsilon_{max}\}. \quad (16)$$

Safety restrictions are also imposed on the FLNN-based observer and NN-based controller weight errors so that they remain confined within predefined bounds given by

$$S_{\tilde{W}_o} = \{\tilde{W}_o : \|\tilde{W}_o\| \leq Z_{\tilde{W}_o}\}, \quad (17)$$

$$S_{\tilde{W}_c,\tilde{V}_c} = \{(\tilde{W}_c,\tilde{V}_c) : \|\tilde{W}_c\| \leq Z_{\tilde{W}_c}, \|\tilde{V}_c\| \leq Z_{\tilde{V}_c}\}. \quad (18)$$

C. Safe NN-based Output Feedback Controller Design

Differentiating $\hat{r}(t)$ and using (1), the filtered tracking error dynamics are obtained as

$$M\dot{\hat{r}}(t) = -V_m\hat{r}(t) - \tau(t) + f(\xi(t)) + \tau_d(t), \quad (19)$$

where the unknown nonlinearities

$$\begin{aligned} f(\xi(t)) =& M(x_1(t))(\ddot{q}_d(t) + \Lambda\dot{\hat{e}}(t)) + V_m(x_1(t), \hat{x}_2(t))(\dot{q}_d(t) \\ &+ \Lambda e(t)) + G(x_1(t)) + F(\hat{x}_2(t)), \end{aligned} \quad (20)$$

with $\xi(t) = [e^T(t)\ \dot{\hat{e}}^T(t)\ q_d^T(t)\ \dot{q}_d^T(t)\ \ddot{q}_d^T(t)]^T$. Under the standard smoothness and boundedness assumptions commonly applied to the robot dynamics, the unknown nonlinear function $f(\xi(t))$ can be approximated with a DNN [3] as

$$\hat{f}(\xi(t)) = \hat{W}_c^T\sigma_c(\hat{V}_c^T\xi(t)), \quad (21)$$

with $\hat{V}_c(t)$ and $\hat{W}_c(t)$ denoting the estimated values of the ideal NN weights V_c and W_c respectively. Then $f(\xi(t)) = \hat{f}(\xi(t)) - \epsilon_c(\xi)$ with $\epsilon_c(\xi)$ denoting the reconstruction error.

Assumption 5. *The ideal weight matrices of the NN-based controller, V_c and W_c, are bounded by known positive constants, i.e., $\|V_c\| \leq Z_{V_c}$ and $\|W_c\| \leq Z_{W_c}$. Define $Z_c = \begin{bmatrix} W_c & 0 \\ 0 & V_c \end{bmatrix}$. Then, $\|Z_c\| \leq Z_M$.*

The control torque is then defined as

$$\tau(t) = \hat{W}_c^T\sigma_c(\hat{V}_c^T\xi(t)) + K_v\hat{r}(t) - v_c(t), \quad (22)$$

with gain matrix $K_v = K_v^T > 0$ and $v_c(t)$ is a user-defined function that provides robustness against uncertainties [3]. The update rules to adapt the weights $\hat{W}_c, \hat{V}_c$ online and the closed-loop analysis of the controlled system have been reported in the literature [3].

Next we design the control torques and, using Theorem 1 and CBF, formally show that the system satisfies the safety constraints. The following lemma is needed for the same.

Lemma 1. *The error dynamics in* (19) *with the control* (22) *can be written as*

$$\begin{aligned} M\dot{\hat{r}}(t) =& -(K_v + V_m)\hat{r}(t) + \tilde{W}_c^T(t)(\hat{\sigma}_c - \hat{\sigma}_c'\hat{V}_c^T(t)\xi(t)) \\ &+ \hat{W}_c^T(t)\hat{\sigma}_c'\tilde{V}_c^T(t)\xi(t) + w_c(t) + v_c(t), \end{aligned} \quad (23)$$

where $w_c(t) = \tilde{W}_c(t)\hat{\sigma}_c'V_c^T(t)\xi(t) + W_c^T(t)O(\tilde{V}_c^T(t)\xi(t))^2 + (\epsilon_c + \tau_d(t))$, with $\hat{\sigma}_c = \sigma_c(\hat{V}_c^T\xi(t))$, $\hat{\sigma}_c'$ is its derivative and $O(.)$ denotes higher order terms of $\tilde{V}_c^T(t)\xi(t)$. Additionally, $w_c(t)$ satisfies$\|w_c(t)\| \leq C_0 + C_1\|\tilde{Z}_c(t)\|_F + C_2\|\tilde{Z}_c(t)\|_F\|\hat{r}(t)\|$, where C_0, C_1, and C_2 are known positive constants.

Proof. Refer to [3].
The main results are presented in the following theorem.

Theorem 2. *Consider the robotic manipulator system in* (1). *Let $q_d(t)$ be the desired trajectory, defined in Assumption 1. Suppose the FLNN-based observer given by* (12) *is used for estimating the unknown joint velocities of the manipulator. Let the weight tuning law for the FLNN be given by*

$$\dot{\hat{W}}_o = F_o(\kappa_o A - \hat{W}_o), \quad (24)$$

where $F_o = F_o^T > 0$ is a design matrix, A is a $N_o \times n$ matrix with entries 1, *and $\kappa_0 > Z_{W_o}$ is a design parameter. Let the weight matrices W_c, V_c for the NN-based controller be updated using*

$$\dot{\hat{W}}_c = F_c\hat{\sigma}_c\hat{r}^T - F_c\hat{\sigma}_c'\hat{V}_c^T\xi\hat{r}^T - \kappa_c F_c\|\hat{r}\|\hat{W}_c, \quad (25)$$

$$\dot{\hat{V}}_c = G_c\xi(\hat{\sigma}_c'^T\hat{W}_c\hat{r})^T - \kappa_c G_c\|\hat{r}\|\hat{V}_c, \quad (26)$$

where $F_c = F_c^T > 0, G_c = G_c^T > 0$ are design parameter matrices and $\kappa_c > 0$ is a scalar design parameter. Let the control torque be given by (22) *with*

$$\begin{aligned} v_c(t) =& -K_Z(\|\hat{Z}_c\|_F + Z_{W_c})\hat{r} - \\ & \frac{[\frac{\kappa(Z_{W_c} + \frac{C_1}{\kappa})^2}{4} + C_0]}{\sqrt{\delta + \|\hat{r}\|^2}} - e - \frac{\psi_B^2}{\delta + \|\hat{r}\|^2}\hat{r}, \end{aligned} \quad (27)$$

where $K_z > C_2$ with C_0, C_1, C_2 defined as in Lemma 1, Z_M as in Assumption 5 and B, ψ_B is defined as in Remark 4. Let the Assumptions 1-5 hold and $S_{\hat{r}}$ be defined as in (15), *$S_{\tilde{x}_1}$, $S_{\tilde{x}_2}$ and $S_{\tilde{W}_o}$ as defined in* (16)-(17). *Define α to be an extended K_∞-class function. Then there exists a ZCBF $H(\tilde{x}_1(t), \tilde{x}_2(t), \hat{r}(t), \hat{W}_o(t), \tilde{W}_c(t), \tilde{V}_c(t))$ with respect to safe set $S_{\tilde{x}_1} \times S_{\tilde{x}_2} \times S_{\hat{r}} \times S_{\tilde{W}_o} \times S_{\tilde{W}_c,\tilde{V}_c}$ and satisfying $\dot{H}(\cdot) \geq -\alpha(H(\cdot))$, which ensures the online safety of the system. Correspondingly, the set $S_{q,\dot{\hat{q}}}$ is forward-invariant, meaning that if $(x_1(t_0), \hat{x}_2(t_0)) \in S_{q,\dot{\hat{q}}}$, then $(x_1(t), \hat{x}_2(t)) \in S_{q,\dot{\hat{q}}}$ for all $t \in \mathbb{R}_{\geq t_0}$. Furthermore, $(x_1(t), x_2(t))$ also remain in their bounded safe set $S_{q,\dot{q}} = \{x_1(t), x_2(t)) : \|q_s(t) - x_1(t)\| \leq \alpha_{max}, \|\dot{q}_s(t) - x_2(t)\| \leq \epsilon_{max} + \alpha_{max}\}$.*

Proof. Refer to Appendix.

Remark 4. *The FLNN-based observer in* (12) *employs a novel weight tuning law* (24) *that differs from classical NN-based observer adaptations [22] and is specifically designed to guarantee CBF-based safety of the closed-loop robot–observer–controller system. From* (24), *we can obtain the dynamics of the FLNN weight estimation error as follows $\dot{\tilde{W}}_o = -F_o(\tilde{W}_o - B)$, where B is a bias matrix, with $\|B\| < 4\psi_B$, where ψ_B is a positive design parameter such that $\psi_B > Z_B + \kappa_0\sqrt{N_0 n}$.*

Remark 5. *The safety constraint on the actual joint velocities depends on the state estimation error, which is governed by the observer reconstruction error and can be made arbitrarily small through appropriate selection of the FLNN size and design parameters, as established in Theorem 2.*

IV. Simulations

The proposed method is validated on a planar two-link rigid robotic manipulator governed by (1). The system parameters are chosen as $l_1 = l_2 = 1$ m, $m_1 = 0.8$ kg, and $m_2 = 2.3$ kg. We choose $q_d(t) = [20\sin(0.5t), 10\sin(t)]^T$, with controller gains $K_v = diag\{20, 20\}$, $F = diag\{10, 10\}$, $G = diag\{12, 12\}$, $F_0 = I$, $\kappa_0 = 6$ and $\Lambda = diag\{5, 5\}$. The NN weights for both the observer and controller are initialized to zero, and the FLNN observer input is selected as $[\tilde{q}^T; \hat{q}^T; \dot{\hat{q}}^T; |\tilde{q}|; |\hat{q}|]^T$. The initial joint positions and velocities are set to $q(0) = [0.1, 0.1]^T$ and $\dot{q}(0) = [10, 10]^T$, respectively, with a constant disturbance $\tau_d = 0.01$.

We draw comparisons between a classical NN-based controller (baseline) without the safety constraints [3], which requires full state knowledge of the system, and our combined observer-controller framework proposed in Theorem 2. We choose case (b) as the representative example and define the safe set as $S_{q,\dot{q}} = \{(x_1(t), \hat{x}_2(t)) : \|x_1(t) - q_s(t)\| \leq 20, \|\hat{x}_2(t) - \dot{q}_s(t)\| \leq 13.5\}$, where $q_s(t)$ and $\dot{q}_s(t)$ are defined as $q_s(t) = \frac{\max_t(q_d(t)) + \min_t(q_d(t))}{2}$ and $\dot{q}_s(t) = \frac{\max_t(\dot{q}_d(t)) + \min_t(\dot{q}_d(t))}{2}$ and correspondingly, $S_{e,\dot{e}} = \{(e(t), \hat{\dot{e}}(t)) : \|e(t)\| \leq 1, \|\hat{\dot{e}}(t)\| \leq 5\}$. As shown in Fig. 2, although the classical NN controller (orange) tracks the desired position trajectory, the resulting velocities violate the safe set $S_{q,\dot{q}}$, leading to corresponding violations of the error constraints $S_{e,\dot{e}}$ in Fig. 3. In contrast, applying the control law in Theorem 2 (Fig. 4, bottom panel) ensures that $(x_1(t), \hat{x}_2(t)) \in S_{q,\dot{q}}$ and $(e(t), \hat{\dot{e}}(t)) \in S_{e,\dot{e}}$ for all time, thereby guaranteeing safety constraint satisfaction.

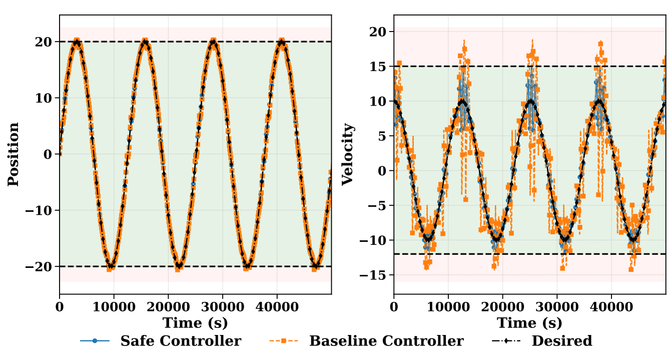

Fig. 2. Comparisons of the positions (left) and velocities (right) for joint 1 as generated using the two controllers. Green shade denotes safe and red shade denotes unsafe regions.

Further, Fig. 4 shows that the torque profiles of the classical NN controller, which does not account for safety, and the proposed observer–controller framework in Theorem 2 are comparable. Importantly, with a similar level of control effort, the proposed controller additionally ensures satisfaction of the safety constraints.

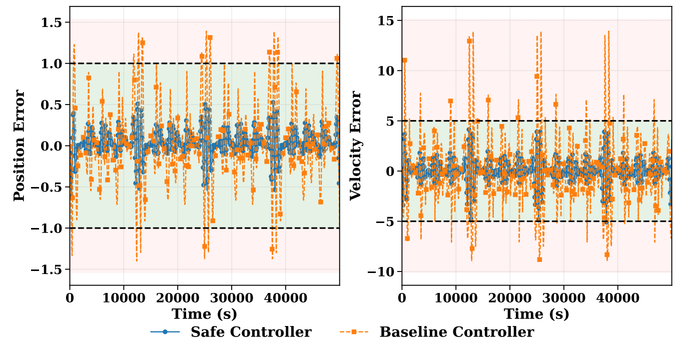

Fig. 3. Comparisons of the position (left) and velocity (right) errors for joint 1 as generated using the two controllers. Green shade denotes safe and red shade denotes unsafe regions.

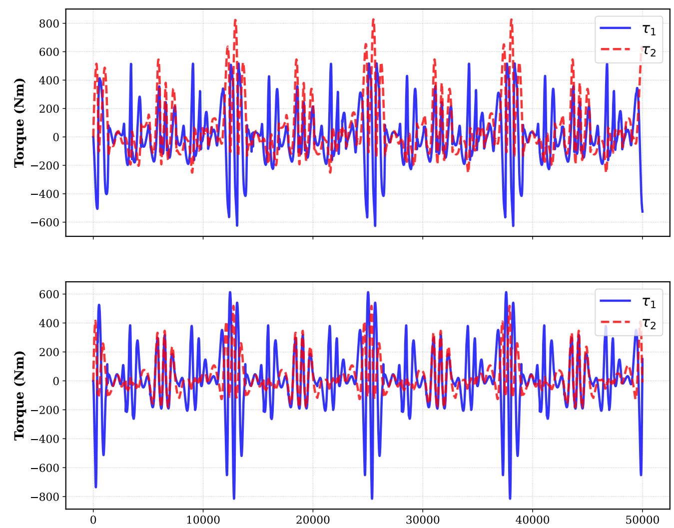

Fig. 4. Comparisons of the torques for the classical NN-based controller (top panel),and using Theorem 2 (bottom panel).

V. Conclusions

This paper presents a safety-aware, online NN-based observer–controller framework for robotic manipulators operating under joint position and velocity constraints. By defining a safe set in the filtered tracking error space, the tracking control problem is reformulated as a regulation task. The proposed online adaptive controller integrates CBF constraints into the NN adaptation law, ensuring that safety requirements are satisfied while maintaining accurate tracking performance. Numerical simulations validate the effectiveness of the framework, and future efforts will focus on hardware implementation and extension to more complex robotic systems.

Acknowledgement

Limited editorial assistance was provided by AI language models (ChatGPT, Gemini). All content was reviewed and approved by the authors.

APPENDIX

Proof Sketch of Theorem 2: (Note that, due to page constraints, only a proof sketch specifying the CBF function and the associated α-function is presented. This result holds under the properties and assumptions of the robotic manipulator system stated in the paper.)

Let

$$\begin{aligned} H_1 =& \frac{1}{2}\{(k_0\epsilon_{max}^2 - k_0\tilde{x}_1^T\tilde{x}_1) + (\epsilon_{max}^2 - \tilde{x}_2^T\tilde{x}_2) \\ &+ (\sigma_{max}(F_o^{-1})Z_{\tilde{W}_o}^2 - tr(\tilde{W}_o^T F_o^{-1}\tilde{W}_o))\}, \end{aligned} \quad (28)$$

$$\begin{aligned} H_2 =& \frac{1}{2}\{(\mathcal{E}_1^2 - e^T e) + (\sigma_{max}(M)\mathcal{R}^2 - \hat{r}^T M\hat{r}) \\ &+ (\sigma_{max}(F_c^{-1})Z_{\tilde{W}_c}^2 - tr(\tilde{W}_c^T F_c^{-1}\tilde{W}_c)) + \\ &(\sigma_{max}(G_c^{-1})Z_{\tilde{V}_c}^2 - tr(\tilde{V}_c^T G_c^{-1}\tilde{V}_c)) \end{aligned} \quad (29)$$

Taking the time derivative of H_1 and substituting the observer error dynamics in (12) together with the FLNN weight-tuning law $\dot{\tilde{W}}_o = -F_o(\tilde{W}_o - B)$, straightforward algebraic manipulations yield

$$\begin{aligned} \dot{H}_1 \geq& 2\sqrt{k_D k_p k_0}\|\tilde{x}_1\|\|\tilde{x}_2\| - k_p k_D\|\tilde{x}_2\|\|\tilde{x}_1\| - \\ &\sqrt{N_o}\|\tilde{x}_2\|\|\tilde{W}_o\|_F - \|\tilde{x}_2\|(\epsilon_{O,M} + \zeta_{O,M}) - \frac{\|B\|_F^2}{4}. \end{aligned} \quad (30)$$

Similarly, differentiating H_2 and using (23), (25)–(26), and the control law (22) with $v_c(t)$ in (27), we obtain

$$\dot{H}_2 \geq e^T\tilde{x}_2 + \psi_B^2 \quad (31)$$

Adding (30) and (31), and performing algebraic manipulations, we have $\dot{H} \geq 0$. Choosing $\alpha(H) = H$, which is a valid k_∞-class function, we have $\dot{H} \geq -\alpha(H)$. Therefore, H is a CBF on $S_{\tilde{x}_1} \times S_{\tilde{x}_2} \times S_{\hat{r}} \times S_{\tilde{W}_o} \times S_{\tilde{W}_c,\tilde{V}_c}$, and the individual sets are forward invariant. Theorem 1 guarantees that forward invariance of $S_{\hat{r}}$ implies forward invariance of $S_{q,\dot{\hat{q}}}$. We now show forward invariance of $S_{q,\dot{q}}$. Since $\|\hat{x}_2 - x_2\| \leq \epsilon_{max}$ and $\|\dot{q}_s(t) - \hat{x}_2\| \leq \alpha_{max}$, $\|\dot{q}_s(t) - x_2\| \leq \alpha_{max} + \epsilon_{max}, \forall t \geq t_0$. Therefore, $S_{q,\dot{q}}$ is forward invariant.

REFERENCES

[1] H. Mahdi, S. A. Akgun, S. Saleh, and K. Dautenhahn, "A survey on the design and evolution of social robots—past, present and future," *Robotics and Autonomous Systems*, vol. 156, p. 104193, 2022.

[2] V. Narayanan, S. Jagannathan, and K. Ramkumar, "Event-sampled output feedback control of robot manipulators using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 6, pp. 1651–1658, 2018.

[3] F. Lewis, S. Jagannathan, and A. Yesildirak, *Neural Network Control of Robot Manipulators and Non-linear Systems*. CRC press, 2020.

[4] 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)*. Ieee, 2019, pp. 3420–3431.

[5] M. A. Murtaza, S. Aguilera, M. Waqas, and S. Hutchinson, "Safety compliant control for robotic manipulator with task and input constraints," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 659–10 664, 2022.

[6] J. Li, Q. Liu, W. Jin, J. Qin, and S. Hirche, "Robust safe learning and control in an unknown environment: An uncertainty-separated control barrier function approach," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6539–6546, 2023.

[7] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[8] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 108–113, 2018.

[9] B. T. Lopez, J.-J. E. Slotine, and J. P. How, "Robust adaptive control barrier functions: An adaptive and data-driven approach to safety," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1031–1036, 2020.

[10] W. D. Compton, M. H. Cohen, and A. D. Ames, "Learning for layered safety-critical control with predictive control barrier functions," *arXiv preprint arXiv:2412.04658*, 2024.

[11] H. M. Sweatland, O. S. Patil, and W. E. Dixon, "Adaptive deep neural network-based control barrier functions," *arXiv preprint arXiv:2406.14430*, 2024.

[12] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 411–444, 2022.

[13] A. J. Taylor and A. D. Ames, "Adaptive safety with control barrier functions," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1399–1405.

[14] V. Chawda, O. Celik, and M. K. O'Malley, "Application of levant's differentiator for velocity estimation and increased z-width in haptic interfaces," in *2011 IEEE World Haptics Conference*. IEEE, 2011, pp. 403–408.

[15] D. R. Agrawal and D. Panagou, "Safe and robust observer-controller synthesis using control barrier functions," *IEEE Control Systems Letters*, vol. 7, pp. 127–132, 2022.

[16] Y. Wang and X. Xu, "Observer-based control barrier functions for safety critical systems," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 709–714.

[17] J. Chen, Z. Gao, and Q. Lin, "Robust control barrier functions for safe control under uncertainty using extended state observer and output measurement," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 8477–8482.

[18] A. Alan, T. G. Molnar, E. Daş, A. D. Ames, and G. Orosz, "Disturbance observers for robust safety-critical control with control barrier functions," *IEEE Control Systems Letters*, vol. 7, pp. 1123–1128, 2022.

[19] Z. Yang and L. Xie, "Disturbance observer-parameterized control barrier function with adaptive safety bounds," *IEEE Control Systems Letters*, 2025.

[20] R. Dey, A. Sahoo, and V. Narayanan, "Safe data-enabled control of human-in-the-loop robotic manipulator systems," *IEEE Transactions on Industrial Informatics*, 2025.

[21] ——, "State-constrained online adaptive control for robotic manipulators," *IEEE Control Systems Letters*, vol. 9, pp. 2855–2860, 2025.

[22] Y. H. Kim and F. L. Lewis, "Neural network output feedback control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 301–309, 2002.