Robust Stability of Neural-Network-Controlled Nonlinear Systems With Parametric Variability

Soumyabrata Talukder, *Student Member, IEEE*, and Ratnesh Kumar

Abstract—**Stability certification and identification of a safe and stabilizing initial set are two important concerns in ensuring operational safety, stability, and robustness of dynamical systems. With the advent of machine-learning tools, these issues need to be addressed for the systems with machine-learned components in the feedback loop. To develop a general theory for stability and stabilizability of neural network (NN)-controlled nonlinear systems subject to bounded parametric variations, a Lyapunov-based stability certificate is proposed and is further used to devise a maximal Lipschitz bound for a class of stabilizing NN controllers, and also a corresponding maximal Region of Attraction (RoA) within a user-specified safety set. To compute a robustly stabilizing NN controller that also maximizes the system's long-run utility, a stability-guaranteed training (SGT) algorithm is proposed. The effectiveness of the proposed framework is validated through an illustrative example.**

Index Terms—**Dynamic stability, imitation learning, Lipschitz bound, Lyapunov function, neural network (NN), Region of Attraction (RoA), reinforcement learning (RL), robust stability.**

I. Introduction

APPLICATION of neural networks (NNs) to control dynamical systems has been gaining attention following the recent architectural innovations in NN and the advancements in training algorithms. The NN controllers are trained either in a supervised way, often referred to as *imitation learning* [1], [2], or in a semisupervised way in the form of reinforcement learning (RL) [3]. RL methods allow data-driven learning of an optimal policy by interacting with the physical system and receiving a reward for each one-step action, without requiring explicit knowledge of the model, e.g., Q-learning [4], and various versions of policy-gradient methods [5], [6], [7], [8]. In contrast, "model-based" RL methods are feasible when a model of the physical system, to be used to train a controller, is either known or can be identified by interacting with the system [9], [10].

Using NNs as controllers offers design flexibility owing to its ability to approximate a large class of Lipschitz functions [11]. Yet their demonstrations are mostly restricted in simulated environments [12], [13], [14]. One key reason is the lack of closed-loop stability assurance of systems under NN control. Their stability analysis is challenging due to the inherent complexity of NN-based control policies [15]. These limitations form our motivation behind developing ways to formally guarantee the stability of NN-controlled systems and compute their Region of Attractions (RoAs [16], [17]).

A. Related Works

In [18] and [19], stability-assured RL algorithms are proposed, where the RL controllers are restricted to be linear and are learned through a gradient-based update. The input to such a controller is a set of manually crafted nonlinear bases of the system states, but the selection of a set of effective bases for a given system is still an unsolved problem [20]. Ma et al. [21] and Jiang et al. [22] designed a similar control scheme for nonlinear multiagent systems. Combining a radial basis function NN and a command filter, Cheng et al. [23] proposed an adaptive decentralized 2-bit-triggered control design for interconnected nonlinear systems in nonstrict-feedback forms with actuator failures. For the aforementioned cases and others [18], [19], [20], [21], [22], [24], [25], [26], the notion of stability is one of uniform *ultimate* boundedness of the state and/or output signals, whereas a method to ensure the safety of the entire state trajectory (so it remains contained within a given safe domain) has not been reported. An $L_2 - L_\infty$-quantized filter and a triggering matrix are codesigned in [27] for a linear plant under a single-layered NN controller and external disturbances, to ensure that denial-of-service-attack induced errors are exponentially stable. The above methods do not generalize for multilayered NN controllers with nonlinear activations due to the additional challenge of underlying nonconvexity in controller training.

A few recent works exist in [28], [29], and [30] which aim to address the problem of guaranteeing the stability of *multilayered* NN-controlled nonlinear systems. However, the majority of these works study a linearized system, with the effect of nonlinearity and/or parametric uncertainty modeled as integral quadratic constraints (QCs) [31]. Among these, the method suggested in [28] guarantees finite $\mathcal{L}_2$ gain with respect to an external disturbance and also computes a corresponding "Lipschitz-like" upper bound needed for the NN controller. However, the designed controller fails to guarantee stability even in absence of any disturbance. In [29], the nonlinearity of an already trained NN controller is locally sector-bounded to

Manuscript received 31 August 2022; revised 2 January 2023; accepted 9 March 2023. This work was supported in part by the National Science Foundation under Grant CSSI-2004766 and Grant PFI-2141084. This article was recommended by Associate Editor X. Wang. *(Corresponding author: Soumyabrata Talukder.)*

The authors are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50010 USA (e-mail: talukder@iastate.edu; rkumar@iastate.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2023.3257269.

Digital Object Identifier 10.1109/TSMC.2023.3257269

attain asymptotic stability of a discrete-time system, and also to estimate an RoA in the form of a sublevel set of a Lyapunov function. While the method can *verify* the stability under a given controller, it cannot be used to *synthesize* a stabilizing NN controller. In a later work [30], Yin et al. proposed an imitation learning-oriented stability-guaranteed training (SGT) algorithm for NN controller synthesis, providing a convex stability certificate for discrete-time system models. However, its application is restricted to systems free from actuator nonlinearity and/or uncertainty since their presence introduces nonconvexity. Moreover, the suggested NN training algorithm solves a semidefinite program (SDP) at each NN parameter update step, making the training computationally expensive.

Among other methods, an iterative counterexample-guided search for a Lyapunov function is introduced in [32] and [33] to provide stability under ReLu-based NN controllers. The algorithm in [32] is guaranteed to converge in finite iterations, but the application domain is limited to piecewise linear discrete-time systems and cannot handle parametric variation. Aydinoglu et al. [34] showed that the ReLu activation function can be represented as the solution of a linear complementarity problem, thereby casting the stability certification of a linear-complementarity system with a ReLu-based NN controller as a linear matrix inequality (LMI). Han et al. [35] and Zhang et al. [36] introduced an "actor–critic" RL algorithm, where the critic NN is structurally constrained to be positive definite as desired of a Lyapunov function. In [37], an augmented random search-based "soft safe" RL algorithm is proposed that employs a corresponding penalty term to the policy NN's objective. None of these methods [35], [36], [37] can yield a formal stability guarantee.

B. Contributions

For the class of locally continuously differentiable continuous-time (CT) nonlinear systems subject to parametric variations within a known bound, under NN-based state-feedback control, we make the following key contributions.

1) A Lyapunov-based sufficient condition is introduced to certify a system's local asymptotic stability, robust to arbitrary parametric variations, under an NN-based state-feedback controller satisfying a certain Lipschitz bound. Our stability condition is not limited to any special class of NN activation functions.
2) An algorithm is introduced using the above result to compute a maximal Lipschitz bound such that any controller satisfying the bound is robustly stabilizing in the presence of bounded parameter variations, and also a corresponding "robust safe initialization set" (RSIS) that is a maximal robust RoA contained within a user-given safe operating domain (so that any initialization of the controlled-system within the RSIS guarantees that the state trajectory never leaves the safe domain and eventually converges at the system's equilibrium (assumed to be independent of the parameter and so it remains unchanged with the parameter change).
3) An actor–critic RL algorithm is proposed to synthesize a multilayered NN controller satisfying the above Lipschitz bound and that also maximizes the system's expected utility with respect to random initializations and parametric variations.

Unlike the studies in [32], [33], and [34] that limit the activation to be ReLu, our stability condition is applicable to any NN activation functions. Furthermore, unlike [35], [36], [37], our analysis is able to offer a formal closed-loop stability guarantee. Furthermore, in contrast to [18], [19], [20], [21], [22], [24], [25], [26], and [28], our method guarantees that the system's trajectory never leaves a given safe domain. Also, in contrast to [29] that only provides a stability verification result, our work also introduces a method for controller synthesis. Moreover, unlike [30], our stability condition allows nonlinearity and parametric variation in the actuator, and our proposed SGT of NN controllers does not suffer from solving a computationally expensive SDP at each update of NN parameters.

C. Organization and Notations

In what follows, Section II presents the problem statement and an overview of our solution approach. Section III provides the mathematical preliminaries, followed by our main stability theorem, which is then used to develop an algorithm to identify a class of robustly stabilizing NN-based controllers that attain a maximal common RSIS. Section IV presents our RL algorithm to search for a stabilizing controller locally within the identified class, which maximizes a long-run expected utility. Section V validates the proposed method through an illustrative example, and Section VI concludes our work.

Notations: $\mathbb{R}$ (resp., $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$) denotes the real (resp., non-negative real, positive real) scalar field, $\mathbb{R}^n$ denotes the n-dimensional real vector field, and $\mathbb{R}^{m\times n}$ denotes the space of all real matrices with m rows and n columns. Operators $\leq, <, \geq$, and $>$ on matrices or vectors indicate elementwise operations. For $x \in \mathbb{R}^n$, x^i denotes its ith element, and $\|x\|_p$ denotes its p-norm for any real $p \geq 1$. If x is an n-length sequence of reals or $x \in \mathbb{R}^n$, $\text{diag}(x)$ denotes the $n \times n$ diagonal matrix, where the ith diagonal element is the ith element of x. For $M \in \mathbb{R}^{m\times n}$, its (i, j)th element is denoted by $M^{i,j}$ and $M^{\mathrm{T}} \in \mathbb{R}^{n\times m}$ denotes its transpose. For $M \in \mathbb{R}^{m\times n}$, $|M| \in \mathbb{R}^{m\times n}$ denotes the matrix comprising the elementwise absolute values, and if M is square and symmetric (i.e., $m = n$ and $M = M^{\mathrm{T}}$), $M \succcurlyeq \mathbf{0}$ (resp., $M \preccurlyeq \mathbf{0}$) denotes its positive (resp., negative) semidefiniteness. The Kronecker product of two matrices M, N is denoted $M \odot N$. For a locally differentiable operator $f : \mathbb{R}^n \to \mathbb{R}^m$, $J_{f,x} \in \mathbb{R}^{m\times n}$ denotes its Jacobian matrix with respect to (w.r.t.) its operand $x \in \mathbb{R}^n$. $\mathbb{E}$ denotes the standard expectation operator. For a set S, $|S|$ denotes its cardinality. Objects having symmetry are often abbreviated by introducing $*$, e.g., we abbreviate $x^{\mathrm{T}}Px$ and $\begin{bmatrix} P_{11} & P_{21}^{\mathrm{T}} \\ P_{21} & P_{22} \end{bmatrix}$, respectively, as $x^{\mathrm{T}}P[\,*\,]$ and $\begin{bmatrix} P_{11} & * \\ P_{21} & P_{22} \end{bmatrix}$.

II. Problem Statement and Solution Approach

We consider a controlled system of the following form:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), \omega(t)) \\ u(t) &= \pi(x(t)) \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^n$ denotes the given nonlinear CT plant dynamics; $\pi : \mathbb{R}^n \to \mathbb{R}^m$ denotes a state-feedback control policy; and $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $\omega(t) \in \mathbb{R}^d$, respectively, denote the state, the control input, and dynamic parametric variable, at time $t \in \mathbb{R}_{\geq 0}$. The ω-values are assumed bounded within a set $\Theta \subset \mathbb{R}^d$ with $0 \in \Theta$. Also a "safe" operational domain $\mathcal{X} \subseteq \mathbb{R}^n$ containing the origin is specified; operating the system at any $x \notin \mathcal{X}$ is deemed unsafe, and hence must be avoided. For $\theta \in \Theta$, $x^*_\theta \in \mathbb{R}^n$ is an equilibrium of (1) if $f(x^*_\theta, \pi(x^*_\theta), \theta) = 0$. As standardly assumed in [18], [29], [30], and [33], we assume that the equilibrium does not change with parameter variation, i.e., $x^*_\theta \equiv x^*$. Also, without loss of generality (WLOG), through a change of coordinates if needed, we take $x^* = 0$ and $\pi(0) = 0$.

Let Ω denote the space of all $\mathbb{R}^d$-valued parametric evolutions $\omega : \mathbb{R}_{\geq 0} \to \Theta$. For a $\omega \in \Omega$, $\omega^t : [0, t) \to \Theta$ denotes its "t-prefix," i.e., $\omega^t(\tau) = \omega(\tau)\ \forall\ \tau \in [0, t)$. The trajectory of (1) under the parametric evolution $\omega \in \Omega$, when initialized at $x \in \mathbb{R}^n$, is denoted $\psi_\pi(\omega^t, x) \in \mathbb{R}^n$ for any $t \in \mathbb{R}_{\geq 0}$; its existence and uniqueness are assured under the following assumption.

Assumption 1: The plant dynamics $f(\cdot, \cdot, \cdot)$ is locally continuously differentiable.

Assumption 1 implies that $f(\cdot, \cdot, \cdot)$ is locally Lipschitz, which is sufficient for local *existence and uniqueness* of $\psi_\pi(\omega, x)$ uniformly over $\omega \in \Omega$. This assumption also allows for a decomposition of the dynamics into a pair of additive linear and nonlinear parameter-dependent portions, with the latter possessing a "sector bound" (as introduced later in Section III). The stability and safety-related notions used in this article are introduced next.

Definition 1 (Stable Equilibrium, Stabilizing Controller, Stabilizability, Stability, and RoA): For the system (1) and the set of parametric evolutions Ω, if exists a policy $\pi(\cdot)$ and a corresponding neighborhood $\mathcal{R}_{\pi,\Omega}$ of the origin such that uniformly over $\omega \in \Omega$

$$x \in \mathcal{R}_{\pi,\Omega} \Rightarrow \lim_{t\to\infty} \psi_\pi\left(\omega^t, x\right) = 0 \tag{2}$$

then the origin is a Ω-*stable equilibrium* under $\pi(\cdot)$; $\pi(\cdot)$ is a *locally* Ω-*stabilizing controller* (or simply Ω-stabilizing controller); the system is *locally* Ω-*stabilizable* (or simply Ω-stabilizable); the controlled system is *locally* Ω-*stable* (or simply Ω-stable) under $\pi(\cdot)$, and $\mathcal{R}_{\pi,\Omega}$ is a Ω-*region-of-attraction* (Ω-RoA) under $\pi(\cdot)$.

An RoA under certain conditions serves as an RSIS defined next.

Definition 2 (RSIS): For the given safe domain $\mathcal{X}$ and a Ω-stabilizing controller $\pi(\cdot)$, if $\mathcal{S}^{\mathcal{X}}_{\pi,\Omega} \subseteq \mathcal{X}$ is a Ω-RoA of system (1) and satisfies the following:

$$x \in \mathcal{S}^{\mathcal{X}}_{\pi,\Omega} \Rightarrow \psi_\pi\left(\omega^t, x\right) \in \mathcal{X}\ \forall\ t \in \mathbb{R}_{\geq 0} \tag{3}$$

then $\mathcal{S}^{\mathcal{X}}_{\pi,\Omega}$ is an RSIS. The space of all $\mathcal{S}^{\mathcal{X}}_{\pi,\Omega}$'s is denoted $\mathbb{S}^{\mathcal{X}}_{\pi,\Omega}$.

We use the notion of Lipschitz bound to constrain a controller $\pi(\cdot)$, which is formalized as follows.

Definition 3 (Lipschitz Function and Bound) [38]: A function $g{:}\mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$, $\mathcal{Y}$ are domains with $\|\cdot\|_\infty$ defined, is called *Lipschitz* w.r.t. $\|\cdot\|_\infty$ (or simply Lipschitz) if there exists $0 \leq L < \infty$ satisfying

$$\|g(x_1) - g(x_2)\|_\infty < L\|x_1 - x_2\|_\infty\ \forall\ x_1, x_2 \in \mathcal{X} \tag{4}$$

and L is called a *Lipschitz bound*.

The set of state-feedback controls that evaluates to zero at the origin and are Lipschitz-bounded by $L \in \mathbb{R}_{\geq 0}$ is denoted Π_L.

A. Objective and Mathematical Formulation

Given the system (1) satisfying Assumption 1, our first objective is to identify the class of state-feedback NN-based controllers such that any controller in that class is Ω-stabilizing, and possesses a maximal common RSIS. Our next objective is to find an optimal NN-based controller in the identified class (which maximizes a long-run expected utility under random initializations and parametric variations).

WLOG, a controller $\pi(\cdot)$ is written as a superposition of a linear gain "nominal controller" $\pi_K(x) := K.x$ for some $K \in \mathbb{R}^{m\times n}$ and an additive "perturbation controller" $\pi_\rho : \mathbb{R}^n \to \mathbb{R}^m$ around the nominal one, to be implemented via an NN having parameter ρ, i.e., $\pi = \pi_K + \pi_\rho$. Then, for the first objective, we compute an optimal linear state-feedback gain $K^* \in \mathbb{R}^{m\times n}$ for the nominal controller and a maximal Lipschitz bound $L^* \in \mathbb{R}_{\geq 0}$ for the perturbation controller such that the corresponding RSIS $\mathcal{S}^*$ is maximal

$$\begin{aligned} K^*, L^*, \mathcal{S}^* := &\underset{\substack{K\in\mathbb{R}^{m\times n},\\ L\in\mathbb{R}_{\geq 0},\ \mathcal{S}\subseteq\mathcal{X}}}{\operatorname{argmax}}\ [\mathrm{vol}(\mathcal{S}) + \mathfrak{w}.L] \\ &\text{s.t. } \mathcal{S} \in \bigcap_{\pi_\rho\in\Pi_L} \mathbb{S}^{\mathcal{X}}_{(\pi_K+\pi_\rho),\Omega} \end{aligned} \tag{5}$$

where for a compact set $\mathcal{S} \subseteq \mathbb{R}^n$, $\mathrm{vol}(\mathcal{S}) := \int_{\mathcal{S}} 1 dx$ denotes its volume, and $\mathfrak{w} \geq 0$ is a tunable "tradeoff" parameter. Note the objective is to maximize $\mathrm{vol}(\mathcal{S})$ to have a maximal RSIS (the fact that it is a common RSIS is ensured by the constraint $\mathcal{S} \in \bigcap_{\pi_\rho\in\Pi_{L^*}} \mathbb{S}^{\mathcal{X}}_{(\pi_K+\pi_\rho),\Omega}$) and also to maximize L to have the largest possible search space for the candidate NN controllers to be explored later. When the solution set $\mathcal{S}^*$ is nonempty, a state-feedback controller $\pi = \pi_{K^*} + \pi_\rho$ is Ω-stabilizing for any $\pi_\rho \in \Pi_{L^*}$. To achieve the first objective, we develop a sufficient condition of Ω-stabilizability of (1) in Section III-D, which extends the existing Lyapunov-based stability results.

For the next objective, the optimal NN controller $\pi_{\rho^*} \in \Pi_{L^*}$ is designed (so that the overall optimal controller is $\pi^* = \pi_{K^*} + \pi_{\rho^*}$) to maximize an expected utility as defined next. For $\omega \in \Omega$, initial state $x \in \mathcal{S}^*$, a specified reward function $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, and time horizon $T \in \mathbb{R}_{\geq 0}$, let the T-horizon expected utility $J_\pi(\omega^{\mathrm{T}}, x) \in \mathbb{R}$ be

$$J_\pi\left(\omega^{\mathrm{T}}, x\right) := \int_0^{\mathrm{T}} r\left(\psi_\pi\left(\omega^t, x\right), \pi\left(\psi_\pi\left(\omega^t, x\right)\right)\right) dt. \tag{6}$$

Then the optimal perturbation controller $\pi_{\rho^*} \in \Pi_{L^*}$ is computed by solving the following optimization problem:

$$\pi_{\rho^*} := \underset{\pi_\rho\in\Pi_{L^*}}{\operatorname{argmax}} \left[\underset{\substack{\omega\sim\mathbb{P}(\Omega),\\ x\in\mathbb{P}(\mathcal{S}^*)}}{\mathbb{E}} \left[J_{\pi_{K^*}+\pi_\rho}\left(\omega^{\mathrm{T}}, x\right)\right] \right] \tag{7}$$

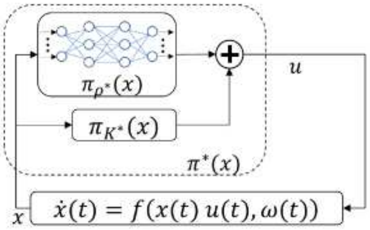

Fig. 1. Block diagram of control architecture.

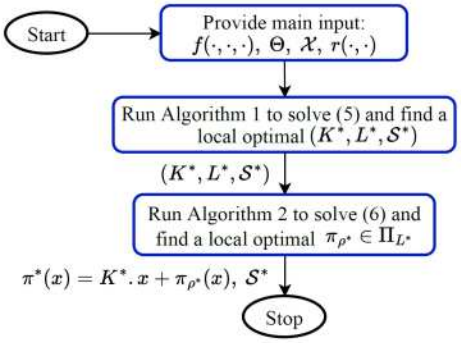
Start
Provide main input: $f(\cdot,\cdot,\cdot)$, Θ, $\mathcal{X}$, $r(\cdot,\cdot)$
Run Algorithm 1 to solve (5) and find a local optimal $(K^*, L^*, \mathcal{S}^*)$
$(K^*, L^*, \mathcal{S}^*)$
Run Algorithm 2 to solve (6) and find a local optimal $\pi_{\rho^*} \in \Pi_{L^*}$
$\pi^*(x) = K^*.x + \pi_{\rho^*}(x)$, $\mathcal{S}^*$
Stop

Fig. 2. Flowchart of the proposed solution approach.

where the distributions $\mathbb{P}(\Omega)$ and $\mathbb{P}(\mathcal{S}^{\mathcal{X}}_{\pi,\Omega})$ in (7) are taken to be uniform in case those are unknown.

A schematic of the overall control architecture is shown in Fig. 1 and a high-level flowchart of the proposed overall method is shown in Fig. 2.

Note (5) and (7) are both nonconvex. We propose Algorithm 1 in Section III-E to iteratively find a local optimal $(K^*, L^*, \mathcal{S}^*)$ solving (5). To find a local optimal control $\pi_{\rho^*} \in \Pi_{L^*}$ solving (7), Algorithm 2 is proposed in Section IV, which extends the traditional actor–critic RL [7] to attain an SGT of the NN controller by way of ensuring its Lipschitz boundedness.

III. Optimal Nominal Control, Maximal Lipschitz Bound for NN Controller, and Maximal RSIS

To enable Ω-stability analysis of the system (1), we introduce in Section III-A an equivalent representation of (1) in the form of a linear system, perturbed by a "nonlinear and parameter variation (NPV)" component, appearing as an additive term. A QC that a Lipschitz-bounded controller $\pi_\rho \in \Pi_L$ necessarily satisfies is presented in Section III-B. In Section III-C, we introduce the notion of "local $(\underline{\mathcal{L}}, \overline{\mathcal{L}})$-sector" to characterize a bound for the NPV. A method to compute the sector-defining parameters $(\underline{\mathcal{L}}, \overline{\mathcal{L}})$ is also presented, and a necessary condition for the NPV to satisfy such a bound in the form of a QC is developed. In Section III-D, given a Lipschitz bound for π_ρ, a sector bound for the system NPV, and a safe operating domain $\mathcal{X} \subset \mathbb{R}^n$, a sufficient condition of Ω-stability of the system (1) is introduced by extending Lyapunov's theory employing the above QCs. This is subsequently used in Section III-E to develop an algorithm to iteratively search for a solution of (5).

A. Equivalent Representation of the Nonlinear System

Following Assumption 1, let (A_θ, B_θ) represent the linearized dynamics of the plant in (1) at the origin for a certain parameter value $\theta \in \Theta$, where, respectively, the state and the input matrices $A_\theta \in \mathbb{R}^{n\times n}$ and $B_\theta \in \mathbb{R}^{n\times m}$ under zero control are defined as: $A_\theta := J_{f,x}|_{\substack{x=0\\u=0}}$ and $B_\theta := J_{f,u}|_{\substack{x=0\\u=0}}$. Then, the nonlinear dynamics under a state-feedback control $u = K.x + u_\rho$ for a $K \in \mathbb{R}^{m\times n}$ and a $u_\rho \in \mathbb{R}^m$ can be written as

$$f\big(x, K.x + u_\rho, \omega\big) = A_{0,K}.x + B_0.u_\rho + \eta_K\big(x, u_\rho, \omega\big) \tag{8}$$

where the pair $(A_{0,K}, B_0)$ denotes the linearized dynamics of (1) at the origin with parameter value $\theta = 0$ under the feedback control $u = K.x + u_\rho$. In other words, $A_{0,K} := (J_{f,x} + J_{f,u}.J_{u,x})|_{\substack{x=0,\theta=0\\u_\rho=0}} \equiv A_0 + B_0.K$. Furthermore, the additive perturbation term is simply the difference

$$\eta_K\big(x, u_\rho, \omega\big) := f\big(x, K.x + u_\rho, \omega\big) - A_{0,K}.x - B_0.u_\rho$$

that is θ-dependent. Ω-stability of the system (1) under a state-feedback controller $u(x) = K.x + \pi_\rho(x)$ is then equivalent to Ω-stability of the following system:

$$\begin{aligned} \dot{x}(t) &= A_{0,K}.x(t) + \underbrace{B_0.u_\rho(t) + \eta_K\big(x(t), u_\rho(t), \omega(t)\big)}_{\text{NPV}\ :\ \zeta_K(x(t), u_\rho(t), \omega(t))} \\ u_\rho(t) &= \pi_\rho(x(t)) \end{aligned} \tag{9}$$

where the effect of the parametric variation and the nonlinearities underlying $f(\cdot,\cdot,\cdot)$ and $u_\rho(\cdot)$ is viewed as a disturbance

$$\zeta_K\big(x, u_\rho, \theta\big) := f\big(x, K.x + u_\rho(x), \theta\big) - A_{0,K}.x$$

additive to the linear system $\dot{x} = A_{0,K}.x$ that we refer to as the "nominal system."

B. Quadratic Condition From Lipschitz-Bounded Control

For an NN-based perturbation controller $\pi_\rho \in \Pi_L$, we define the notion of "L-bounded control-subspace" based on its Lipschitz-bound:

Definition 4 (L-Bounded Control-Subspace): For a Lipschitz bound $L \in \mathbb{R}_{\geq 0}$ and a domain $\mathcal{X} \subset \mathbb{R}^n$, the *$L$-bounded control-subspace* $\mathcal{U}_{L,\mathcal{X}} \subset \mathbb{R}^m$ of a controller $\pi_\rho \in \Pi_L$ is

$$\begin{aligned} \mathcal{U}_{L,\mathcal{X}} := \big\{u_\rho \in \mathbb{R}^m \mid \exists\, x \in \mathcal{X} : \pi_\rho(x) \\ = u_\rho, \|u_\rho\|_\infty \leq L\|x\|_\infty\big\}. \end{aligned} \tag{10}$$

Next, we provide a necessary condition for a controller $\pi_\rho(\cdot) \in \Pi_L$ to be Lipschitz-bounded by L, in form of a QC, which is a variation of [28, Lemma 4.2].

Proposition 1: For a Lipschitz bound $L \in \mathbb{R}_{\geq 0}$, let $\pi_\rho(\cdot) \in \Pi_L$ be a controller (with $\pi_\rho(0) = 0$). Then, there exists $\chi : \mathbb{R}^n \to \mathbb{R}^{mn}$ satisfying $\chi(0) = 0$, such that

$$\pi_\rho(x) = \underbrace{\big[\mathbf{I}_m \odot \mathbf{1}_{1\times n}\big]}_{:=Q}.\chi(x) \tag{11}$$

and the following QC globally holds for all $\gamma_{i,j} \geq 0\ \forall\ i \in 1, \ldots, m, j \in 1, \ldots, n$:

$$\begin{bmatrix} x \\ \chi \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} L^2.\mathrm{diag}\big(\{\Gamma_j\}\big) & \mathbf{0}_{n\times mn} \\ * & \mathrm{diag}\big(\{-\gamma_{i,j}\}\big) \end{bmatrix} [*] \geq 0 \tag{12}$$

where $\Gamma_j := \sum_{i=1}^m \gamma_{i,j}$.

Proof: The proof is provided in Appendix A. ■

C. Bound on Nonlineariy and Parametric Variation

To characterize a bound of the NPV $\zeta_K(\cdot,\cdot,\cdot)$ in (9), we introduce the notion of "local $(\underline{\mathcal{L}},\overline{\mathcal{L}})$-sector".

Definition 5 (Local $(\underline{\mathcal{L}},\overline{\mathcal{L}})$-Sector): For a $K \in \mathbb{R}^{m\times n}$, a Lipschitz bound $L \in \mathbb{R}_{\geq \nvdash}$, and matrices $\underline{\mathcal{L}},\overline{\mathcal{L}} \in \mathbb{R}^{n\times(n+m)}$ satisfying $\underline{\mathcal{L}} \leq \overline{\mathcal{L}}$, the NPV $\zeta_K(x,u_\rho,\theta)$ of system (9) under a controller $\pi_\rho \in \Pi_L$ is said to be *locally $(\underline{\mathcal{L}},\overline{\mathcal{L}})$-sector-bounded* over $\mathcal{X} \subset \mathbb{R}^n$, if the following:

$$\underline{\mathcal{L}}^{i,j} \leq J^{i,j}_{\zeta_K,x}\Big|_{\substack{x=\hat{x}\\ \theta=\hat{\theta}\\ u_\rho=\hat{u}}} \leq \overline{\mathcal{L}}^{i,j}\ \forall\, i,j \in 1,\ldots,n, \text{ and}$$
$$\underline{\mathcal{L}}^{i,j+n} \leq J^{i,j}_{\zeta_K,u_\rho}\Big|_{\substack{x=\hat{x}\\ \theta=\hat{\theta}\\ u_\rho=\hat{u}}} \leq \overline{\mathcal{L}}^{i,j+n}\ \begin{matrix}\forall\, i \in 1,\ldots,n\\ \forall\, j \in 1,\ldots,m\end{matrix} \tag{13}$$

holds uniformly $\forall\, \hat{x} \in \mathcal{X}, \hat{\theta} \in \Theta$, and $\hat{u} \in \mathcal{U}_{L,\mathcal{X}}$, where $\mathcal{U}_{L,\mathcal{X}} \subset \mathbb{R}^m$ denotes the L-bounded control-subspace corresponding to $\mathcal{X}$.

Computation of $(\underline{\mathcal{L}},\overline{\mathcal{L}})$-Sector: Recall $\zeta_K(x,u_\rho,\theta) = f(x,K.x+u_\rho,\theta) - A_{0,K}.x$, and so $J_{\zeta_K,x} = J_{f,x} + J_{f,u}.J_{u,x} - A_{0,K} = J_{f,x} + J_{f,u_\rho}.K - A_{0,K}$ and $J_{\zeta_K,u_\rho} = J_{f,u_\rho}$. Thus following Assumption 1, $J_{f,x}$ and J_{f,u_ρ} are well-defined locally, and so are $J_{\zeta_K,x}$ and J_{ζ_K,u_ρ}. Then, the (i,j)th element of the sector-defining matrices, given a $K \in \mathbb{R}^{m\times n}$, a $L \in \mathbb{R}_{\geq 0}$, and a $\mathcal{X} \subset \mathbb{R}^n$ can be computed as follows. $\forall\, i,j \in 1,\ldots,n$

$$\underline{\mathcal{L}}^{i,j} := \inf_{\substack{\hat{x}\in\mathcal{X},\hat{u}\in\mathcal{U}_{L,\mathcal{X}},\\ \theta\in\Theta}} \left(J^{i,j}_{\zeta_K(\cdot,\cdot,\cdot),x}\Big|_{\substack{x=\hat{x}\\ u_\rho=\hat{u}}}\right)$$
$$\overline{\mathcal{L}}^{i,j} := \sup_{\substack{\hat{x}\in\mathcal{X},\hat{u}\in\mathcal{U}_{L,\mathcal{X}},\\ \theta\in\Theta}} \left(J^{i,j}_{\zeta_K(\cdot,\cdot,\cdot),x}\Big|_{\substack{x=\hat{x}\\ u_\rho=\hat{u}}}\right) \tag{14}$$

and $\forall\, i \in \{1,\ldots,n\},\ j \in \{1,\ldots,m\}$

$$\underline{\mathcal{L}}^{i,j+n} := \inf_{\substack{\hat{x}\in\mathcal{X},\hat{u}\in\mathcal{U}_{L,\mathcal{X}},\\ \theta\in\Theta}} \left(J^{i,j}_{\zeta_K(\cdot,\cdot,\cdot),u_\rho}\Big|_{\substack{x=\hat{x}\\ u_\rho=\hat{u}}}\right)$$
$$\overline{\mathcal{L}}^{i,j+n} := \sup_{\substack{\hat{x}\in\mathcal{X},\hat{u}\in\mathcal{U}_{L,\mathcal{X}},\\ \theta\in\Theta}} \left(J^{i,j}_{\zeta_K(\cdot,\cdot,\cdot),u_\rho}\Big|_{\substack{x=\hat{x}\\ u_\rho=\hat{u}}}\right). \tag{15}$$

Note for simplicity, the infima (resp., suprema) in (14) and (15) can be relaxed by replacing those with the respective lower (resp., upper) bounds at the cost of slight conservativeness to the sector. The value of each such bound can be computed to a desired degree of accuracy via a binary search using a satisfiability-modulo-theory (SMT) solver (such as dReal [39]), wherein the constraints regarding a postulated lower/upper bound, the boundedness of state domain, the L-boundedness of control subspace, and the parametric set Θ get represented as the conjunction of first-order formulas over the reals.

Next, a necessary condition for $\zeta_K(x,u_\rho,\theta)$ to be $(\underline{\mathcal{L}},\overline{\mathcal{L}})$-sector-bounded locally over $\mathcal{X} \subset \mathbb{R}^n$ is proposed, in the form of a $(K,L,\mathcal{X},\Theta)$-dependent QC.

Proposition 2: For a $K \in \mathbb{R}^{m\times n}$ and a $L \in \mathbb{R}_{\geq 0}$, consider the $\mathbb{R}^n$-valued NPV $\zeta_K(\cdot,\cdot,\cdot)$ of a system (9) that is locally $(\underline{\mathcal{L}},\overline{\mathcal{L}})$-sector-bounded over $\mathcal{X} \subset \mathbb{R}^n$ under a controller $u_\rho(t) = \pi_\rho(x(t))$ with $\pi_\rho(\cdot) \in \Pi_L$. Then, for each $\theta \in \Theta$, a $\xi_\theta : \mathbb{R}^n \to \mathbb{R}^{n(n+m)}$ exists satisfying $\xi_\theta(0) = 0$ such that

$$\zeta_K\big(x,u_\rho,\theta\big) = \underbrace{\big[\mathbf{I}_n \odot \mathbf{1}_{1\times(n+m)}\big]}_{:=R}.\xi_\theta(x),\ \forall\, x \in \mathcal{X}. \tag{16}$$

Furthermore, for $i \in 1,\ldots,n$ and $j \in 1,\ldots,n+m$, let $c_{ij} := (\underline{\mathcal{L}}^{i,j} + \overline{\mathcal{L}}^{i,j})/2$, $\overline{c}_{ij} := \max(|\underline{\mathcal{L}}^{i,j}|,|\overline{\mathcal{L}}^{i,j}|)$, and $k_{ij} := i + (j-1)n$. Then, uniformly for any $\theta \in \Theta$, $\pi_\rho(\cdot) \in \Pi_L$, and $\Lambda \in \mathbb{R}^{n.(n+m)} \geq 0$, the following locally holds:

$$\begin{bmatrix} x\\ \chi\\ \xi_\theta \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} M_{x\Lambda} & \mathbf{0}_{n\times m.n} & N_{x\Lambda}\\ * & M_{\chi\Lambda} & N_{\chi\Lambda}\\ * & * & M_{\xi\Lambda} \end{bmatrix} [*] \geq 0,\ \forall\, x \in \mathcal{X} \tag{17}$$

where recall $u_\rho = \pi_\rho(x) = Q.\chi(x)$ from (11), and the matrices $M_{x\Lambda}, M_{\chi\Lambda}, M_{\xi,\Lambda}, N_{x\Lambda}$, and $N_{\chi\Lambda}$ are defined as follows:

$$\begin{aligned}
M_{x\Lambda} &:= \mathrm{diag}\left(\left(\sum_{i=1}^{n} \Lambda^{k_{ij}}\left(\overline{c}^2_{ij} - c^2_{ij}\right) \mid j \in 1,\ldots,n\right)\right)\\
M_{\chi\Lambda} &:= Q^{\mathrm{T}}\mathrm{diag}\left(\left(\sum_{i=1}^{n} \Lambda^{k_{ij}}\left(\overline{c}^2_{ij} - c^2_{ij}\right) \mid \right.\right.\\
&\qquad\qquad \left.\left. j \in n+1,\ldots,n+m\right)\right)Q\\
M_{\xi\Lambda} &:= \mathrm{diag}(-\Lambda)\\
N_{x\Lambda} &:= \big[D_{x,1}\ D_{x,2}\ \ldots\ D_{x,n}\big], \text{ where:}\\
D_{x,i} &:= \left[\mathrm{diag}\left(\left(\Lambda^{k_{ij}}.c_{ij} \mid j \in 1,\ldots,n\right)\right)\ \mathbf{0}_{n\times m}\right]\\
N_{\chi\Lambda} &:= Q^{\mathrm{T}}.\big[D_{u,1}\ D_{u,2}\ \ldots\ D_{u,n}\big], \text{ where:}\\
D_{\chi,i} &:= \left[\mathbf{0}_{m\times n}\ \ \mathrm{diag}\left(\left(\Lambda^{k_{ij}}.c_{ij} \mid j \in n+1,\ldots,n+m\right)\right)\right].
\end{aligned} \tag{18}$$

Proof: The proof is provided in Appendix B. ■

D. Lyapunov-Based Ω-Stability Certification

We begin by recalling some existing Lyapunov-based stability-related results.

Definition 6 (Common Lyapunov Function (CLF) [40], [41]): Consider the system (1) under a given controller $\pi(\cdot)$. A continuously differentiable function $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$, where $\mathcal{X} \subset \mathbb{R}^n$ is a compact domain containing the origin, is a CLF if uniformly for each $\omega \in \Omega$

$$\begin{aligned}
&V(x) > 0,\ \dot{V}(x) < 0\ \forall\, x \in \mathcal{X}\backslash\{0\}\\
&V(0) = \dot{V}(0) = 0.
\end{aligned} \tag{19}$$

It is known that if a CLF exists for the system (1), then the system is Ω-stable, $\pi(\cdot)$ is Ω-stabilizing, and the origin is a Ω-stable equilibrium [40], [41]. However, in general, finding a $\pi(\cdot)$ and its corresponding CLF is challenging.

Taking $\pi(\cdot)$ to be of form $\pi(x) = \pi_K(x) + \pi_\rho(x)$ for a $K \in \mathbb{R}^{m\times n}$ and a $\pi_\rho \in \Pi_L$, along with the QC characterizations of the bound of $\pi_\rho(\cdot)$ and the local $(\underline{\mathcal{L}},\overline{\mathcal{L}})$-sector bound of the NPV in (9) (see Sections III-A–III-C), enables an efficient search for a CLF as demonstrated next: We state our key theorem, that for a given $(K,L) \in \mathbb{R}^{m\times n} \times \mathbb{R}_{\geq 0}$, enables the verification of whether a state-feedback controller $\pi = \pi_K + \pi_\rho$ is Ω-stabilizing for the system (1) uniformly for each $\pi_\rho \in \Pi_L$, by way of a search for a quadratic CLF.

Theorem 1: Given $L \in \mathbb{R}_{\geq 0}$ and a neighborhood of the origin $\mathcal{X} \subset \mathbb{R}^n$, consider the system in (1) under a controller $\pi(x) = \pi_K(x) + \pi_\rho(x)$ satisfying Assumption 1, where $K \in \mathbb{R}^{m \times n}$ and $\pi_\rho \in \Pi_L$, so that its equivalent representation of (9) and a corresponding local $(\underline{\mathcal{L}}, \overline{\mathcal{L}})$-sector bound for its NPV exist. Then, the system is Ω-stable at the origin, uniformly for each $\pi_\rho \in \Pi_L$, if exist $K \in \mathbb{R}^{m \times n}$, $P \succ 0$, $\Lambda \geq 0$, and $\gamma_{ij} \geq 0$ for all $i \in 1, \ldots, m$, $j \in 1, \ldots, n$ satisfying

$$\begin{bmatrix} V_{L,\{\Gamma_j\},P,K} & * & * \\ \mathbf{0}_{m.n \times n} & M_{\chi\Lambda} - \mathrm{diag}\big(\{\gamma_{ij}\}\big) & * \\ N_{x\Lambda}^{\mathrm{T}} + R^{\mathrm{T}}.P & N_{\chi\Lambda}^{\mathrm{T}} & M_{\xi\Lambda} \end{bmatrix} \prec 0 \quad (20)$$

where recall $\Gamma_j = \sum_{i=1}^{m} \gamma_{i.j}$, and $V_{L,\{\Gamma_j\},P,K}$ is defined as

$$V_{L,\{\Gamma_j\},P,K} = M_{x\Lambda} + L^2.\mathrm{diag}\big(\{\Gamma_j\}\big) + P.A_{0,K} + A_{0,K}^{\mathrm{T}}.P. \quad (21)$$

Proof: See Appendix C. ■

Recall the matrices $M_{x\Lambda}$, $M_{\chi\Lambda}$, $N_{x\Lambda}$, and $N_{\chi\Lambda}$ are derived from the $(\underline{\mathcal{L}}, \overline{\mathcal{L}})$-sector, which reveals their inherent $(K, L, \mathcal{X}, \Theta)$-dependence. For a certain $(L, \mathcal{X}, \Theta)$, the presence of the bilinear terms in (20) makes the latter nonconvex when both K and P are search variables. On the other hand, if a K is given, (20) becomes an LMI that can be solved efficiently, and the existence of a feasible $(P \succ 0, \Lambda \geq 0, \{\lambda_{ij} \geq 0\})$ certifies the Ω-stability of (1) with the corresponding $V(x) = x^{\mathrm{T}}.P.x$ serving as a CLF. Our Algorithm 1 in the next section enables a local search for a quadruple $(K, P \succ 0, \Lambda \geq 0, \{\lambda_{ij} \geq 0\})$ satisfying (20).

Corollary 1 (Existence of RSIS): Consider the setting of Theorem 1. If the LMI (20) is feasible for a $P \succ 0$, then exists $\sigma > \mathbb{R}_{>0}$ such that the ellipsoid $\mathcal{E}_{P,\sigma} := \{x \in \mathbb{R}^n | x^{\mathrm{T}}Px \leq \sigma\}$ is contained in a given safe domain $\mathcal{X} = \{x \in \mathbb{R}^n | a_i^{\mathrm{T}}.x \leq b_i, i \in 1, \ldots, n_{\mathcal{X}}\}$ and serves as an inner-estimate of the maximal RSIS of the system (1), uniformly for each $\pi_\rho \in \Pi_L$.

Proof: See Appendix D. ■

Remark 1: Our proposed method is applicable in presence of any actuator saturation if it can be modeled by a continuously differentiable map. For example, if $g : \mathbb{R}^m \to \mathbb{R}^m$ is a continuously differentiable input saturation model (e.g., elementwise sigmoid, tan-hyperbolic, etc.), then its effect can be subsumed within the dynamics of (1) as: $\dot{x} = f(x, g(u), \omega)$. Also, note that (9) expresses the original system as the superposition of a linear nominal plant and a nonlinear time-varying part to capture nonlinearity and parametric variation, which can be sector-bounded over any domain of interest (via Assumption 1). Thus, the effect of any additive random disturbance that is sector-bounded can also be similarly incorporated into the proposed framework.

E. Optimal Nominal Control, Maximal Lipschitz Bound for NN Controller, and Inner Estimate of Maximal RSIS

We employ Theorem 1 and Corollary 1 to devise an iterative method of solving (5) in Algorithm 1, which finds a locally Pareto optimal pair (K^*, L^*) and an inner estimate of its corresponding maximal RSIS $\mathcal{S}^*$, where for computational purposes, the parametric set Θ as well as the safe operational domain $\mathcal{X}$ are taken to be polytopic, with $\mathcal{X} := \{x \in \mathbb{R}^n | a_i^{\mathrm{T}}.x \leq b_i, i \in 1, \ldots, n_{\mathcal{X}}\}$. The strategy is to find a (K^*, L^*), a corresponding $P^* \succ 0$, and the largest "safety sublevel-set" denoted $\mathcal{X}^* := \{x | a_i^{\mathrm{T}}.x \leq \delta^*.b_i, i \in 1, \ldots, n_{\mathcal{X}}; \delta^* \in (0, 1]\}$ where (20) is feasible. Next, following Corollary 1, the largest hyperellipse $\mathcal{E}_{P*,\sigma*}$ contained in $\mathcal{X}^*$ is output as an inner estimate of $\mathcal{S}^*$.

We begin with $L = \delta = 0$, i.e., with a linear controller (since $L = 0$) and the safety sublevel-set restricted to the origin (since $\delta = 0$), over which the nonlinear dynamics is equivalent to the linear dynamics (A_θ, B_θ) under the control of a nominal linear controller $\pi(x) = \pi_K(x) = K.x$. The initialization of Algorithm 1 requires computing a polytopic bound of (A_θ, B_θ) for any $\theta \in \Theta$. Let $\mathcal{I}$ denote the set of indices of θ-dependent elements in (A_θ, B_θ). Note $|\mathcal{I}| \leq n^2 + m.n$. For each $\wp \subseteq \mathcal{I}$, let $(A_\wp, B_\wp)$ be obtained by replacing the θ-dependent elements of (A_θ, B_θ) corresponding to the indices in $\wp$ (resp. $\mathcal{I} \backslash \wp$) with their respective upper (resp. lower) bounds over Θ. Then, for any $\theta \in \Theta$, (A_θ, B_θ) belongs to the polytope with $(A_\wp, B_\wp)$'s as the vertices, i.e.,

$$\forall\, \theta \in \Theta : [A_\theta \;\; B_\theta] = \sum_{\wp \subseteq \mathcal{I}} \gamma_\wp \big[A_\wp \;\; B_\wp\big] \quad (22)$$

where $\gamma_\wp \in [0, 1]$ such that $\sum_{\wp \subseteq \mathcal{I}} \gamma_\wp = 1$. To find the vertices $(A_\wp, B_\wp)$'s, the bounds of its respective θ-dependent elements can be computed via an SMT solver-based search (similar to that for the elements of $(\underline{\mathcal{L}}, \overline{\mathcal{L}})$ in Section III-C).

The granularity of the search in Algorithm 1 and hence that of the computed locally optimal solution is decided by the number of iterations n_{step} (a user-selected parameter) that is used to iteratively enlarge (by a fixed amount $1/n_{\text{steps}}$ in each iteration) the safety sublevel set of search from the initial singleton point—the origin—to finally the entire specified safety domain $\mathcal{X}$. The algorithm is initialized with a (K, P) found by the convex search of (23) employing $(A_\wp, B_\wp)$'s as parameters, which ensures that $\pi_K(x)$ is Ω-stabilizing for the *linearized model* (A_θ, B_θ) [42, pp. 100–102], while the initializing P defines a quadratic CLF whose maximal sublevel set inscribed in $\mathcal{X}$ upper bounds the volume of $\mathcal{E}_{P*,\sigma*}$ of the *nonlinear* model [43, p. 411]. In each successive iteration, to deal with the nonconvexity of (20) when finding (K, P) together, we split the search into two successive convex problems (24) and (25) in each iteration. In (24), holding P fixed at its most recent value, we search for a K in the neighborhood of its most recent iterate, subject to (20), while keeping the matrices $M_{x\Lambda}$, $M_{\chi\Lambda}$, $N_{x\Lambda}$, and $N_{\chi\Lambda}$ unchanged, i.e., ignoring the effect on their value due to a change in K over its past iterate. Next in (25), those missing effects are restored when searching for a feasible P in the neighborhood of its most recent iterate, while keeping K fixed at its most recent value.

In an iteration, if (24) and (25) are both feasible, then the resultant K, P satisfy (20) for the current L, δ-sublevel set of $\mathcal{X}$, $\Lambda \geq 0$, and $\gamma_{ij} \geq 0$ for all $i \in 1, \ldots, m$ and $j \in 1, \ldots, n$. σ corresponding to the largest sublevel set $\mathcal{E}_{P,\sigma}$ of the CLF $V(x) = x^{\mathrm{T}}.P.x$ contained in the δ-sublevel set of $\mathcal{X}$, is then computed solving (26) by a quasi-convex search [44],

Algorithm 1 Iterative Local-Optimal Solution of (5)

```
Input: The dynamic model $f(\cdot,\cdot,\cdot)$ and its parametric set $\Theta$, the tradeoff parameter $\mathfrak{w} \in \mathbb{R}_{\geq 0}$, the maximum iterative steps $n_{\text{steps}}$, and the safe domain: $\mathcal{X} = \{x | a_i^T.x \leq b_i, i \in 1,\ldots,n_{\mathcal{X}}\}$.
Initialize: $k = 1$, $\Delta = 1/n_{\text{steps}}$, $\delta^0 = L^0 = 0$, $P^0 = Q^{-1}$, $K^0 = Y.Q^{-1}$, where $Q \in \mathbb{R}^{n\times n}$ and $Y \in \mathbb{R}^{m\times n}$ are found as:
    $Q, Y = \operatorname*{argmax}_{Q \succ 0, Y \in \mathbb{R}^{m\times n}} \ln(\det(Q))$[†]
    s.t. $Q.{A_\wp}^T + A_\wp.Q + B_\wp.Y + Y^T.B_\wp^T \prec 0 \quad \forall\, \wp \subseteq \mathcal{I},$
         $\|Q.a_i\|_2 \leq b_i \ \forall\, i \in 1,\ldots,n_{\mathcal{X}},$        (23)
where $(A_\wp, B_\wp)$'s are such that (22) holds.
while $k \leq n_{\text{steps}}$ do
    $\delta^k = \delta^{k-1} + \Delta,\ L^k = L^{k-1} + \mathfrak{w}.\Delta$
    $\mathcal{X}^k = \{x | a_i^T.x \leq \delta^k.b_i, i \in 1,\ldots,n_{\mathcal{X}}\}$
    Compute $M_{x\Lambda}, M_{\chi\Lambda}, N_{x\Lambda}, N_{\chi\Lambda}$ using (18) corresponding to $(K^{k-1}, L^k, \mathcal{X}^k, \Theta)$ and find $K^+$:
        $K^+ = \operatorname*{argmin}_{K \in \mathbb{R}^{m\times n}, \Lambda \geq 0, \left\{\gamma_{ij} \geq 0 \,\middle|\, {i \in 1,\ldots,m, \atop j \in 1,\ldots,n}\right\}} \|K - K^{k-1}\|_2$        (24)
        s.t.: LMI in (20) given $P = P^{k-1}$
    if (24) is Feasible, then
        Update $M_{x\Lambda}, M_{\chi\Lambda}, N_{x\Lambda}, N_{\chi\Lambda}$ for $(K^+, L^k, \mathcal{X}^k, \Theta)$ and find $P^+$
            $P^+ = \operatorname*{argmin}_{P \succcurlyeq 0, \Lambda \geq 0, \left\{\gamma_{ij} \geq 0 \,\middle|\, {i \in 1,\ldots,m, \atop j \in 1,\ldots,n}\right\}} \|P - P^{k-1}\|_2$        (25)
            s.t.: LMI in (20) given $K = K^+$
    end if
    if (24) is Infeasible or (25) is Infeasible, then
        break
    else
        $K^k = K^+$, $P^k = P^+$, and
            $\sigma^k = \max_{\sigma \in \mathbb{R}_{\geq 0}, x \in \mathcal{X}^k} \sigma$        (26)
            s.t.: $x^T.P^k.x \leq \sigma.$
        Store $(K^k, L^k, P^k, \sigma^k)$
        $k \leftarrow k + 1$
    end if
end while
Output: Find $\ell^* = \operatorname*{argmax}_{\ell}(\{\sigma^\ell/\det([P^\ell]^{-1}) + \mathfrak{w}.L^\ell | \ell \in 1,\ldots,k\})$ where $\sigma^\ell/\det([P^\ell]^{-1}) \equiv vol(\mathcal{E}_{P^\ell,\sigma^\ell})$, and output $K^* = K^{\ell^*}$, $L^* = L^{\ell^*}$, $P^* = P^{\ell^*}$, $\sigma^* = \sigma^{\ell^*}$.
```

and the current iterate (K, L, P, σ) is stored. Iterating forward, as δ changes in fixed increments, the feasible domain for $\mathcal{E}_{P,\sigma}$ increases, whereas the amount of nonlinearity allowed also increases, restricting the $\mathcal{E}_{P,\sigma}$. Due to this dual effect, the growth of $\mathcal{E}_{P,\sigma}$ over the iterations is not necessarily monotonic. Hence, on termination of the iterative loop, the stored iterate that corresponds to the largest value of $(\sigma/\det(P^{-1})) + \mathfrak{w}.L$ is reported as the locally optimal output, where note $\sigma/\det(P^{-1})$ is simply a measure of $vol(\mathcal{E}_{P,\sigma})$ in the objective of (5).

Remark 2: The optimizations (24)–(26) within Algorithm 1 are convex, and as such their worst case solution complexity employing a standard interior point method scales polynomially with respect to state dimension n and the control dimension m [45]. Also, since Algorithm 1 iterates at the most n_{step} times to explore the search space, its complexity grows linearly with n_{step}, Thus, the overall complexity still is polynomial in m, n, n_{step}.

IV. Optimal NN Control and Stability-Guaranteed Training

In this section, given the output of Algorithm 1, i.e., given a (K^*, L^*) and a corresponding $\mathcal{E}_{P^*,\sigma^*}$ that is the largest hyperelliptical inner estimate of the maximal RSIS $\mathcal{S}^*$, our goal is to solve (7) to find the NN-based "perturbation controller" $\pi_{\rho^*} \in \Pi_{L^*}$ such that the overall controller $\pi^*(x) = K^*.x + \pi_{\rho^*}(x)$ maximizes the expected long-run utility of the closed-loop system (1) under parametric variations $\omega \sim \mathbb{P}(\Omega)$ and random initializations $x(0) \sim \mathbb{P}(\mathcal{E}_{P^*,\sigma^*})$. Our gradient descent-based SGT to search for a locally optimal ρ^*, which extends the traditional "actor–critic" RL [3], [7], is presented in Algorithm 2. It should be noted that although Algorithm 2 extends actor–critic RL, the approach proposed in this article is general enough to be applied to any machine-learning-based deterministic controller design algorithm (e.g., imitation learning [1], [2], deterministic policy gradient-based RL methods including the "off-policy" ones [6], [46], etc.).

Let $x(k) \in \mathbb{R}^n$ and $u_\rho(k) \in \mathbb{R}^m$, respectively, denote the state and the NN-based perturbation control values at the kth discrete sample instant under a uniform sampling period $\tau \in \mathbb{R}_{>0}$, available for training the NN π_ρ. Then, the integral involved in defining the system's utility in (6) can be approximated by the corresponding discrete sum. Accordingly, the "value" of a state $x \in \mathcal{E}_{P^*,\sigma^*}$ employing an NN controller $\pi_\rho(\cdot)$, denoted $v_{\pi_\rho}(x) \in \mathbb{R}$, is [3]

$$v_{\pi_\rho}(x) := \mathop{\mathbb{E}}_{\omega\sim\mathbb{P}(\Omega)}\left[\sum_{k=0}^{\infty} r(x(k), u(k)) \,\middle|\, \begin{aligned} &\pi = \pi_{K^*} + \pi_\rho \\ &x(k) = \psi_\pi\left(\omega^{k.\tau}, x\right),\ u(k) = \pi(x(k)) \end{aligned}\right]. \tag{27}$$

Then, the optimal NN controller $\pi_{\rho^*}(\cdot)$ is characterized by Bellman's optimality condition [47]

$$\begin{aligned} v^*(x(k)) = {} & r\big(x(k), \pi_{K^*}(x(k)) + \pi_{\rho^*}(x(k))\big) \\ & + \mathop{\mathbb{E}}_{\omega\sim\mathbb{P}(\Omega)} v^*(x(k+1)) \end{aligned} \tag{28}$$

where $v^*(x) := \max_\rho v_{\pi_\rho}(x)$.

As commonly practiced, in Algorithm 2, the value function is approximated by the "critic" NN denoted $\hat{v}_\phi(\cdot)$, while the "actor" NN $\pi_\rho(\cdot)$ serves as the controller. Both NNs are jointly trained over n_t number of training trajectories, each

[†] In (23), $\ln(\cdot)$ denotes natural logarithm of its real scalar argument, and for a square matrix M, $\det(M)$ denotes its determinant.

comprising n_s number of discrete time steps. To enable effective exploration of the control space, at each training step, we choose $u_\rho(k)$ randomly from the Gaussian distribution $\mathcal{N}(\pi_\rho(x(k)), \Sigma)$ with mean $\pi_\rho(x(k))$ and covariance matrix $\Sigma \in \mathbb{R}^{m\times m}$. Σ is initialized as a user-specified non-negative diagonal matrix, the elements of which are uniformly scaled down as the training proceeds. At the end of the training, the deterministic NN controller π_{ρ^*} is deployed as the optimal perturbation controller.

To improve training robustness, the n_a-step average of the computed gradients is used as the estimate of the true gradient in contrast to a single-step gradient estimate. To ensure Ω-stability of the overall controller π, we constrain the search space of the NN controller π_ρ within Π_{L^*} by the following means: 1) we add to the policy gradient a regularizer (see line 13 of Algorithm 2) proportional to the change in Lipschitz bound $L_{\pi_\rho} \in \mathbb{R}_{\geq 0}$ of $\pi_\rho(\cdot)$, estimated using the computationally efficient method of [48] (with $\beta \in \mathbb{R}_{\geq 0}$ serving as a weight) and 2) the elements of ρ are uniformly scaled if the parameter update in a training step results in $L_{\pi_\rho} > L^*$ (see lines 14 and 15 of Algorithm 2).

Remark 3: Like any other RL algorithm for multilayered NN controller training, the resultant policy from Algorithm 2 is only locally optimal, in general. But it enjoys the added property of Ω-stability of the controlled system, and the guarantee that the computed ellipse $\mathcal{E}_{P^*,\sigma^*} \subset \mathcal{X}$ is its RSIS. Computationally, Algorithm 2 adds only the complexity of the lines 14–16 to that of a standard actor–critic deep RL algorithm [7]. This additional complexity scales linearly with the number of actor NN layers and quadratically with the maximum number of neurons within a layer of the actor NN [48]. Notably, Algorithm 2 does not involve any LMI solution within its optimization loop unlike the recently proposed method in [49].

V. Illustrative Example

To validate the correctness and effectiveness of our proposed method, we consider the following illustrative nonlinear system of the form (1) possessing continuously differentiable dynamics (here, the ith element of $x \in \mathbb{R}^n$ is denoted x_i)

$$\dot{x} = \begin{bmatrix} -(1+\omega_1)x_2 \\ x_1 + (1+\omega_2)\left(x_1^2 - 1\right)x_2 \end{bmatrix} + u$$
$$u = \pi(x) \tag{29}$$

where $\omega = [\omega_1\ \omega_2]^{\mathrm{T}}$ denotes the vector of time-varying parameters bounded within the range $\Theta \equiv [-0.05, 0.05] \times [-0.1, 0.1]$. If $\pi(0) = 0$, regardless of ω-value, the origin is an equilibrium of the above system. Let the reward function and the safe domain of the system be, respectively, given as: $r(x, u) = -(x^{\mathrm{T}}.x + 0.1u^{\mathrm{T}}.u)$ and a polytope $\mathcal{X} \subset \mathbb{R}^n$ with vertices $(0.3, 0.6)$, $(0.1962, 0.8077)$, $(-0.3375, 0.1406)$, $(-0.3375, -0.8523)$, $(0.3, -0.2727)$ as shown in Fig. 3.

Our objective is to find a Ω-stabilizing $\pi^*(\cdot)$ and a corresponding maximal RSIS $\mathcal{S}^{\mathcal{X}}_{\pi^*,\Omega} \subset \mathcal{X}$ so that the expected long-run utility of (6) is maximized under random parametric variation in $\Theta \equiv [-0.05, 0.05] \times [-0.1, 0.1]$ and state initializations within $\mathcal{S}^{\mathcal{X}}_{\pi^*,\Omega}$. As proposed, $\pi^*(x) = K^*.x + \pi_{\rho^*}(x)$ with $\pi_{\rho^*} \in \Pi_{L^*}$, where (K^*, L^*) and the associated hyperelliptical inner estimate of the maximal $\mathcal{S}^{\mathcal{X}}_{\pi^*,\Omega}$ are found employing Algorithm 1, and $\pi_{\rho^*} \in \Pi_{L^*}$ is implemented using an NN,

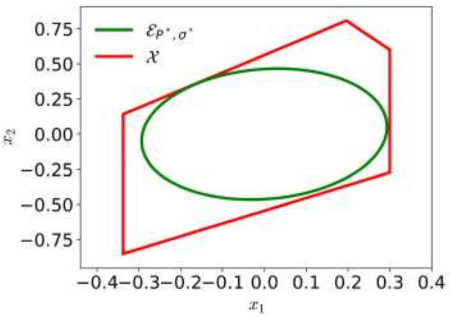

Fig. 3. Maximal elliptical RSIS inner estimate $\mathcal{E}_{P^*,\sigma^*} \subset \mathcal{X}$.

Algorithm 2 Actor–Critic RL With Stability Guarantee

Input: Actor and critic NNs parameterized by ρ and ϕ, sampling interval $\tau \in \mathbb{R}_{>0}$, training step sizes $\alpha_\rho, \alpha_\phi \in \mathbb{R}_{>0}$, a diagonal matrix $\Sigma \in \mathbb{R}^{m\times m}$ s.t. $\Sigma \geq 0$, decay rate of exploration $\nu_d \in (0, 1)$ and its minimum value $\nu_{\min} \in (0, 1)$, no. of training trajectories n_t, integers n_s and n_a s.t. $n_s.\tau = T$ and $n_a < n_s$, tradeoff parameter $\beta \in \mathbb{R}_{\geq 0}$, and as introduced before $f(\cdot,\cdot,\cdot)$, (K^*, L^*), $r(\cdot,\cdot)$, $\mathbb{P}(\mathcal{E}_{P^*,\sigma^*})$, and $\mathbb{P}(\Omega)$.

Initialize: Exploration coefficient $\nu = 1$, trajectory count $e = 1$, initialize ρ, ϕ in their respective parameter spaces.

```
1:  while $e \leq n_t$ do
2:      Set gradients $d\rho = 0$, $d\phi = 0$, sample index $k = 0$;
3:      Randomly choose $x(0) \sim \mathcal{E}_{P^*,\sigma^*}$ and $\omega \sim \mathbb{P}(\Omega)$;
4:      while $k < n_s$ do
5:          Given $x(k)$, $\omega(t)\ \forall\ t \in [k.\tau, (k+1).\tau)$, apply random control $u_\rho(k) \sim \mathcal{N}(\pi_\rho(x(k)), \Sigma)$ through a zero-order hold to observe $x(k+1)$, and compute reward $r(k) := r(x(k), \pi_{K^*}(x(k)) + u_\rho(k))$;
6:          if $k \geq n_a$ then
7:              Compute $n_a$-step advantage: $a(k) := \sum_{l=0}^{N-1} r(k-l) + \hat{v}_\phi(x(k+1)) - \hat{v}_\phi(x(k-n_a+1))$;
8:              $d\rho \leftarrow [(k-n_a)d\rho + \{(u(k) - \pi_\rho(x(k)))^T .\Sigma^{-1}\nabla_\rho\pi_\rho(x(k))\}a(k)]/(k-n_a+1)$;‡
9:              $d\phi \leftarrow [(k-n_a)d\phi + \nabla_\phi\{\hat{v}_\phi(x(k-n+1)) - \hat{v}_\phi(x(k+1))\}.a(k)]/(k-n_a+1)$;‡
10:             $k \leftarrow k+1$;
11:         end if
12:     end while
13:     $\rho \leftarrow \rho + \alpha_\rho(d\rho - \beta.\nabla_\rho L_{\pi_\rho})$;
14:     if $L_{\pi_\rho} > L^*$ then
15:         $\rho \leftarrow \rho\left(\frac{L^*}{L_{\pi_\rho}}\right)^{\frac{1}{n_l}}$;        ▷ $n_l =$ # layers in $\pi_\rho(\cdot)$
16:     end if
17:     $\phi \leftarrow \phi - \alpha_\phi.d\phi$, $\nu \leftarrow \max(\nu_{\min}, \nu.\nu_d)$, $\Sigma \leftarrow \nu.\Sigma$;
18:     $e \leftarrow e+1$;
19: end while
```

Output: Local optimal parameter $\rho^* = \rho$ for actor NN.

‡For a scalar differentiable function $f(x)$ of $x \in \mathbb{R}^n$, $\nabla_x f(x_0) \in \mathbb{R}^n$ denotes the gradient of f w.r.t. x at $x = x_0$.

with its parameter ρ^* trained using Algorithm 2, maximizing the expected utility.

A. Computation of (K^*, L^*) and Inner Estimate of $\mathcal{S}^{\mathcal{X}}_{\pi^*,\Omega}$

Following Assumption 1, by linearizing the dynamics of (29) at $x = u = 0$, the matrices A_θ and B_θ for a $\theta \in \Theta$ are derived as

$$A_\theta = \begin{bmatrix} 0 & -(1+\theta_1) \\ 1 & -(1+\theta_2) \end{bmatrix},\ B_\theta = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{30}$$

Since two of eight elements of (A_θ, B_θ) are θ-dependent, $2^2 = 4$ $(A_\wp, B_\wp)$ vertices are computed such that (22) holds. Using those as parameters, we first solve (23) and find a feasible pair

$$K^0 = \begin{bmatrix} -2.8299 & 0.3352 \\ 1.9226 & -0.9035 \end{bmatrix},\ P^0 = \begin{bmatrix} 3.6841 & -0.5629 \\ -0.5629 & 1.7448 \end{bmatrix}$$

that certifies the Ω-stabilizability of (29) along with the existence of a neighborhood of the origin as its Ω-RoA under the Ω-stabilizing controller $\pi_{K^0}(x) = K^0.x$.

Next we initialize Algorithm 1 with (K^0, P^0) and run it using $\mathfrak{w} = 1.1$ and over $n_{\text{steps}} = 20$ iterations to search for (K^*, L^*). At any iteration $k \leq n_{\text{steps}}$, the elements of matrices $\underline{\mathcal{L}}^k$ (resp., $\overline{\mathcal{L}}^k$) are conservatively computed to the accuracy of 0.001 via binary search employing the SMT solver dReal [39]. The solutions of the convex problems (24) and (25) certify the Ω-stability of the nonlinear system (29) under the control of $\pi^k = \pi_{K^k} + \pi_\rho$ for any $\pi_\rho \in \Pi_{L^k}$ and any initialization within $\mathcal{E}_{P^k,\sigma^k} \subseteq \mathcal{X}^k \subseteq \mathcal{X}$ obtained by solving (26). The iterative loop continues for $n_{\text{steps}} = 20$ iterations, yielding (K^*, L^*) and the corresponding P^* as

$$K^* = \begin{bmatrix} -2.9714 & -0.1204 \\ 1.5924 & -2.1744 \end{bmatrix},\ L^* = 1.1$$
$$P^* = \begin{bmatrix} 3.8426 & -0.2612 \\ -0.2612 & 1.5241 \end{bmatrix}. \tag{31}$$

Also, the level value $\sigma^* = 0.3272$ for defining the hyperellipse $\mathcal{E}_{P^*,\sigma^*}$ is computed solving (26).

Recall $V(x) = x^{\mathrm{T}}.P^*.[\,*\,]$ is a CLF over $\mathcal{E}_{P^*,\sigma^*}$ for the system (29) under any controller $\pi = \pi_{K^*} + \pi_\rho$ with $\pi_\rho \in \Pi_{L^*}$, which certifies the Ω-stability of the system at the origin according to Theorem 1. Also, $\mathcal{E}_{P^*,\sigma^*}$ serves as a hyperelliptical inner estimate of $\mathcal{S}^{\mathcal{X}}_{\pi,\Omega}$ following Corollary 1. The computed $\mathcal{E}_{P^*,\sigma^*} \subset \mathcal{X}$ is shown in Fig. 3.

To illustrate the principle underlying the algorithm, the evolution of the eigenvalues of A_{0,K^k}, i.e., the nominal system's state-matrix [see the representation of (9)] is shown in Fig. 4. Clearly, as the permitted Lipschitz bound L^k of the perturbation controller and the level δ^k of the safe domain increase over the successive iterations, (K^k, P^k) get adjusted so that the eigenvalues are placed further away from the imaginary axis toward the left of the complex plane, thus securing higher "margin of stability" to allow larger NPV ζ_{K^k}.

This part of the algorithm is implemented in Python 3.7, and (23)–(26) are solved using CVXPY 1.2 with MOSEK 9.2.47 as the backend solver.

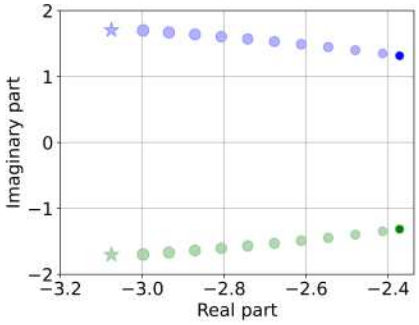

Fig. 4. Iterative progression of the eigenvalues of A_{0,K^k}. Bubbles of the two colors denote the respective two eigenvalues, whose sizes increase with the iterations; their final values are shown by the stars of the respective colors.

B. Computing NN Controller $\pi_{\rho^*}(\cdot)$

The NN controller is trained using Algorithm 2. Both the controller and value NNs, i.e., $\pi_\rho(\cdot) : \mathbb{R}^2 \to \mathbb{R}^2$ and $\hat{v}_\phi(\cdot) : \mathbb{R}^2 \to \mathbb{R}$, respectively, have two trainable layers. The hidden layer of each NN has five neurons, each with "tanh(·)" activation, and the activation of the single neuron of the output layer is the identity function. The "on-policy" gradient decent for both the NNs is performed using Adam [50] with step-size of $\alpha_\rho = \alpha_\phi = 0.001$. The other parameters of the algorithm are set as follows: $\beta = 10^{-15}$, $n_s = 200$, $n_a = 20$, $\tau = 0.1$, $n_t = 600$, $\nu_d = 0.98$, $\nu_{\min} = 10^{-4}$, and each diagonal element in the diagonal covariance matrix Σ is set to $(0.15)^2 = 0.0225$. The trainable bias of the two layers for both $\pi_\rho(\cdot)$ and $\hat{v}_\phi(\cdot)$ are set to zero; this ensures $\pi_\rho(0) = 0$. Upon termination of Algorithm 2, the trained weight matrices of the respective layers of the optimal NN controller $\pi_{\rho^*}(\cdot)$ are found to be

$$W_1 = \begin{bmatrix} -0.0503 & -0.4911 & 0.4001 & -0.2690 & 0.0077 \\ -0.3338 & -0.2768 & 0.0496 & 0.3172 & -0.1867 \end{bmatrix}^{\mathrm{T}}$$
$$W_2 = \begin{bmatrix} 0.0119 & 0.0393 & -0.3223 & -0.2757 & -0.1733 \\ 0.1496 & 0.2292 & 0.1309 & 0.2942 & 0.2662 \end{bmatrix}.$$

The Lipschitz bound of $\pi_{\rho^*}(\cdot)$ computed using the method proposed in [48] is: $L_{\pi_{\rho^*}} = 0.8218$, which is well below the value $L^* = 0.99$ in (31). Hence, by Theorem 1, the controller $\pi^* = \pi_{K^*} + \pi_{\rho^*}$ is Ω-stabilizing for system (29) with $\mathcal{E}_{P^*,\sigma^*}$ as an inner estimate of the maximal RSIS.

This part of the algorithm is implemented in Python 3.7, and the architecture and backpropagation of the controller and value NNs are implemented using Tensorflow 2.3.

C. Performance Evaluation of Trained Controller

An instance of transient performance of the trained NN-based controller $\pi^*(x)$ is depicted in Fig. 5, where the parameters of the system are held fixed at $\omega_1 \equiv \theta_1 = -0.0253$, $\omega_2 \equiv \theta_2 = 0.0532$, and the system is initialized at $x_1(0) = 0.2752$ and $x_2(0) = 0.1866$. The response of the system under the above-computed controller $u = \pi^*(x)$ is plotted.

For a comparative validation of the performance of the proposed controller, we pick as benchmark the linear quadratic regulator (LQR) designed for the linear nominal system. We compute the LQR gain for (A_0, B_0) and the given reward function solving the algebraic Ricatti equation

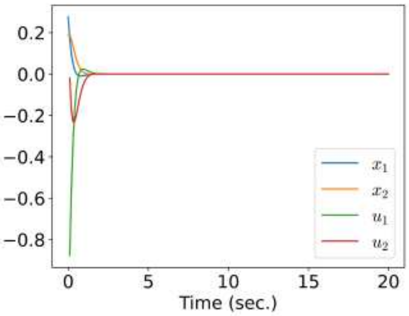

Fig. 5. System's transient response under $\pi^*(x)$, where the parameter value is: $\theta = [-0.0253, 0.0532]^{\mathrm{T}}$, and the initialization is at: $x(0) = [0.2752, 0.1866]^{\mathrm{T}}$.

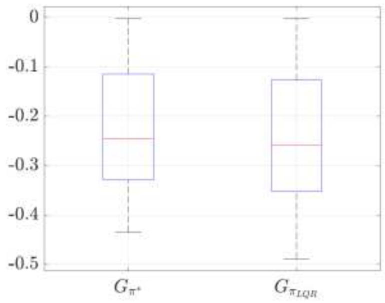

Fig. 6. Box-whisker plots of G_u, for 40 simulations with $\omega(k.\tau)$, $x(0)$ chosen randomly, under $u = \pi_{\mathrm{LQR}}(x) := K_{\mathrm{LQR}}.x$ and under $u = \pi^*(x)$, respectively.

using MATLAB R2020b

$$K_{\mathrm{LQR}} = \begin{bmatrix} -0.8350 & 0.1414 \\ 0.1414 & -0.5043 \end{bmatrix}$$

and set $K = K_{\mathrm{LQR}}$, $u_\rho = 0$ in the equivalent representation of (9). While LQR can guarantee optimality and stability for the linear nominal dynamics whenever that is stabilizable and gets to be widely used even for the nonlinear systems, obtained against their local linearized models [51], [52], [53]; yet, in general, an estimate for the corresponding RoA is not available in the presence of plant nonlinearity and/or parametric variation. Additionally, LQR cannot guarantee the boundedness of the system's trajectory within $\mathcal{X}$ either.

Next, we simulated 40 trajectories of the system's response, each with 200 discrete time steps at a sampling interval of $\tau = 0.1$ sec., where $\{\omega(k.\tau)|k \in 0, \dots, n_s\}$ and $x(0)$ for each simulation were chosen uniformly randomly from their respective domains: Θ^{n_s+1} and $\mathcal{E}_{P^*,\sigma^*}$. For each selected $\{\omega(k.\tau)\}$ and $x(0)$, the system responses under LQR and also under the controller $\pi^*(\cdot)$ were simulated, and their utilities were computed using (6). The statistics of the utilities over these 40 simulations are shown in Fig. 6, where the median value of the utility slightly improved by 4.92% under $\pi^*(\cdot)$ compared to that under LQR. Also, using our approach, the RSIS $\mathcal{E}_{P^*,\sigma^*}$ could also be computed as shown in Fig. 3, but that is not known for a typical LQR. In addition, note LQR computation is feasible only when $r(\cdot,\cdot)$ is quadratic, as chosen in this example, whereas our Algorithm 2 does not have such restriction.

Remark 4: Our proposed method can be applied to NN-based state-feedback controller synthesis for safe, stable, and optimal regulation of any nonlinear plant at a given setpoint, whenever the CT physical plant model satisfies Assumption 1, and a polytopic set of parametric variation, a polytopic safe operating domain, and a utility function for regulation are specified. One such real-world use-case can be the way-point tracking of quadrotors in the presence of variations in aerodynamic thrust, drag, and/or payload as commonly encountered in practice. Our proposed method can be used to design an NN-based locally optimal controller that guarantees robust stability, regulation to the given way-point, and boundedness of the trajectories upon parametric variations. The existing RL-based NN-controls cannot guarantee these properties [54].

VI. Conclusion

The presented framework provides a way to design and certify NN controllers for nonlinear systems subject to parameter variations for safety, stability, and robustness. Its a first framework for designing safe, stabilizing, and robust NN-based state-feedback controllers for nonlinear CT systems, where the dynamic model is known but is subject to unknown parametric variation over a given bounded set. A stability certificate is introduced extending the existing Lyapunov-based results and is further used to compute a maximal Lipschitz bound for a stabilizing NN-based controller, together with a corresponding maximal RoA contained in a user-given safe operating domain, starting from where the asymptotic closed-loop stability of the system is guaranteed regardless of arbitrary parametric variation, and at the same time the state trajectory remains confined to the safe domain. A SGT algorithm is also presented to design such a safe and robustly stabilizing NN controller that also maximizes the system's expected long-run utility, with respect to random initializations and parametric variations. The illustrative example validates the correctness of the proposed theory and the effectiveness of the proposed algorithms. Future work can generalize the proposed framework for the case of partial observability and output trajectory tracking.

Appendix A
Proof of Proposition 1

It follows from (4) that a controller $\pi_\rho \in \Pi_L$ satisfies the following $\forall\ x_1, x_2 \in \mathbb{R}^n$:

$$\begin{aligned}\left\|\pi_\rho(x_1) - \pi_\rho(x_2)\right\|_\infty &\le L\|x_1 - x_2\|_\infty \\ &\le L\sum_{j=1}^{n}\left|x_1^j - x_2^j\right|. \end{aligned} \tag{32}$$

From the above, it further follows that there exists a set of functions: $\delta_{ij} : \mathbb{R}^n \times \mathbb{R}^n \to [-L, L]\ \forall\ i \in \{1, \dots, m\}\ \forall\ j \in \{1, \dots, n\}$ such that $\forall\ x_1, x_2 \in \mathbb{R}^n$

$$\pi_\rho(x_1) - \pi_\rho(x_2) = \begin{bmatrix} \sum_{j=1}^{n} \delta_{1j}(x_1, x_2).\left(x_1^j - x_2^j\right) \\ \vdots \\ \sum_{j=1}^{n} \delta_{mj}(x_1, x_2).\left(x_1^j - x_2^j\right) \end{bmatrix}. \tag{33}$$

Also, since $\pi_\rho(0) = 0$, we get the following by setting $x_1 = x$ and $x_2 = 0$ in (33):

$$\pi_\rho(x) = \begin{bmatrix} \sum_{j=1}^{n} \delta_{1j}(x, 0).x^j \\ \vdots \\ \sum_{j=1}^{n} \delta_{mj}(x, 0).x^j \end{bmatrix} = \left[\mathbf{I}_m \odot \mathbf{1}_{1\times n}\right].\chi(x) \tag{34}$$

where for $k := i + (j-1)m \in \{1, \dots, mn\}$, the kth element of $\chi(x)$ is defined as $\chi^k \equiv \chi^{i+(j-1)m} := \delta_{ij}(x, 0).x^j$. Note this implies $\chi(0) = 0$, also since $\delta_{ij}(x, 0)^2 \le L^2$, we get

$$\begin{aligned} &\left(\chi^{i+(j-1)m}\right)^2 \le L^2\left(x^j\right)^2 \\ &\quad \Rightarrow \sum_{i,j} \gamma_{i,j} L^2\left(x^j\right)^2 - \sum_{i,j} \gamma_{i,j}\left(\chi^{i+(j-1)m}\right)^2 \ge 0 \ \forall\ \gamma_{i,j} \ge 0 \\ &\quad \Rightarrow \begin{bmatrix} x \\ \chi \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} L^2\mathrm{diag}\left(\Gamma_j\right) & \mathbf{0}_{n\times mn} \\ * & \mathrm{diag}\left(\left\{-\gamma_{i,j}\right\}\right) \end{bmatrix} [*] \ge 0. \end{aligned}$$

Appendix B
Proof of Proposition 2

To simplify notation, let us denote the space $\mathcal{X} \times \mathcal{U}_{L,\mathcal{X}} \subset \mathbb{R}^{n+m}$ by $\mathcal{Z}$, where $\mathcal{U}_{L,\mathcal{X}}$ is the L-bounded control subspace of a controller $\pi_\rho \in \Pi_L$ over $\mathcal{X}$. Accordingly, $(x, u_\rho) \in \mathcal{X} \times \mathcal{U}_{L,\mathcal{X}}$ is equivalently written as $z \in \mathcal{Z}$, where $z := [x^{\mathrm{T}} \ \ u_\rho^{\mathrm{T}}]^{\mathrm{T}}$. Also, the NPV $\zeta_K(x, u_\rho, \theta)$ is simply denoted $\zeta_K(z, \theta)$. Then, $\forall\ i \in \{1, \dots, n\}$ $\forall\ z_1, z_2 \in \mathcal{Z}$, and for each $\theta \in \Theta$

$$\zeta_K^i(z_1, \theta) - \zeta_K^i(z_2, \theta) = \sum_{j=1}^{n+m} \left\{\zeta_K^i\left(z_{2,j}, \theta\right) - \zeta_K^i\left(z_{2,j-1}, \theta\right)\right\} \tag{35}$$

where $z_{2,0} := z_2$, and for $j > 0$, the kth element of $z_{2,j}$ is

$$z_{2,j}^k := \begin{cases} z_1^k, & k \le j \\ z_2^k, & k > j. \end{cases} \tag{36}$$

Note that in the jth term of the summation in (35), the vectors $z_{2,j}$, $z_{2,j-1} \in \mathbb{R}^{m+n}$ are componentwise identical except for their jth component. This implies: $z_{2,j} - z_{2,j-1} = (z_1^j - z_2^j).1_j$, where $1_j \in \mathbb{R}^{m+n}$ is a binary vector with only the jth entry 1 and other entries zero. Since $\zeta_K(\cdot, \cdot, \cdot)$ is locally componentwise $(\underline{\mathcal{L}}, \overline{\mathcal{L}})$-sector-bounded over $\mathcal{X}$, we have from (13) that $\forall\ i \in \{1, \dots, n\}$ and $\forall\ j \in \{1, \dots, n+m\}$

$$\underline{\mathcal{L}}^{i,j} \le J^{i,j}_{\zeta_K(z,\theta),z}\Big|_{\substack{z=\hat{z} \\ \theta\ =\hat{\theta}}} \le \overline{\mathcal{L}}^{i,j} \ \forall\ \hat{z} \in \mathcal{Z}\ \forall\ \hat{\theta} \in \Theta. \tag{37}$$

It then follows that $\forall\ i \in \{1, \dots, n\}$ $\forall\ j \in \{1, \dots, n+m\}$ $\forall\ \theta \in \Theta$, and for $z_{2,j}, z_{2,j-1}$ as defined in (35)

$$\underline{\mathcal{L}}^{i,j}\left(z_1^j - z_2^j\right) \le \zeta_K^i\left(z_{2,j}, \theta\right) - \zeta_K^i\left(z_{2,j-1}, \theta\right) \le \overline{\mathcal{L}}^{i,j}\left(z_1^j - z_2^j\right). \tag{38}$$

Combining (35) and (38), we obtain $\forall\ \theta \in \Theta$

$$\begin{aligned} \sum_{j=1}^{n+m} \underline{\mathcal{L}}^{i,j}\left(z_1^j - z_2^j\right) &\le \zeta_K^i(z_1, \theta) - \zeta_K^i(z_2, \theta) \\ &\le \sum_{j=1}^{n+m} \overline{\mathcal{L}}^{i,j}\left(z_1^j - z_2^j\right). \end{aligned} \tag{39}$$

This implies that for each $\theta \in \Theta$, there exists a set of functions: $\delta_\theta^{ij}: \mathcal{Z} \times \mathcal{Z} \to [\underline{\mathcal{L}}^{i,j}, \overline{\mathcal{L}}^{i,j}]$ $\forall\ i \in \{1, \dots, n\}$ $\forall\ j \in \{1, \dots, n+m\}$ such that $\forall\ z_1, z_2 \in \mathcal{Z}$

$$\zeta_K(z_1, \theta) - \zeta_K(z_2, \theta) = \begin{bmatrix} \sum_{j=1}^{n+m} \delta_\theta^{1j}(z_1, z_2).\left(z_1^j - z_2^j\right) \\ \vdots \\ \sum_{j=1}^{n+m} \delta_\theta^{nj}(z_1, z_2).\left(z_1^j - z_2^j\right) \end{bmatrix}. \tag{40}$$

Also, since $\zeta_K(0, \theta) = 0\ \forall\ \theta \in \Theta$, we get the following for each $\theta \in \Theta$ by setting $z_1 = z$ and $z_2 = 0$ in (40):

$$\zeta_K(z, \theta) = \begin{bmatrix} \sum_{j=1}^{n+m} \delta_\theta^{1j}(z, 0).z^j \\ \vdots \\ \sum_{j=1}^{n+m} \delta_\theta^{nj}(z, 0).z^j \end{bmatrix} = \left[\mathbf{I}_n \odot \mathbf{1}_{1\times(n+m)}\right].\xi_\theta(z) \tag{41}$$

where for $k := i + (j-1)n \in \{1, \dots, n(m+n)\}$, the kth element of $\xi_\theta(z)$ is defined as, $\xi_\theta^k \equiv \xi_\theta^{i+(j-1)n} := \delta_\theta^{ij}(z, 0).z^j$.

From the definition of $\overline{c}_{i,j}$ and $c_{i,j}$, it follows that $\forall\ i \in \{1, \dots, n\}$ and $\forall\ j \in \{1, \dots, n+m\}$:

$$\begin{aligned} &|\overline{c}_{ij}| \ge |\delta_\theta^{ij}(z, 0) - c_{ij}| \\ &\quad \Leftrightarrow \overline{c}_{ij}^2\left(z^j\right)^2 \ge \left(\delta_\theta^{ij}(z, 0)z^j - c_{ij}z^j\right)^2 \\ &\quad \Leftrightarrow \left(\overline{c}_{ij}^2 - c_{ij}^2\right).\left(z^j\right)^2 + 2c_{ij}.z^j.\xi_\theta^{i+(j-1)n} - \left(\xi_\theta^{i+(j-1)n}\right)^2 \ge 0. \end{aligned} \tag{42}$$

Equation (42) further implies that $\forall\ \Lambda > 0, k_{ij} := i + (j-1)n$

$$\begin{aligned} \sum_{\substack{i\in\{1,\dots,n\} \\ j\in\{1,\dots,n+m\}}} \Lambda^{k_{ij}} \Big\{ &\left(\overline{c}_{ij}^2 - c_{ij}^2\right).\left(z^j\right)^2 \\ &+ \ \ 2c_{ij}.z^j.\xi_\theta^{k_{ij}} - \left(\xi_\theta^{k_{ij}}\right)^2 \Big\} \ge 0. \end{aligned} \tag{43}$$

Next, we get the following for each $\theta \in \Theta$, by writing (43) in matrix form, splitting variable z into x and u_ρ, and recognizing that $u_\rho = \pi_\rho(x) = Q.\chi(x)$ with $\pi_\rho(\cdot) \in \Pi_L$, for which $x \in \mathcal{X} \Rightarrow u_\rho \in \mathcal{U}_{L,\mathcal{X}}$

$$\begin{bmatrix} x \\ \chi \\ \xi_\theta \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} M_{x\Lambda} & \mathbf{0}_{n\times m.n} & N_{x\Lambda} \\ * & M_{\chi\Lambda} & N_{\chi\Lambda} \\ * & * & M_{\xi\Lambda} \end{bmatrix} [*] \ge 0\ \forall\ x \in \mathcal{X} \tag{44}$$

where $M_{x\Lambda}$, $M_{q\Lambda}$, $M_{\xi\Lambda}$, $N_{x\Lambda}$, and $N_{q\Lambda}$ are as defined in (18).

We note that the above proof is partially inspired from the proof of [28, Lemma 4.2].

Appendix C
Proof of Theorem 1

In the given setting, i.e., given $L \in \mathbb{R}_{\ge 0}$, $\mathcal{X} \subset \mathbb{R}^n$, and the system (1) under control of $\pi(x) = \pi_K(x) + \pi_\rho(x)$ satisfying Assumption 1, assume that there exist $K \in \mathbb{R}^{m\times n}$, $P \succcurlyeq 0$, $\Lambda \ge 0$, and $\gamma_{i,j} \ge 0$ for all $i \in 1, \dots, m, j \in 1, \dots, n$ satisfying (20), or equivalently, except at the origin the following holds:

$$\begin{bmatrix} x \\ \chi \\ \xi_\theta \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} V_{L,\{\Gamma_j,P,K\}} & * & * \\ \mathbf{0}_{m.n\times n} & M_{\chi\Lambda} & * \\ N_{x\Lambda}^{\mathrm{T}} + R^{\mathrm{T}}.P & N_{\chi\Lambda}^{\mathrm{T}} & M_{\xi\Lambda} \end{bmatrix} [*] < 0. \tag{45}$$

Also, owing to the local $(\underline{\mathcal{L}}, \overline{\mathcal{L}})$-sector bound of the NPV of the equivalent system (9), we can combine (9) and Proposition 2 to get the following under a controller $u_\rho = \pi_\rho(x)$, uniformly $\forall\, x \in \mathcal{X}, \theta \in \Theta, \pi_\rho(\cdot) \in \Pi_L$

$$\dot{x} = f\big(x, u_\rho, \theta\big) \equiv f_\theta\big(x, u_\rho\big) = A_{0,K}.x + \underbrace{R.\xi_\theta(x)}_{=\zeta_K(x,u_\rho,\theta)} . \tag{46}$$

Accordingly, by algebraic manipulation it follows that in the given setting, (45) is equivalent to the following, uniformly $\forall\, x \in \mathcal{X}, \theta \in \Theta, \pi_\rho(\cdot) \in \Pi_L$

$$\left\{x^{\mathrm{T}} P f_\theta + f_\theta^{\mathrm{T}} P x\right\} + \left\{\begin{bmatrix} x \\ \chi \\ \xi_\theta \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} M_{x\Lambda} & \mathbf{0}_{n\times m.n} & N_{x\Lambda} \\ * & M_{\chi\Lambda} & N_{\chi\Lambda} \\ * & * & M_{\xi\Lambda} \end{bmatrix} [*]\right\}$$
$$+ \left\{\begin{bmatrix} x \\ \chi \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} L^2 \mathrm{diag}\big(\{\Lambda_j\}\big) & \mathbf{0}_{n\times mn} \\ * & \mathrm{diag}\big(\{-\lambda_{i,j}\}\big) \end{bmatrix} [*]\right\} < 0. \tag{47}$$

From Proposition 2, the local $(\underline{\mathcal{L}}, \overline{\mathcal{L}})$-sector bound of the NPV of system (9) also implies that uniformly $\forall\, x \in \mathcal{X}, \theta \in \Theta, \pi_\rho(\cdot) \in \Pi_L$, we have the second term of (47) non-negative. Moreover from Proposition 1, $\pi_\rho(\cdot) \in \Pi_L$ implies that the third term is non-negative. Hence, in the given setting, uniformly $\forall\, \theta \in \Theta, \pi_\rho(\cdot) \in \Pi_L$, (47) is equivalent to

$$\begin{aligned} & x^{\mathrm{T}}.P.f_\theta + f_\theta^{\mathrm{T}}.P.x < 0 \ \forall\, x \in \mathcal{X}\backslash\{0\} \\ & \Leftrightarrow \dot{V}(x) < 0 \ \forall\, x \in \mathcal{X}\backslash\{0\} \end{aligned} \tag{48}$$

where $V(x) = x^{\mathrm{T}}.P.x$. It can be seen that $V(\cdot)$ is continuously differentiable and satisfies the conditions in (19) over $\mathcal{X}$, regardless of how θ evolves over time. Hence, $V(x)$ is a CLF for (9), and equivalently, also for system (1) under controller $\pi(x) = \pi_K(x) + \pi_\rho(x)$, which implies that in the given setting, system (1) is Ω-stable, uniformly for $\pi_\rho(\cdot) \in \Pi_L$.

If the value of either of K and P is given, then note (47) serves as a variant of "S-procedure" [42, pp. 23–24] used in various control applications to formulate conservative LMI relaxations for solving sets of indefinite QCs [28], [29], [30].

Appendix D
Proof of Corollary 1

Since the safe domain $\mathcal{X} \subset \mathbb{R}^n$ is a neighborhood of the origin, there exists a $\sigma \in \mathbb{R}_{>0}$ s.t. the set $\mathcal{E}_{P,\sigma}$, which is a hyperellipse since $P \succ 0$, is contained within $\mathcal{X}$.

In the given setting, i.e., given $L \in \mathbb{R}_{\geq 0}$, $\mathcal{X} \subset \mathbb{R}^n$, and the system (1) under control of $\pi(x) = \pi_K(x) + \pi_\rho(x)$ satisfying Assumption 1, say P satisfies (20) for a certain $K \in \mathbb{R}^{m\times n}$. Then, following Theorem 1, since $V(x) = x^{\mathrm{T}}.P.x$ is a CLF of system (1) locally over $\mathcal{X}$, we have $\dot{V}(x) < 0 \ \forall\, x \in \mathcal{X}\backslash\{0\}$ uniformly for any $\pi_\rho \in \Pi_L$. Also since $V(x) = \sigma$ uniformly over the boundary of $\mathcal{E}_{P,\sigma} \subset \mathcal{X}$, $\mathcal{E}_{P,\sigma}$ is an invariant set, i.e.,

$$x \in \mathcal{E}_{P,\sigma} \Rightarrow \psi_\pi\big(\omega^t, x\big) \in \mathcal{E}_{P,\sigma} \ \forall\, t \in \mathbb{R}_{\geq 0}. \tag{49}$$

Hence, uniformly for each $x \in \mathcal{E}_{P,\sigma}\backslash\{0\}$, we have

$$\left\|P^{\frac{1}{2}}.\psi_\pi\Big(\omega^{t'}, x\Big)\right\|_2 < \left\|P^{\frac{1}{2}}.\psi_\pi\big(\omega^t, x\big)\right\|_2 \ \forall\, t' > t. \tag{50}$$

In other words, $\psi_\pi(\omega^t, x)$ quadratically converges to the origin as $t \to \infty \ \forall\, \omega \in \Omega$ if the system is initialized within $\mathcal{E}_{P,\sigma}$. Hence, $\mathcal{E}_{P,\sigma}$ is a Ω-RoA of the system (1) under a controller $\pi(x) = \pi_K(x) + \pi_\rho(x)$, uniformly for any $\pi_\rho \in \Pi_L$. Using $\mathcal{E}_{P,\sigma} \subset \mathcal{X}$, it further follows that $\mathcal{E}_{P,\sigma}$ is an RSIS under a controller $\pi(x) = \pi_K(x) + \pi_\rho(x)$, uniformly for any $\pi_\rho \in \Pi_L$.

References

[1] A. Venkatraman, M. Hebert, and J. A. Bagnell, "Improving multi-step prediction of learned time series models," in *Proc. AAAI Conf. Artif. Intell.*, vol. 29, 2015, pp. 3024–3030.

[2] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philos. Trans. Roy. Soc. London B, Biol. Sci.*, vol. 358, no. 1431, pp. 537–547, 2003.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[4] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[5] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.

[6] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[7] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[8] K. Choromanski, M. Rowland, V. Sindhwani, R. Turner, and A. Weller, "Structured evolution with compact architectures for scalable policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 970–978.

[9] I. Clavera, J. Rothfuss, J. Schulman, Y. Fujita, T. Asfour, and P. Abbeel, "Model-based reinforcement learning via meta-policy optimization," in *Proc. Conf. Robot Learn.*, 2018, pp. 617–629.

[10] T. Kurutach, I. Clavera, Y. Duan, A. Tamar, and P. Abbeel, "Model-ensemble trust-region policy optimization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15. [Online]. Available: https://openreview.net/forum?id=SJJinbWRZ

[11] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[12] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electron. Imag.*, vol. 2017, no. 19, pp. 70–76, 2017.

[13] Q. Huang, R. Huang, W. Hao, J. Tan, R. Fan, and Z. Huang, "Adaptive power system emergency control using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1171–1182, Mar. 2020.

[14] C. Wang, J. Wang, Y. Shen, and X. Zhang, "Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2124–2136, Mar. 2019.

[15] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, "Explainability in deep reinforcement learning," *Knowl.-Based Syst.*, vol. 214, Feb. 2021, Art. no. 106685.

[16] S. Talukder, M. Ibrahim, and R. Kumar, "Resilience indices for power/cyberphysical systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 4, pp. 2159–2172, Apr. 2021.

[17] S. Talukder and R. Kumar, "An enhancement in sum-of-squares optimization based region of attraction estimation for power systems," in *Proc. IEEE Power Energy Soc. General Meeting (PESGM)*, 2019, pp. 1–5.

[18] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[19] S.-M. Lu, D.-P. Li, and Y.-J. Liu, "Adaptive neural network control for uncertain time-varying state constrained robotics systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 12, pp. 2511–2518, Dec. 2019.

[20] R. Padhi, N. Unnikrishnan, X. Wang, and S. N. Balakrishnan, "A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems," *Neural Netw.*, vol. 19, no. 10, pp. 1648–1660, 2006.

[21] H. Ma, Z. Wang, D. Wang, D. Liu, P. Yan, and Q. Wei, "Neural-network-based distributed adaptive robust control for a class of nonlinear multiagent systems with time delays and external noises," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 6, pp. 750–758, Jun. 2016.

[22] H. Jiang, H. Zhang, Y. Luo, and J. Han, "Neural-network-based robust control schemes for nonlinear multiplayer systems with uncertainties via adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 3, pp. 579–588, Mar. 2019.

[23] F. Cheng, H. Wang, L. Zhang, A. M. Ahmad, and N. Xu, "Decentralized adaptive neural two-bit-triggered control for nonstrict-feedback nonlinear systems with actuator failures," *Neurocomputing*, vol. 500, pp. 856–867, Aug. 2022.

[24] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.

[25] D. Liu, X. Yang, D. Wang, and Q. Wei, "Reinforcement-learning-based robust controller design for continuous-time uncertain nonlinear systems subject to input constraints," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1372–1385, Jul. 2015.

[26] C. Mu, D. Wang, and H. He, "Novel iterative neural dynamic programming for data-based approximate optimal control design," *Automatica*, vol. 81, pp. 240–252, Jul. 2017.

[27] Y. Zhou and X.-H. Chang, "Event-triggered quantized $L_2 - L_\infty$ filtering for neural networks under denial-of-service attacks," *Int. J. Robust Nonlinear Control*, vol. 32, no. 10, pp. 5897–5918, 2022.

[28] M. Jin and J. Lavaei, "Control-theoretic analysis of smoothness for stability-certified reinforcement learning," in *Proc. IEEE Conf. Decis. Control (CDC)*, 2018, pp. 6840–6847.

[29] H. Yin, P. Seiler, and M. Arcak, "Stability analysis using quadratic constraints for systems with neural network controllers," *IEEE Trans. Autom. Control*, vol. 67, no. 4, pp. 1980–1987, Apr. 2022.

[30] H. Yin, P. Seiler, M. Jin, and M. Arcak, "Imitation learning with stability and safety guarantees," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 409–414, 2022.

[31] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE Trans. Autom. Control*, vol. 42, no. 6, pp. 819–830, Jun. 1997.

[32] S. Chen, M. Fazlyab, M. Morari, G. J. Pappas, and V. M. Preciado, "Learning Lyapunov functions for piecewise affine systems with neural network controllers," 2020, *arXiv:2008.06546*.

[33] H. Dai, B. Landry, L. Yang, M. Pavone, and R. Tedrake, "Lyapunov-stable neural-network control," 2021, *arXiv:2109.14152*.

[34] A. Aydinoglu, M. Fazlyab, M. Morari, and M. Posa, "Stability analysis of complementarity systems with neural network controllers," 2020, *arXiv:2011.07626*.

[35] M. Han, L. Zhang, J. Wang, and W. Pan, "Actor-critic reinforcement learning for control with stability guarantee," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6217–6224, Oct. 2020.

[36] L. Zhang, R. Zhang, T. Wu, R. Weng, M. Han, and Y. Zhao, "Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5435–5444, Dec. 2021.

[37] T. L. Vu, S. Mukherjee, T. Yin, R. Huang, J. Tan, and Q. Huang, "Safe reinforcement learning for emergency load shedding of power systems," in *Proc. IEEE Power Energy Soc. General Meeting (PESGM)*, 2021, pp. 1–5.

[38] H. K. Khalil, *Nonlinear Control*. New York, NY, USA: Pearson, 2015.

[39] S. Gao, S. Kong, and E. M. Clarke, "dReal: An SMT solver for nonlinear theories over the reals," in *Automated Deduction (CADE-24)*, M. P. Bonacina, Ed. Berlin, Germany: Springer, 2013, pp. 208–214.

[40] L. Vu and D. Liberzon, "Common Lyapunov functions for families of commuting nonlinear systems," *Syst. Control Lett.*, vol. 54, no. 5, pp. 405–416, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167691104001598

[41] D. Liberzon, *Switching in Systems and Control*, vol. 190. Boston, MA, USA: Springer, 2003.

[42] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA, USA: SIAM, 1994.

[43] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[44] A. Agrawal and S. Boyd, "Disciplined quasiconvex programming," *Optim. Lett.*, vol. 14, no. 7, pp. 1643–1657, 2020.

[45] F. A. Potra and S. J. Wright, "Interior-point methods," *J. Comput. Appl. Math.*, vol. 124, nos. 1–2, pp. 281–302, 2000.

[46] Z. Zhang, J. Chen, Z. Chen, and W. Li, "Asynchronous episodic deep deterministic policy gradient: Toward continuous control in computationally complex environments," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 604–613, Feb. 2021.

[47] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, 3rd Quart., 2009.

[48] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, "Regularisation of neural networks by enforcing Lipschitz continuity," *Mach. Learn.*, vol. 110, no. 2, pp. 393–416, 2021.

[49] H. Yin, P. Seiler, and M. Arcak, "Stability analysis using quadratic constraints for systems with neural network controllers," 2020, *arXiv:2006.07579*.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[51] E. Okyere, A. Bousbaine, G. T. Poyi, A. K. Joseph, and J. M. Andrade, "LQR controller design for quad-rotor helicopters," *J. Eng.*, vol. 2019, no. 17, pp. 4003–4007, 2019.

[52] L. Chrif and Z. M. Kadda, "Aircraft control system using LQG and LQR controller with optimal estimation-Kalman filter design," *Procedia Eng.*, vol. 80, pp. 245–257, Oct. 2014.

[53] K. D. Rao and S. Kumar, "Modeling and simulation of quarter car semi active suspension system using LQR controller," in *Proc. 3rd Int. Conf. Front. Intell. Comput. Theory Appl. (FICTA)*, 2015, pp. 441–448.

[54] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2096–2103, Oct. 2017.

Soumyabrata Talukder (Student Member, IEEE) received the bachelor's degree in electrical engineering from Jadavpur University, Kolkata, India, in 2009. He is currently pursuing the Ph.D. degree in electrical engineering with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA.

He is also serving as a Specialist Engineer with Eaton Research Labs, Dublin, Ireland. He served as a Project Engineer with GE Grid Solutions (formerly, Alstom Grid), Bengaluru, India, from 2009 to 2012, and thereafter, as an Assistant Manager with Central Projects Organization, ITC Ltd., Kolkata, India, from 2012 to 2017. His current research interests include control theory, resilience of cyber–physical systems, polynomial optimization, machine learning, and coalition games.

Ratnesh Kumar received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Kanpur, Kanpur, India, in 1987, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 1989 and 1991, respectively.

He is a Palmer Professor with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA, where he directs the ESSeNCE (Embedded Software, Sensors, Networks, Cyberphysical, and Energy) Lab. He previously held faculty position with the University of Kentucky, Lexington, KY, USA, and various visiting positions with the University of Maryland at College Park, College Park, MD, USA; the Applied Research Laboratory, Pennsylvania State University, State College, PA, USA; the NASA Ames, Moffett Field, CA, USA; Idaho National Laboratory, Idaho Falls, ID, USA; the United Technologies Research Center, East Hartford, CT, USA; and Air Force Research Laboratory, Wright-Patterson AFB, OH, USA.

Dr. Kumar is a recipient of the *D. R. Boylan Eminent Faculty Award for Research* and the *Award for Outstanding Achievement in Research* from Iowa State University, and also the *Distinguished Alumni Award* from IIT Kanpur. He received Gold Medals for the Best EE Undergrad, the Best All Rounder, and the Best EE Project from IIT Kanpur, and the Best Dissertation Award from UT Austin, the Best Paper Award from the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and has been Keynote Speaker and a paper award recipient from multiple conferences. He is or has been an Editor of several journals, including IEEE, SIAM, ACM, Springer, IET, and MDPI. He was a Distinguished Lecturer of the IEEE Control Systems Society. He is a Fellow of AAAS.