arXiv:1810.11505v1 [cs.SY] 26 Oct 2018

STABILITY-CERTIFIED REINFORCEMENT LEARNING: A CONTROL-THEORETIC PERSPECTIVE[*]

MING JIN[†] AND JAVAD LAVAEI[‡]

Abstract. We investigate the important problem of certifying stability of reinforcement learning policies when interconnected with nonlinear dynamical systems. We show that by regulating the input-output gradients of policies, strong guarantees of robust stability can be obtained based on a proposed semidefinite programming feasibility problem. The method is able to certify a large set of stabilizing controllers by exploiting problem-specific structures; furthermore, we analyze and establish its (non)conservatism. Empirical evaluations on two decentralized control tasks, namely multi-flight formation and power system frequency regulation, demonstrate that the reinforcement learning agents can have high performance within the stability-certified parameter space, and also exhibit stable learning behaviors in the long run.

Key words. Reinforcement learning, robust control, policy gradient optimization, decentralized control synthesis, safe reinforcement learning

AMS subject classifications. 68T05, 93E35, 93D09

1. Introduction.

Remarkable progress has been made in reinforcement learning (RL) using (deep) neural networks to solve complex decision-making and control problems [43]. While RL algorithms, such as policy gradient [52, 26, 41], Q-learning [49, 35], and actor-critic methods [32, 34] aim at optimizing control performance, the security aspect is of great importance for mission-critical systems, such as autonomous cars and power grids [20, 4, 44]. A fundamental problem is to analyze or certify stability of the interconnected system in both RL exploration and deployment stages, which is challenging due to its dynamic and nonconvex nature [20].

The problem under study focuses on a general continuous-time dynamical system:

$$\dot{x}(t) = f_t(x(t), u(t)), \tag{1.1}$$

with the state $x(t) \in \mathbb{R}^{n_s}$ and the control action $u(t) \in \mathbb{R}^{n_a}$. In general, f_t can be a time-varying and nonlinear function, but for the purpose of stability analysis, we study the important case that

$$f_t(x(t)) = Ax(t) + Bu(t) + g_t(x(t)), \tag{1.2}$$

where f_t comprises of a linear time-invariant (LTI) component $A \in \mathbb{R}^{n_s \times n_s}$ that is Hurwitz (i.e., every eigenvalue of A has strictly negative real part), a control matrix $B \in \mathbb{R}^{n_s \times n_a}$, and a slowly time-varying component g_t that is allowed to be nonlinear and even uncertain.[1] The condition that A is stable is a basic requirement, but the goal of reinforcement learning is to design a controller that optimizes some performance metric that is not necessarily related to the stability condition. For feedback control, we also allow the controller to obtain observations $y(t) = Cx(t) \in \mathbb{R}^{n_s}$ that are a linear function of the states, where $C \in \mathbb{R}^{n_s \times n_s}$ may have a sparsity pattern to account for partial observations in the context of decentralized controls [8].

[*] This work was supported by the ONR grants N00014-17-1-2933 and N00014-15-1-2835, DARPA grant D16AP00002, and AFOSR grant FA9550-17-1-0163.

[†] Department of Industrial Engineering and Operations Research, University of California, Berkeley. Email: jinming@berkeley.edu.

[‡] Department of Industrial Engineering and Operations Research, and the Tsinghua-Berkeley Shenzhen Institute, University of California, Berkeley. Email: lavaei@berkeley.edu.

[1] This requirement is not difficult to meet in practice, because one can linearize any nonlinear systems around the equilibrium point to obtain a linear component and a nonlinear part.

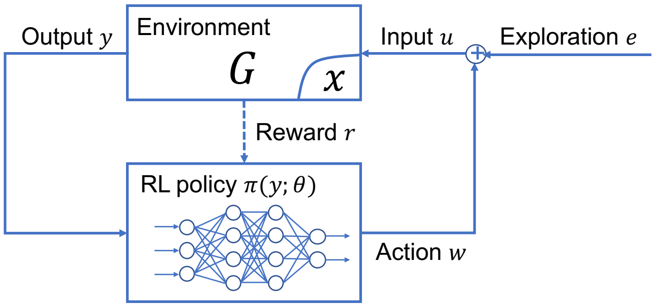


Fig. 1: Overview of the interconnected system of an RL policy and the environment. The goal of RL is to maximize expected rewards through interaction and exploration.

Suppose that $u(t) = \pi_t(y(t); \theta_t) + e(t)$ is a neural network given by an RL agent (parametrized by θ_t, which can be time-varying due to learning) to optimize some reward $r(x, u)$ revealed through the interaction with the environment. The exploration vector $e(t) \in \mathbb{R}^{n_a}$ captures the additive randomization effect during the learning phase, and is assumed to have a bounded energy over time ($\|e\|_2 = \sqrt{\int |e(t)|_2^2 dt} \leq \infty$). The main goal is to analyze the stability of the system with the actuation of π_t, which is typically a neural network controller, as illustrated in Figure 1. Specifically, the stability criterion is stated using the concept of L_2 gain [55, 16].[2]

Definition 1.1 (Input-output stability). *The L_2 gain of the system G controlled by π is the worst-case ratio between total output energy and total input energy:*

$$\gamma(G, \pi) = \sup_{u \in L_2} \frac{\|y\|_2}{\|u\|_2}, \tag{1.3}$$

where L_2 is the set of all square-summable signals, $\|y\|_2 = \sqrt{\int |y(t)|_2^2 dt}$ is the total energy over time, and $u(t) = \pi_t(y(t); \theta_t) + e(t)$ is the control input with exploration. If $\gamma(G, \pi)$ is finite, then the interconnected system is said to have input-output stability (or finite L_2 gain).

This study investigates the possibility of using the gradient information of the policy $\pi_t(y(t); \theta_t)$ to obtain a stability certificate, because this information can be easily extracted in real-time and is generic enough to include a large set of performance-optimizing nonlinear controllers. Let $[n] = \{1, ..., n\}$ be the set notation. By denoting

$$\mathcal{P}(\xi) = \left\{ \pi \;\middle|\; \underline{\xi}_{ij} \leq \partial_j \pi_i(y) \leq \overline{\xi}_{ij}, \forall i \in [n_a], j \in [n_s], y \in \mathbb{R}^{n_s} \right\} \tag{1.4}$$

as the set of controllers whose partial derivatives are bounded by $\underline{\xi} \in \mathbb{R}^{n_a \times n_s}$ and $\overline{\xi} \in \mathbb{R}^{n_a \times n_s}$, it is desirable to provide stability certificate as long as the RL policy remains within the above "safety set." Indeed, this can be checked efficiently, as stated (informally) in the following theorem.

[2] This stability metric is widely adopted in practice, and is closely related to bounded-input bounded-output (BIBO) stability and absolute stability (or asymptotic stability). For controllable and observable LTI systems, the equivalence can be established.

THEOREM 1.2 (Main result). *If there exist constants $\underline{\xi}$ and $\overline{\xi}$ such that the condition (3.21) is feasible for the system (1.1), then the interconnected system has a finite L_2 gain as long as $\pi_t \in \mathcal{P}(\underline{\xi}, \overline{\xi})$ for all $t \geq 0$.*

We call the constants $\underline{\xi}$ and $\overline{\xi}$ stability-certified gradient bounds for the underlying system. The above result is based on the intuition that a real-world stable controller should exhibit "smoothness" in the sense that small changes in the input should lead to small changes in the output. This incorporates the special case where controllers are known to have bounded Lipschitz constants (a simple strategy to calculate the Lipschitz constant of a deep neural network is suggested in [48]). To compute the gradient bounds, we borrow powerful ideas from the framework of integral quadratic constraint (in frequency domain) [33] and dissipativity theory (in time domain) [51] for robustness analysis. While these tools are celebrated with their non-conservatism in the robust control literature, existing characterizations of multi-input multi-output (MIMO) Lipschitz functions are insufficient. Thus, one major obstacle is to derive non-trivial bounds that could be of use in practice.

To this end, we develop a new quadratic constraint on gradient-bounded functions, which exploits the sparsity of the control architecture and the non-homogeneity of the output vector. Some key features of the stability-certified smoothness bounds are as follows: **(a)** the bounds are inherent to the targeted real-world control task; **(b)** they can be computed efficiently by solving some semi-definite programming (SDP) problem; **(c)** they can be used to certify stability when reinforcement learning is employed in real-world control with either off-policy or on-policy learning [47]. Furthermore, the stability certification can be regarded as an $\mathcal{S}$-procedure, and we analyze its conservatism to show that it is necessary for the robustness of a surrogate system that is closely related to the original system.

The paper is organized as follows. Preliminaries on policy gradient reinforcement learning, the integrated quadratic constraint (IQC) and dissipativity frameworks are presented in Section 2. Main results on gradient bounds for a linear or nonlinear system G are presented in Section 3, where we also analyze the conservatism of the certificate. The method is evaluated in Section 4 on two nonlinear decentralized control tasks. Conclusions are drawn in Section 5.

2. Preliminary.

In this section, we give an overview of the main topics relevant to this study, namely policy gradient reinforcement learning and robustness analysis based on IQC framework and dissipativity theory.

2.1. Reinforcement learning using policy gradient.

Reinforcement learning aims at guiding an agent to perform a task as efficiently and skillfully as possible through interactions with the environment. The control task is modeled as a Markov decision process (MDP), defined by the tuple $(\mathcal{X}, \mathcal{U}, \mathcal{T}, r, \rho)$, where $\mathcal{X}$ is the set of states x, $\mathcal{U}$ is a set of actions u, $\mathcal{T} : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ indicates the world dynamics as in (1.1), $r(x, u)$ is the reward at state x and action u, and $\rho \in (0, 1]$ is the factor to discount future rewards. A control strategy is defined by a policy $\pi_\theta(x)$, which can be approximated by a neural network with parameters θ. For a continuous control, the actions follow a multivariate normal distribution, where $\pi_\theta(x)$ is the mean, and the standard deviation in each action dimension is set to be a diminishing number during exploration or learning, and 0 during actual deployment. With a slight abuse of notations, we use $\pi_\theta(u|x)$ to denote this normal distribution over actions, and use

x_t to denote $x(t)$ for simplicity. The goal of RL is to maximize the expected return:

$$\eta(\pi_\theta) = \mathop{\mathbb{E}}_{x_0, u_t \sim \pi_\theta(\cdot|x_t), x_{t+1} \sim \mathcal{T}(x_t, u_t)} \left[\sum\nolimits_{t=0}^{T} \rho^t r(x_t, u_t) \right], \tag{2.1}$$

where T is the control horizon, and the expectation is taken over the policy, the initial state distribution and the world dynamics.

From a practitioner's point of view, the existing methods can be categorized into four groups based on how the optimal policy is determined: **(a)** policy gradient methods directly optimize the policy parameters θ by estimating the gradient of the expected return (e.g., REINFORCE [52], natural policy gradient [26], and trust region policy optimization (TRPO) [41]); **(b)** value-based algorithms like Q-learning do not aim at optimizing the policy directly, but instead approximate the Q-value of the optimal policy for the available actions [49, 35]; **(c)** actor-critic algorithms keep an estimate of the value function (critic) as well as a policy that maximizes the value function (actor) (e.g., DDPG [32] and A3C [34]); lastly, **(d)** model-based methods focus on the learning of the transition model for the underlying dynamics, and then use it for planning or to improve a policy (e.g., Dyna [46] and guided policy search [30]). We adopt an approach based on end-to-end policy gradient that combines TRPO [41] with natural gradient [26] and smoothness penalty (this method is very useful for RL in dynamical systems described by partial or difference equations).

Trust region policy optimization is a policy gradient method that constrains the step length to be within a "trust region" so that the local estimation of the gradient/curvature has a monotonic improvement guarantee. By manipulating the expected return $\eta(\pi)$ using the identity proposed in [25], the "surrogate objective" $L_{\pi_{\text{old}}}(\pi)$ can be designed:

$$L_{\pi_{\text{old}}}(\pi) = \mathop{\mathbb{E}}_{x,u \sim \pi_{\text{old}}} \left[\frac{\pi(u|x)}{\pi_{\text{old}}(u|x)} \Lambda^{\pi_{\text{old}}}(x,u) \right], \tag{2.2}$$

where the expectation is taken over the old policy π_{old}, the ratio inside the expectation is also known as the importance weight, and $\Lambda^{\pi_{\text{old}}}(x,u)$ is the advantage function given by:

$$\Lambda^{\pi_{\text{old}}}(x,u) = \mathop{\mathbb{E}}_{x' \sim \mathcal{T}(x,u)} \left[r(x,u) + \rho V^{\pi_{\text{old}}}(x') - V^{\pi_{\text{old}}}(x) \right], \tag{2.3}$$

where the expectation is with respect to the dynamics $x' \sim \mathcal{T}(x,u)$ (the dependence on θ_{old} is omitted), and it measures the improvement of taking action u at state x over the old policy in terms of the value function $V^{\pi_{\text{old}}}$. A bound on the difference between $L_{\pi_{\text{old}}}(\pi)$ and $\eta(\pi)$ has been derived in [41], which also proves a monotonic improvement result as long as the KL divergence between the new and old policies is small (i.e., the new policy stays within the trust region). In practice, the surrogate loss $L_{\pi_{\text{old}}}(\pi)$ can be estimated using trajectories sampled from π_{old} as follows,

$$\widehat{L}_{\pi_{\text{old}}}(\pi) = \sum_t \frac{\pi(u_t|x_t)}{\pi_{\text{old}}(u_t|x_t)} \widehat{\Lambda}^{\pi_{\text{old}}}(x,u), \tag{2.4}$$

and the averaged KL divergence over observed states $\frac{1}{T}\sum_t KL\left[\pi_{\text{old}}(\cdot|x_t), \pi(\cdot|x_t)\right]$ can be used to estimate the trust region.

Natural gradient is defined by a metric based on the probability manifold induced by the KL divergence. It improves the standard gradient by making a step invariant to reparametrization of the parameter coordinates [3]:

$$\theta_{t+1} \leftarrow \theta_t - \lambda H_\theta^{-1}\zeta_t, \tag{2.5}$$

where ζ_t is the standard gradient, $H_\theta = \frac{1}{T}\sum_t \left(\frac{\partial}{\partial\theta}\pi_\theta(\log u_t|x_t)\right)\left(\frac{\partial}{\partial\theta}\log\pi_\theta(u_t|x_t)\right)^\top$ is the Fisher information matrix estimated with the trajectory data, and λ is the step size. In practice, when the number of parameters is large, conjugate gradient is employed to estimate the term $H_\theta^{-1}\zeta_t$ without requiring any matrix inversion. Since the Fisher information matrix coincides with the second-order approximation of the KL divergence, one can perform a back-tracking line search on the step size λ to ensure that the updated policy stays within the trust region.

Smoothness penalty is introduced in this study to empirically improve learning performance on physical dynamical systems. Specifically, we propose to use

$$L_{\text{explore}} = \sum\nolimits_{t=1}^{T} \|u_{t-1} - \pi_\theta(x_t)\|^2 \tag{2.6}$$

as a regularization term to induce consistency during exploration. The intuition is that since the change in states between two consecutive time steps is often small, it is desirable to ensure small changes in output actions. This is closely related to another penalty term that has been used in [15], which is termed “double backpropagation”, and recently rediscovered in [37, 22]:

$$L_{\text{smooth}} = \sum\nolimits_{t=1}^{T} \left\|\frac{\partial}{\partial\theta}\pi_\theta(x_t)\right\|^2, \tag{2.7}$$

which penalizes the gradient of the policy along the trajectories. Since bounded gradients lead to bounded Lipshitz constant, these penalties will induce smooth neural network functions, which is essential to ensure generalizability and, as we will show, stability. In addition, we incorporate a hard threshold (HT) approach that rescales the weight matrices at each layer by $(l^\circ/l(\pi_\theta))^{1/n_L}$ if $l(\pi_\theta) > l^\circ$, where $l(\pi_\theta)$ is the Lipschitz constant of the neural network π_θ, n_L is the number of layers of the neural network and l° is the certified Lipschitz constant. This ensures that the Lipschitz constant of the RL policy remains bounded by l°.

In summary, our policy gradient is based on the weighted objective:

$$L_{\text{pol}}(\pi_\theta) = \widehat{L}_{\pi_{\text{old}}}(\pi_\theta) + w_1 L_{\text{explore}}(\pi_\theta) + w_2 L_{\text{smooth}}(\pi_\theta), \tag{2.8}$$

where the penalty coefficients w_1 and w_2 are selected such that the scales of the corresponding terms are about $[0.01, 0.05]$ of the surrogate loss value $\widehat{L}_{\pi_{\text{old}}}(\pi_\theta)$. In each round, a set of trajectories are collected using π_{old}, which are used to estimate the gradient $\frac{\partial}{\partial\theta}L_{\text{pol}}(\pi_\theta)$ and the Fisher information matrix H_θ; a backtracking line search on the step size is then conducted to ensure that the updated policy stays within the trust region. This learning procedure is known as on-policy learning [47].

2.2. Overview of IQC framework. The IQC theory is celebrated for systematic and efficient stability analysis of a large class of uncertain, dynamic, and interconnected systems [33]. It unifies and extends classical passivity-based multiplier theory, and has close connections to dissipativity theory in the time domain [42].

To state the IQC framework, some terminologies are necessary. We define the space $L_2^n[0,\infty) = \{x : \int_{t=0}^{\infty} |x(t)|_2^2 dt < \infty\}$ for signals supported on $t \geq 0$, where n denotes the spatial dimension of $x(t)$, and the extended space $L_{2e}^n[0,\infty) = \{x :$

$\int_{t=0}^{T} |x(t)|_2^2 dt < \infty, \forall T \ \geq 0\}$ (we will use L_2 and L_{2e} if it is not necessary to specify the dimension and signal support), where we use x to denote the signal in general and $x(t)$ to denote its value at time t. For a vector or matrix, we use superscript $*$ to denote its conjugate transpose. An operator is causal if the current output does not depend on future inputs. It is bounded if it has a finite L_2 gain. Let $\Phi : \mathcal{H} \to \mathcal{H}$ be a bounded linear operator on a Hilbert space. Then, its Hilbert adjoint is the operator $\Phi^* : \mathcal{H} \to \mathcal{H}$ such that $\langle \Phi x, y\rangle = \langle x, \Phi^* y\rangle$ for all $x, y \in \mathcal{H}$, where $\langle\cdot,\cdot\rangle$ denotes the inner product. It is *self-adjoint* if $\Phi = \Phi^*$.

Consider the system (see also Figure 1)

$$y = G(u) \tag{2.9}$$

$$u = \Delta(y) + e, \tag{2.10}$$

where G is the transfer function of a causal and bounded LTI system (i.e., it maps input $u \in L^{n_a}$ to output $y \in L^{n_o}$ through the internal state dynamics $\dot{x} = Ax(t) + Bu(t)$), $e \in L^{n_a}$ is the disturbance, and $\Delta : L^{n_o} \to L^{n_a}$ is a bounded and causal function that is used to represent uncertainties in the system. IQC provides a framework to treat uncertainties such as nonlinear dynamics, model approximation and identification errors, time-varying parameters and disturbance noise, by using their input-output characterizations.

Definition 2.1 (Integral quadratic constraints). *Consider the signals $w \in L_2$ and $y \in L_2$ associated with Fourier transforms $\hat{w}$ and $\hat{y}$, and $w = \Delta(y)$, where Δ is a bounded and causal operator. We present both the frequency- and time-domain IQC definitions:*

(a) (Frequency domain) Let Π be a bounded and self-adjoint operator. Then, Δ is said to satisfy the IQC defined by Π (i.e., $\Delta \in IQC(\Pi)$) if:

$$\sigma_\Pi(\hat{y}, \hat{w}) = \int_{-\infty}^{\infty} \begin{bmatrix} \hat{y}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix}^* \Pi(j\omega) \begin{bmatrix} \hat{y}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix} d\omega \geq 0. \tag{2.11}$$

(b) (Time domain) Let (Ψ, M) be any factorization of $\Pi = \Psi^* M \Psi$ such that Ψ is stable and $M = M^\top$. Then, Δ is said to satisfy the hard IQC *defined by (Ψ, M) (i.e., $\Delta \in IQC(\Psi, M)$) if:*

$$\int_0^T z(t)^\top M z(t) dt \geq 0, \qquad \forall\, T \geq 0, \tag{2.12}$$

where $z = \Psi \begin{bmatrix} y \\ w \end{bmatrix}$ is the filtered output given by the stable operator Ψ. If instead of requiring nonnegativity at each time T, the nonnegativity is considered only when $T \to \infty$, then the corresponding condition is called soft IQC.

As established in [42], the time- and frequency-domain IQC definitions are equivalent if there exists $\Pi = \Psi^* M \Psi$ as a spectral factorization of Π with $M = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$ such that Ψ and Ψ^{-1} are stable.

Example 2.2 (Sector IQC). *A single-input single-output uncertainty $\Delta : \mathbb{R} \to \mathbb{R}$ is called "sector bounded" between $[\alpha, \beta]$ if $\alpha y(t) \leq \Delta(y(t)) \leq \beta y(t)$, for all $y \in \mathbb{R}$ and $t \geq 0$. It thus satisfies the sector $IQC(\Psi, M)$ with $\Psi = I$ and $M = \begin{bmatrix} -2\alpha\beta & \alpha+\beta \\ \alpha+\beta & -2 \end{bmatrix}$. It also satisfies $IQC(\Pi)$ with $\Pi = M$ defined above.*

EXAMPLE 2.3 (L_2 gain bound). *A MIMO uncertainty $\Delta : \mathbb{R}^n \to \mathbb{R}^m$ has the L_2 gain γ if $\int_0^\infty \|w(t)\|^2 dt \leq \gamma^2 \int_0^\infty \|y(t)\|^2 dt$, where $w(t) = \Delta(y(t))$. Thus, it satisfies IQC(Ψ, M) with $\Psi = I_{n+m}$ and $M = \begin{bmatrix} \lambda\gamma^2 I_n & 0 \\ 0 & -\lambda I_m \end{bmatrix}$, where $\lambda > 0$. It also satisfies IQC(Π) with $\Pi = M$ defined above. This can be used to characterize nonlinear operators with fast time-varying parameters.*

Before stating a stability result, we define the system (2.9)–(2.10) (see Figure 1) to be well-posed if for any $e \in L_{2e}$, there exists a solution $u \in L_{2e}$, which depends causally on e. A main IQC result for stability is stated below:

THEOREM 2.4 ([33]). *Consider the interconnected system (2.9)–(2.10). Assume that: **(i)** the interconnected system $(G, \tau\Delta)$ is well posed for all $\tau \in [0,1]$; **(ii)** $\tau\Delta \in$ IQC(Π) for $\tau \in [0,1]$; and **(iii)** there exists $\epsilon > 0$ such that*

$$\begin{bmatrix} \hat{G}(j\omega) \\ I(j\omega) \end{bmatrix}^* \Pi(j\omega) \begin{bmatrix} \hat{G}(j\omega) \\ I(j\omega) \end{bmatrix} \leq -\epsilon I, \qquad \forall\, \omega \in [0, \infty). \tag{2.13}$$

Then, the system (2.9)–(2.10) is input-output stable (i.e., finite L_2 gain).

The above theorem requires three technical conditions. The well-posedness condition is a generic property for any acceptable model of a physical system. The second condition is implied if $\Pi = \begin{bmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{12}^* & \Pi_{22} \end{bmatrix}$ has the properties $\Pi_{11} \succeq 0$ and $\Pi_{22} \preceq 0$. The third condition is central, and it requires checking the feasibility at every frequency, which represents a main obstacle. As discussed in Section Section 3.2, this condition can be equivalently represented as a linear matrix inequality (LMI) using the Kalman-Yakubovich-Popov (KYP) lemma. In general, the more IQCs exist for the uncertainty, the better characterization can be obtained. If $\Delta \in \mathrm{IQC}(\Pi_k)$, $k \in [n_K]$, where n_K is the number of IQCs satisfied by Δ, then it is easy to show that $\Delta \in \mathrm{IQC}(\sum_{k=1}^{n_K} \tau_k \Pi_k)$, where $\tau_k \geq 0$; thus, the stability test (2.13) becomes a convex program, i.e., to find $\tau_k \geq 0$ such that:

$$\begin{bmatrix} \hat{G}(j\omega) \\ I(j\omega) \end{bmatrix}^* \left(\sum_{k=1}^{n_K} \tau_k \Pi_k(j\omega) \right) \begin{bmatrix} \hat{G}(j\omega) \\ I(j\omega) \end{bmatrix} \leq -\epsilon I,\ \forall\, \omega \in [0, \infty). \tag{2.14}$$

The counterpart for the frequency-domain stability condition in the time-domain can be stated using a standard dissipation argument [42].

2.3. Related work. To close this section, we summarize some connections to existing literature. This work is closely related to the body of works on *safe reinforcement learning*, defined as the process of learning policies that maximize performance in problems where safety is required during the learning and/or deployment [20]. A detailed literature review can be found in [20], which has categorized two main approaches by modifying: **(1)** the optimality condition with a safety factor, and **(2)** the exploration process to incorporate external knowledge or risk metrics. Risk-aversion can be specified in the reward function, for example, by defining risk as the probability of reaching a set of unknown states in a discrete Markov decision process setting [14, 21]. Robust MDP is designed to maximize rewards while safely exploring the discrete state space [36, 50]. For continuous states and actions, robust model predictive control can be employed to ensure robustness and safety constraints for the learned model with bounded errrors [7]. These methods require an accurate or estimated models for policy learning. Recently, model-free policy optimization has been successfully demonstrated in real-world tasks such as robotics, business management, smart grid and transportation [31]. Safety requirement is high in these settings. Existing approaches are based on constraint satisfaction that holds with high probability [45, 1].

The present analysis tackles the safe reinforcement learning problem from a robust control perspective, which is aimed at providing theoretical guarantees for stability [55]. Lyapunov functions are widely used to analyze and verify stability when the system and its controller are known [39, 10]. For nonlinear systems without global convergence guarantees, region of convergence is often estimated, where any state trajectory that starts within this region stays within the region for all times and converges to a target state eventually [27]. For example, recently, [9] has proposed a learning-based Lyapunov stability verification for physical systems, whose dynamics are sequentially estimated by Gaussian processes. In the same vein, [2] has employed reachability analysis to construct safe regions in the state space by solving a partial differential equation. The main challenge of these methods is to find a suitable non-conservative Lyapunov function to conduct the analysis.

The IQC framework proposed in [33] has been widely used to analyze the stability of large-scale complex systems such as aircraft control [19]. The main advantages of IQC are its computational efficiency, non-conservatism, and unified treatment of a variety of nonlinearities and uncertainties. It has also been employed to analyze the stability of small-sized neural networks in reinforcement learning [28, 5]; however, in their analysis, the exact coefficients of the neural network need to be known a priori for the static stability analysis, and a region of safe coefficients needs to be calculated at each iteration for the dynamic stability analysis. This is computationally intensive, and it quickly becomes intractable when the neural network size grows. On the contrary, because the present analysis is based on a broad characterization of control functions with bounded gradients, it does not need to access the coefficients of the neural network (or any forms of the controller). In general, robust analysis using advanced methods such as structured singular value [38] or IQC can be conservative. There are only few cases where the necessity conditions can be established, such as when the uncertain operator has a block diagonal structure of bounded singular values [16], but this set of uncertainties is much smaller than the set of performance-oriented controllers learned by RL. To this end, we are able to reduce conservatism of the results by introducing more informative quadratic constraints for those controllers, and analyze the necessity of the certificate criteria. This significantly extends the possibilities of stability-certified reinforcement learning to large and deep neural networks in nonlinear large-scale real-world systems, whose stability is otherwise impossible to be certified using existing approaches.

3. Main results.

This section will introduce a set of quadratic constraints on gradient-bounded functions, describe the computation of a smoothness margin for linear (Theorem 3.3) and nonlinear systems (Theorem 3.4). Furthermore, we examine the conservatism of the certificate condition in Theorem 3.3 for linear systems.

3.1. Quadratic constraints on gradient-bounded functions.

The starting point of this analysis is a less conservative constraint on general vector-valued functions. We start by recalling the definition of a Lipschitz continuous function:

Definition 3.1 (Lipschitz continuous function). *We define both the local and global versions of the Lipschitz continuity for a function $f : \mathbb{R}^n \to \mathbb{R}^m$:*

(a) The function f is locally Lipschitz continuous *on the open subset $\mathcal{B}$ if there*

exists a constant $\xi > 0$ (i.e., Lipschitz constant of f on $\mathcal{B}$) such that

$$|f(x) - f(y)| \leq \xi |x - y|, \qquad \forall\, x, y \in \mathcal{B}. \tag{3.1}$$

(b) If f is Lipschitz continuous on $\mathbb{R}^n$ with a constant ξ (i.e., $\mathcal{B} = \mathbb{R}^n$ in (3.1)), then f is called globally Lipschitz continuous *with the Lipschitz constant ξ.*

Lipschitz continuity implies uniform continuity. The above definition also establishes a connection between locally and globally Lipschitz continuity. The norm $|\cdot|$ in the definition can be any norm, but the Lipschitz constant depends on the particular choice of the norm. Unless otherwise stated, we use the Euclidean norm in our analysis.

To explore some useful properties of Lipschitz continuity, consider a scalar-valued function (i.e., $m = 1$). Let $h_{xy}^{(j)} = \left[y_1, y_2, \ldots, y_j, x_{j+1}, \ldots, x_n\right]^\top \in \mathbb{R}^n$ denote a *hybrid vector* between x and y, with $h_{xy}^{(0)} = x$ and $h_{xy}^{(n)} = y$. Then, local Lipschitz continuity of $f : \mathbb{R}^n \to \mathbb{R}$ on $\mathcal{B}$ implies that

$$\frac{|f(h_{xy}^{(j)}) - f(h_{xy}^{(j-1)})|}{|x_j - y_j|} \leq \xi, \qquad \forall\, x, y \in \mathcal{B}, x_j \neq y_j, j \in [n]. \tag{3.2}$$

If we were to assume that f is differentiable, then it follows that its (partial) derivative is bounded by the Lipschitz constant. For a vector-valued function $f = \left[f_1, \ldots, f_m\right]^\top$ that is ξ-Lipschitz, it is necessary that every component f_i be ξ-Lipschitz. In general, every continuously differentiable function is locally Lipschitz, but the reverse is not true, since the definition of Lipschitz continuity does not require differentiability. Indeed, by the Rademacher's theorem, if f is locally Lipschitz on $\mathcal{B}$, then it is differentiable at *almost* every point in $\mathcal{B}$ [13].

For the purpose of stability analysis, we can express (3.1) as a point-wise quadratic constraint:

$$\begin{bmatrix} x - y \\ f(x) - f(y) \end{bmatrix}^\top \begin{bmatrix} \xi^2 I_n & 0 \\ 0 & -I_m \end{bmatrix} \begin{bmatrix} x - y \\ f(x) - f(y) \end{bmatrix} \geq 0, \qquad \forall\, x, y \in \mathcal{B}. \tag{3.3}$$

The above constraint, nevertheless, can be sometimes too conservative, because it does not explore the structure of a given problems. To elaborate on this, consider the function $f : \mathbb{R}^2 \to \mathbb{R}^2$ defined as

$$f(x_1, x_2) = \left[\tanh(0.5x_1) - ax_1, \sin(x_2)\right]^\top, \tag{3.4}$$

where $x_1, x_2 \in \mathbb{R}$ and $|a| \leq 0.1$ is a deterministic but unknown parameter with a bounded magnitude. Clearly, to satisfy (3.1) on $\mathbb{R}^2$ for all possible tuples (a, x_1, x_2), we need to choose $\xi \geq 1$ (i.e., the function has the Lipshitz constant 1). However, this characterization is too general in this case, because it ignores the *non-homogeneity* of f_1 and f_2, as well as the *sparsity* of the problem representation. Indeed, f_1 only depends on x_1 with its slope restricted to $[-0.1, 0.6]$ for all possible $|a| \leq 0.1$, and f_2 only depends on x_2 with its slope restricted to $[-1, 1]$. In the context of controller design, the non-homogeneity of control outputs often arises from physical constraints and domain knowledge, and the sparsity of control architecture is inherent in scenarios with distributed local information. To explicitly address these requirements, we state the following quadratic constraint.

Lemma 3.2. *For a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$ that is differentiable with bounded partial derivatives on $\mathcal{B}$ (i.e., $\underline{\xi}_{ij} \leq \partial_j f_i(x) \leq \overline{\xi}_{ij}$ for all $x \in \mathcal{B}$), the following* *quadratic constraint is satisfied for all $\lambda_{ij} \geq 0$, $i \in [m]$, $j \in [n]$, and $x, y \in \mathcal{B}$:*

$$\begin{bmatrix} x-y \\ q(x,y) \end{bmatrix}^\top M(\lambda;\xi) \begin{bmatrix} x-y \\ q(x,y) \end{bmatrix} \geq 0, \tag{3.5}$$

where $M(\lambda;\xi)$ is given by

$$\begin{bmatrix} \operatorname{diag}\left(\{\sum_i \lambda_{ij}(\bar{c}_{ij}^2 - c_{ij}^2)\}\right) & U(\{\lambda_{ij}, c_{ij}\})^\top \\ U(\{\lambda_{ij}, c_{ij}\}) & \operatorname{diag}\left(\{-\lambda_{ij}\}\right) \end{bmatrix}, \tag{3.6}$$

where $\operatorname{diag}(x)$ denotes a diagonal matrix with diagonal entries specified by x, and $q(x,y) = \left[q_{11}, \ldots, q_{1n}, \ldots, q_{m1}, \ldots, q_{mn}\right]^\top$ is determined by x and y, $\{-\lambda_{ij}\}$ is a set of non-negative multipliers that follow the same index order as q, $U(\{\lambda_{ij}, c_{ij}\}) = \left[\operatorname{diag}\left(\{-\lambda_{1j}c_{1j}\}\right) \ \cdots \ \operatorname{diag}\left(\{-\lambda_{mj}c_{mj}\}\right)\right] \in \mathbb{R}^{n\times mn}$, $c_{ij} = \frac{1}{2}\left(\underline{\xi}_{ij} + \overline{\xi}_{ij}\right)$, $\bar{c}_{ij} = \overline{\xi}_{ij} - c_{ij}$, and q is related to the output of f by the constraint:

$$f(x) - f(y) = \left[I_m \otimes 1_{1\times n}\right] q = Wq, \tag{3.7}$$

where $\otimes$ denotes the Kronecker product.

Proof. For a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$ that is differentiable with bounded partial derivatives on $\mathcal{B}$ (i.e., $\underline{\xi}_{ij} \leq \partial_j f_i(x) \leq \overline{\xi}_{ij}$ for all $x \in \mathcal{B}$), there exist functions $\delta_{ij} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ bounded by $\underline{\xi}_{ij} \leq \delta_{ij}(x,y) \leq \overline{\xi}_{ij}$ for all $i \in [m]$ and $j \in [n]$ such that

$$f(x) - f(y) = \begin{bmatrix} \sum_{j=1}^n \delta_{1j}(x,y)(x_j - y_j) \\ \vdots \\ \sum_{j=1}^n \delta_{mj}(x,y)(x_j - y_j) \end{bmatrix}. \tag{3.8}$$

By defining $q_{ij} = \delta_{ij}(x,y)(x_j - y_j)$, since $\left(\delta_{ij}(x,y) - c_{ij}\right)^2 \leq \bar{c}_{ij}^2$, it follows that

$$\begin{bmatrix} x_j - y_j \\ q_{ij} \end{bmatrix}^\top \begin{bmatrix} \bar{c}_{ij}^2 - c_{ij}^2 & c_{ij} \\ c_{ij} & -1 \end{bmatrix} [\star] \geq 0. \tag{3.9}$$

The result follows by introducing nonnegative multipliers $\lambda_{ij} \geq 0$, and the fact that $f_i(x) - f_i(y) = \sum_{j=1}^m q_{ij}$. ∎

This above bound is a direct consequence of standard tools in real analysis [54]. To understand this result, it can be observed that (3.5) is equivalent to:

$$\sum_{i,j} \lambda_{ij}\left((\bar{c}_{ij}^2 - c_{ij}^2)(x_j - y_j)^2 + 2c_{ij}q_{ij}(x_j - y_j) - q_{ij}^2\right) \geq 0, \qquad \forall \lambda_{ij} \geq 0, \tag{3.10}$$

with $f_i(x) - f_i(y) = \sum_{j=1}^n q_{ij}$, where q_{ij} depends on x and y. Since (3.10) holds for all $\lambda_{ij} \geq 0$, it is equivalent to the condition that $(\bar{c}_{ij}^2 - c_{ij}^2)(x_j - y_j)^2 + 2c_{ij}q_{ij}(x_j - y_j) - q_{ij}^2 \geq 0$ for all $i \in [m]$ and $j \in [n]$, which is a direct result of the bounds imposed on the partial derivatives of f_i. To illustrate its usage, let us apply the constraint to characterize the example function (3.4), where $\underline{\xi}_{11} = -0.1, \overline{\xi}_{11} = 0.6, \underline{\xi}_{22} = -1, \overline{\xi}_{22} = 1$, and all the other bounds $(\underline{\xi}_{12}, \overline{\xi}_{12}, \underline{\xi}_{21}, \overline{\xi}_{21})$ are zero. This clearly yields a more informative constraint than merely relying on the Lipschitz constraint (3.3). In fact, for a differentiable l-Lipschitz function, we have $\overline{\xi}_{ij} = -\underline{\xi}_{ij} = l$, and by limiting the choice of $\lambda_{ij} = \begin{cases} \lambda & \text{if } i = 1 \\ 0 & \text{if } i \neq 1 \end{cases}$, (3.10) is reduced to (3.3). However, as illustrated in this example, the quadratic constraint in Lemma 3.2 can incorporate richer information about the structure of the problem; therefore, it often gives rise to non-trivial stability bounds in practice.

The constraint introduced above is not a classical IQC, since it involves an intermediate variable q that relates to the output f through a set of linear equalities. For stability analysis, let $y = x^* \in \mathcal{B}$ be the equilibrium point, and without loss of generality, assume that $x^* = 0$ and $f(x^*) = 0$. Then, one can define the quadratic functions

$$\phi_{ij}(x, q) = (\overline{c}_{ij}^2 - c_{ij}^2)x_j^2 + 2c_{ij}q_{ij}x_j - q_{ij}^2,$$

and the condition (3.5) can be written as

$$\sum_{ij} \lambda_{ij}\phi_{ij}(x, q) \geq 0, \qquad \forall \lambda_{ij} \geq 0, \tag{3.11}$$

which can be used to characterize the set of (x, q) associated with the function f, as we will discuss in Section 3.4.

To simplify the mathematical treatment, we have focused on differentiable functions in Lemma 3.2; nevertheless, the analysis can be extended to non-differentiable but continuous functions (e.g., the ReLU function $\max\{0, x\}$) using the notion of generalized gradient [13, Chap. 2]. In brief, by re-assigning the bounds on partial derivatives to uniform bounds on the set of generalized partial derivatives, the constraint (3.5) can be directly applied.

In relation to the existing IQCs, this constraint has wider applications for the characterization of gradient-bounded functions. The Zames-Falb IQC introduced in [53] has been widely used for single-input single-output (SISO) functions $f : \mathbb{R} \to \mathbb{R}$, but it requires the function to be monotone with the slope restricted to $[\alpha, \beta]$ with $\alpha \geq 0$, i.e., $0 \leq \alpha \leq \frac{f(x)-f(y)}{x-y} \leq \beta$ whenever $x \neq y$. The MIMO extension holds true only if the nonlinear function $f : \mathbb{R}^n \to \mathbb{R}^n$ is restricted to be the gradient of a convex real-valued function [40, 24]. As for the sector IQC, the scalar version can not be used (because it requires $f_i(x) = 0$ whenever there exists $j \in [n]$ such that $x_j = 0$, which is extremely restrictive), and the vector version is in fact (3.3). In contrast, the quadratic constraint in Lemma 3.2 can be applied to non-monotone, vector-valued Lipschitz functions.

3.2. Computation of the smoothness margin. With the newly developed quadratic constraint in place, this subsection explains the computation for a smoothness margin of an LTI system G, whose state-space representation is given by:

$$\begin{cases} \dot{x}_G &= Ax_G + Bu \\ w &= \pi(x_G) \\ u &= e + w \end{cases} \tag{3.12}$$

where $x_G \in \mathbb{R}^{n_s}$ is the state (the dependence on t is omitted for simplicity). The system is assumed to be stable, i.e., A is Hurwitz. We can connect this linear system in feedback with a controller $\pi : \mathbb{R}^{n_s} \to \mathbb{R}^{n_a}$. The signal $e \in \mathbb{R}^{n_a}$ is the exploration vector introduced in reinforcement learning, and $w \in \mathbb{R}^{n_a}$ is the policy action. We are interested in certifying the set of gradient bounds $\xi \in \mathbb{R}^{n_s \times n_a}$ of π such that the interconnected system is input-output stable at all time $T \geq 0$, i.e.,

$$\int_0^T |y(t)|^2 \, dt \leq \gamma^2 \int_0^T |e(t)|^2 \, dt, \tag{3.13}$$

where γ is a finite upper bound for the L_2 gain. Let $A \succeq B$ or $A \succ B$ denote that $A - B$ is positive semidefinite or positive definite, respectively. To this end, define the $\mathrm{SDP}(P, \lambda, \gamma, \xi)$ as follows:

$$\mathrm{SDP}(P, \lambda, \gamma, \xi) : \begin{bmatrix} O(P, \lambda, \xi) & S(P) \\ S(P)^\top & -\gamma I \end{bmatrix} \prec 0, \tag{3.14}$$

where $P = P^\top \succeq 0$ and

$$O(P, \lambda, \xi) = \begin{bmatrix} A^\top P + PA & PBW \\ W^\top B^\top P & 0 \end{bmatrix} + \frac{1}{\gamma} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} + M(\lambda; \xi), \quad S(P) = \begin{bmatrix} PB \\ 0 \end{bmatrix},$$

where $M(\lambda; \xi)$ is defined in (3.6). We will show next that the stability of the interconnected system can be certified using linear matrix inequalities.

Theorem 3.3. *Let G be stable (i.e., A is Hurwitz) and $\pi \in \mathbb{R}^{n_s} \to \mathbb{R}^{n_a}$ be a bounded causal controller. Assume that:*

(i) the interconnection of G and π is well-posed;

(ii) π has bounded partial derivatives on $\mathcal{B}$ (i.e., $\underline{\xi}_{ij} \leq \partial_j \pi_i(x) \leq \overline{\xi}_{ij}$, for all $x \in \mathcal{B}$, $i \in [n_a]$ and $j \in [n_s]$).

If there exist $P \succeq 0$ and a scalar $\gamma > 0$ such that SDP$(P, \lambda, \gamma, \xi)$ is feasible, then the feedback interconnection of G and π is stable (i.e., it satisfies (3.13)).

Proof. The proof follows a standard dissipation argument. To proceed, we multiply $\begin{bmatrix} x_G^\top & q^\top & e^\top \end{bmatrix}^\top$ to the left and its transpose to the right of the augmented matrix in (3.14), and use the constraints $w = Wq$ and $y = x_G$. Then, $\mathrm{SDP}(P, \lambda, \gamma, \xi)$ can be written as a dissipation inequality:

$$\dot{V}(x_G) + \begin{bmatrix} x_G \\ q \end{bmatrix}^\top M(\lambda; \xi) \begin{bmatrix} x_G \\ q \end{bmatrix} < \gamma e^\top e - \frac{1}{\gamma} y^\top y,$$

where $V(x_G) = x_G^\top P x_G$ is known as the storage function, and $\dot{V}(\cdot)$ is its derivative with respect to time t. Because the second term is guaranteed to be non-negative by Lemma 3.2, if $\mathrm{SDP}(P, \lambda, \gamma, \xi)$ is feasible with a solution $(P, \lambda, \gamma, \xi)$, we have:

$$\dot{V}(x_G) + \frac{1}{\gamma} y^\top y - \gamma e^\top e < 0, \tag{3.15}$$

which is satisfied at all times t. From well-posedness, the above inequality can be integrated from $t = 0$ to $t = T$, and then it follows from $P \succeq 0$ that:

$$\int_0^T |y(t)|^2 dt \leq \gamma^2 \int_0^T |e(t)|^2 dt. \tag{3.16}$$

Hence, the interconnected system with the RL policy π is stable. ∎

The above theorem requires that G be stable when there is no feedback policy π. This is automatically satisfied in many physical systems with an existing stabilizing (but not performance-optimizing) controller. In the case that the original system is not stable, one needs to first design a controller to stablize the system or design the controller under uncertainty (in this case, the RL policy), which are well-studied problems in the literature (e.g., H_∞ controller synthesis [16]). Then, the result can be used to ensure stability while delegating reinforcement learning to optimize the performance of the policy under gradient bounds.

The above result essentially suggests a computational approach in robust control analysis. Given a stable LTI system depicted in (3.12), the first step is to represent the RL policy as an uncertainty block in a feedback interconnection. Because the parameters of the neural network policy may not be known *a priori* and will be continuously updated during learning, we characterize it using bounds on partial gradients (e.g., if it is known that the action is positively correlated with certain observation metric, we can specify its partial gradient to be mostly positive with only a small negative margin). A simple but conservative choice is a L_2-gain bound IQC; nevertheless, to achieve a less conservative result, we can employ the quadratic constraint developed in Lemma 3.2, which exploits both the sparsity of the control architecture and the non-homogeneity of the outputs. For a given set of gradient bounds ξ, we find the smallest γ such that (3.14) is feasible, and γ corresponds to the upper bound on the L_2 gain of the interconnected system both during learning (with the excitation e added to facilitate policy exploration) and actual deployment. If γ is finite, then the system is provably stable in the sense of (3.13).

We remark that $\text{SDP}(P, \lambda, \gamma, \xi)$ is quasiconvex, in the sense that it reduces to a standard LMI with a fixed γ. To solve it numerically, we start with a small γ and gradually increase it until a solution (P, λ) is found. This is repeated for multiple sets of ξ. Each iteration (i.e., LMI for a given set of γ and ξ) can be solved efficiently by interior-point methods. As an alternative to searching on γ for a given ξ, more sophisticated methods for solving the generalized eigenvalue optimization problem can be employed [11].

3.3. Extension to nonlinear systems with uncertainty. The previous analysis for LTI systems can be extended to a generic nonlinear system described in (1.1). The key idea is to model the nonlinear and potentially time-varying part $g_t(x(t))$ as an uncertain block with IQC constraints on its behavior. Specifically, consider the LTI component $\underline{G}$:

$$\begin{cases} \dot{x}_G &= Ax_G + Bu + v \\ y &= x_G \end{cases} \tag{3.17}$$

where $x_G \in \mathbb{R}^{n_s}$ is the state and $y \in \mathbb{R}^{n_s}$ is the output. The linearized system is assumed to be stable, i.e., A is Hurwitz. The nonlinear part is connected in feedback:

$$\begin{cases} u &= e + w \\ w &= \pi(y) \\ v &= g_t(y) \end{cases} \tag{3.18}$$

where $e \in \mathbb{R}^{n_a}$ and $w \in \mathbb{R}^{n_a}$ are defined as before, and $g_t : \mathbb{R}^{n_s} \to \mathbb{R}^{n_s}$ is the nonlinear and time-varying component. In addition to characterizing π using the Lipschitz property as in (3.5), we assume that $g_t : \mathbb{R}^{n_s} \to \mathbb{R}^{n_s}$ satisfies the IQC defined by

(Ψ, M_g) as in Definition 2.1. The system Ψ has the state-space representation:

$$\begin{cases} \dot{\psi} &= A_\psi \psi + B_\psi^v v + B_\psi^y y \\ z &= C_\psi \psi + D_\psi^v v + D_\psi^y y \end{cases}, \tag{3.19}$$

where $\psi \in \mathbb{R}^{n_s}$ is the internal state and $z \in \mathbb{R}^{n_z}$ is the filtered output. By denoting $x = \begin{bmatrix} x_G^\top & \psi^\top \end{bmatrix}^\top \in \mathbb{R}^{2n_s}$ as the new state, one can combine (3.17) and (3.19) via reducing y and letting $w = Wq$:

$$\begin{cases} \dot{x} = \underbrace{\begin{bmatrix} A & 0 \\ B_\psi^y & A_\psi \end{bmatrix}}_{\underline{A}} x + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{\underline{B}_e} e + \underbrace{\begin{bmatrix} BW \\ 0 \end{bmatrix}}_{\underline{B}_q} q + \underbrace{\begin{bmatrix} I \\ B_\psi^v \end{bmatrix}}_{\underline{B}_v} v \\ z = \underbrace{\begin{bmatrix} D_\psi^y & C_\psi \end{bmatrix}}_{\underline{C}} x + D_\psi^v v \end{cases}, \tag{3.20}$$

where $\underline{A}$, $\underline{B}_e$, $\underline{B}_q$, $\underline{B}_v$, $\underline{C}$ are matrices of proper dimensions defined above. Similar to the case of LTI systems, the objective is to find the gradient bounds on π such that the system becomes stable in the sense of (3.13). In the same vein, we define $\underline{\mathrm{SDP}}(P, \lambda, \gamma, \xi)$ as:

$$\underline{\mathrm{SDP}}(P, \lambda, \gamma, \xi): \begin{bmatrix} O(P, \lambda, \xi) & O_v(P) & S(P) \\ O_v(P)^\top & D_\psi^{v\top} M_q D_\psi^v & 0 \\ S(P)^\top & 0 & -\gamma I \end{bmatrix} \prec 0, \tag{3.21}$$

where $P \succeq 0$, and

$$O(P, \lambda, \xi) = \begin{bmatrix} \underline{A}^\top P + P\underline{A} & P\underline{B}_q \\ \underline{B}_q^\top P & 0 \end{bmatrix} + \begin{bmatrix} \underline{C}^\top M_g \underline{C} & 0 \\ 0 & 0 \end{bmatrix} + M(\lambda; \xi) + \frac{1}{\gamma} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix},$$

$$O_v(P) = \begin{bmatrix} \underline{C}^\top M_q D_\psi^v + P\underline{B}_v \\ 0 \end{bmatrix}, \; S(P) = \begin{bmatrix} P\underline{B}_e \\ 0 \end{bmatrix},$$

where $M(\lambda; \xi)$ is defined in (3.6). The next theorem provides a stability certificate for the nonlinear time-varying system (1.1).

THEOREM 3.4. *Let $\underline{G}$ be stable (i.e., A in (3.17) is Hurwitz) and $\pi \in \mathbb{R}^{n_s} \to \mathbb{R}^{n_a}$ be a bounded causal controller. Assume that:*

(i) the interconnection of $\underline{G}$, π, and g_t is well-posed;

(ii) π has bounded partial derivatives on $\mathcal{B}$ (i.e., $\underline{\xi}_{ij} \le \partial_j \pi_i(x) \le \overline{\xi}_{ij}$ for all $x \in \mathcal{B}$, $i \in [n_a]$ and $j \in [n_s]$);

(iii) $g_t \in IQC(\Psi, M_g)$, where Ψ is stable.

If there exist $P \succeq 0$ and a scalar $\gamma > 0$ such that $\underline{\mathrm{SDP}}(P, \lambda, \gamma, \xi)$ in (3.21) is feasible, then the feedback interconnection of the nonlinear system (1.1) and π is stable (i.e., it satisfies (3.13)).

Proof. The proof is in the same vein as that of Theorem 3.3. The main technical difference is the consideration of the filtered state ψ and the output z to impose IQC constraints on the nonlinearities $g_t(y)$ in the dynamical system [33]. The dissipation inequality follows by multiplying both sides of the matrix in (3.21) by $\begin{bmatrix} x^\top & q^\top & v^\top & e^\top \end{bmatrix}^\top$ and its transpose:

$$\dot{V}(x) + z^\top M_g z + \begin{bmatrix} x_G \\ q \end{bmatrix}^\top M_\pi \begin{bmatrix} x_G \\ q \end{bmatrix} < \gamma e^\top e - \frac{1}{\gamma} y^\top y,$$

where x and z are defined in (3.20), and $V(x) = x^\top Px$ is the storage function with $\dot{V}(\cdot)$ as its time derivative. The second term on the left side is non-negative because $g_t \in \mathrm{IQC}(\Psi, M_g)$, and the third term is non-negative due to the smoothness quadratic constraint in Lemma 3.2. Thus, if there exists a feasible solution $P \succeq 0$ to $\underline{\mathrm{SDP}}(P, \lambda, \gamma, \xi)$, integrating the inequality from $t = 0$ to $t = T$ yields that:

$$\int_0^T |y(t)|^2 dt \leq \gamma^2 \int_0^T |e(t)|^2 dt. \tag{3.22}$$

Hence, the nonlinear system interconnected with the RL policy π is certifiably stable in the sense of a finite L_2 gain. □

3.4. Analysis of conservatism of the stability certificate. We focus on the case where an LTI system G is interconnected with an RL policy $\pi \in \mathcal{P}(\xi)$ (i.e., a function with bounded partial gradients). This corresponds to the system (3.12) studied in Section 3.2. To certify the stability of (3.12), as will be shown in the next proposition, it suffices to examine the stability of the following system:

$$\begin{cases} \dot{x}_G &= Ax_G + Bu \\ q &= \tilde{\pi}(x_G) \\ w &= Wq \\ u &= e + w \end{cases} \tag{3.23}$$

where $\tilde{\pi} \in \widetilde{\mathcal{P}}$ is a function in the uncertainty set:

$$\widetilde{\mathcal{P}}(\xi) = \left\{ \tilde{\pi} \;\middle|\; \underline{\xi}_{ij} x_j \leq \tilde{\pi}_{ij}(x) \leq \overline{\xi}_{ij} x_j, \forall x \in \mathbb{R}^{n_s}, i \in [n_a], j \in [n_s] \right\}. \tag{3.24}$$

Proposition 3.5. *If the system (3.23) is stable for all $\tilde{\pi} \in \widetilde{\mathcal{P}}(\xi)$, then the system (3.12) is stable for all $\pi \in \mathcal{P}(\xi)$.*

Proof. It suffices to show that for any $\pi \in \mathcal{P}(\xi)$, there exists a policy $\tilde{\pi} \in \widetilde{\mathcal{P}}(\xi)$ such that $\pi = W\tilde{\pi}$. Let $y_j^0 = \begin{bmatrix} 0 & \cdots & 0 & y_{j+1} & \cdots & y_{n_s} \end{bmatrix} \in \mathbb{R}^{n_s}$ for every $j \in \{0, 1, ..., n_s\}$, and $y_0^0 = y$, $y_{n_s}^0 = 0$. Then, one can write:

$$\pi_i(y) = \sum_{j=1}^{n_s} \pi_i(y_{j-1}^0) - \pi_i(y_j^0) = \sum_{j=1}^{n_s} \tilde{\pi}_{ij}(y),$$

where $\tilde{\pi}_{ij}(y)$ satisfies

$$\frac{\tilde{\pi}_{ij}(y)}{y_j} = \frac{\pi_i(y_{j-1}^0) - \pi_i(y_j^0)}{|y_{j-1}^0 - y_j^0|} \in [\underline{\xi}_{ij}, \overline{\xi}_{ij}]$$

if $y_j \neq 0$ and $\tilde{\pi}_{ij}(y) = 0$ if $y_j = 0$. The bound is due to the mean-value theorem and the bounds on the partial derivatives of π_i. Since the above argument is valid for all $i \in [n_a]$, it means that $\tilde{\pi} \in \widetilde{\mathcal{P}}(\xi)$, and $\pi = W\tilde{\pi}$. □

Proposition 3.5 implies that one potential source of conservatism comes from the decomposition of a gradient-bounded function into a sum of sector-bounded components. Henceforth, we focus the subsequent analysis by examining (3.23). By considering the state-space representation of $G = \left[\begin{array}{c|cc} A & BW & B \\ \hline I & 0 & 0 \end{array}\right] = \begin{bmatrix} G_{11} & G_{12} \end{bmatrix}$, one can write system (3.23) as:

$$\begin{cases} x_G = \begin{bmatrix} G_{11} & G_{12} \end{bmatrix} \begin{bmatrix} q \\ e \end{bmatrix} \\ q \quad = \tilde{\pi}(x_G) \end{cases}. \tag{3.25}$$

It is known that the system is input-output stable if and only if $I - G_{11}\tilde{\pi}$ is nonsingular [16]. To understand this, note that if $I - G_{11}\tilde{\pi}$ is nonsingular, then the transfer from e to x_G is given by:

$$e \mapsto x_G = H(e) = (I - G_{11}\tilde{\pi})^{-1} G_{12} e,$$

and $|x_G| = |H(e)| \leq \|(I - G_{11}\tilde{\pi})^{-1} G_{12}\| |e|$. From the previous section (in particular, Lemma 3.2), we know that if the function π is gradient-bounded, then the set of input/output signals belongs to:

$$\mathcal{S}(\xi) = \left\{(x, q) \mid \phi_{ij}(x, q) = (\bar{c}_{ij}^2 - c_{ij}^2) x_j^2 + 2 c_{ij} q_{ij} x_j - q_{ij}^2 \geq 0, \quad \forall i \in [n_a], j \in [n_s]\right\},$$

where we use $c_{ij} = \frac{1}{2}\left(\underline{\xi}_{ij} + \bar{\xi}_{ij}\right)$, $\bar{c}_{ij} = \bar{\xi}_{ij} - c_{ij}$ for simplicity. We now show that the pair (x, q) belongs to $\mathcal{S}(\xi)$ if and only if there exists a sector-bounded function $\tilde{\pi} \in \tilde{P}(\xi)$ such that it satisfies $q = \tilde{\pi}(x)$.

Lemma 3.6. *Suppose that $x \in \mathbb{R}^{n_s}$ and $q \in \mathbb{R}^{n_a n_s}$, and $\bar{c}_{ij} \geq 0$ for every $i \in [n_a]$ and $j \in [n_s]$. Then, the pair (x, q) belongs to $\mathcal{S}(\xi)$ if and only if there exists an operator $\tilde{\pi} : \mathbb{R}^n \to \mathbb{R}^{n_a n_s}$, such that $q = \tilde{\pi}(x)$, and $\tilde{\pi}$ satisfies the following conditions:* ***(i)*** *$\tilde{\pi}_{ij}(x) = 0$ if $x_j = 0$, and* ***(ii)*** *$\tilde{\pi}$ is sector bounded, i.e., $(c_{ij} - \bar{c}_{ij}) x_j \leq \tilde{\pi}_{ij}(x) \leq (\bar{c}_{ij} + c_{ij}) x_j$ holds for all $i \in [n_a]$ and $j \in [n_s]$.*

Proof. To show the sufficiency direction, conditions (i) and (ii) yield that

$$(\bar{c}_{ij} x_j)^2 \geq \left(\left(\frac{\pi_{ij}(x)}{x_j} - c_{ij}\right) x_j\right)^2 = (q_{ij} - c_{ij} x_j)^2.$$

By rearranging the above inequality, it can be concluded that $(x, q) \in \mathcal{S}(\xi)$.

For the necessary direction, note that the condition $\phi_{ij}(x, q) \geq 0$ is equivalent to $|q_{ij} - c_{ij} x_j| \leq |\bar{c}_{ij} x_j|$. Thus, we have $\pi_{ij}(x) = 0$ if $x_j = 0$. Since $\bar{c}_{ij} \geq 0$, one can obtain $\left|\frac{q_{ij}}{x_j} - c_{ij}\right| \leq \bar{c}_{ij}$, which is equivalent to the sector bounds. □

By slightly overloading the notations, we can extend the result of the previous lemma from static mapping to the case that $x \in L^{n_s}$ and $q \in L^{n_a n_s}$ with the operator $\tilde{\pi} : L^{n_s} \to L^{n_a n_s}$. We can then extend the definition of $\mathcal{S}(\xi)$ to this space accordingly.

Lemma 3.7. *Suppose that $x \in L^{n_s}$ and $q \in L^{n_a n_s}$, and that $\bar{c}_{ij} \geq 0$ for all $i \in [n_a]$ and $j \in [n_s]$. Then, the pair (x, q) belongs to $\mathcal{S}(\xi)$ where*

$$\mathcal{S}(\xi) = \{(x, q) \mid \phi_{ij}(x, q) \geq 0, \quad \forall i \in [n_a], j \in [n_s]\}, \tag{3.26}$$

and

$$\phi_{ij}(x, q) = (\bar{c}_{ij}^2 - c_{ij}^2)\|x_j\|^2 + 2 c_{ij} \langle q_{ij}, x_j \rangle - \|q_{ij}\|^2 \geq 0, \tag{3.27}$$

if and only if there exists an operator $\tilde{\pi} : L^{n_s} \to L^{n_a n_s}$ such that $q = \tilde{\pi}(x)$ and $\tilde{\pi}$ satisfies the following conditions: ***(i)*** *$\tilde{\pi}_{ij}(x) = 0$ if $x_j = 0$, and* ***(ii)*** *$\tilde{\pi}$ is sector bounded, i.e., $(c_{ij} - \bar{c}_{ij})\|x_j\| \leq \|\tilde{\pi}_{ij}(x)\| \leq (\bar{c}_{ij} + c_{ij})\|x_j\|$ for all $i \in [m]$ and $j \in [n]$.*

Proof. For the sufficiency condition, since $\tilde{\pi}_{ij}$ is sector bounded, and $\tilde{\pi}_{ij}(x) = 0$ if $x_j = 0$, without loss of generality, assume that $c_{ij} \leq 0$. One can write

$$\begin{aligned}
\|\overline{c}_{ij} x_j\|^2 &\geq \left\| \left| \frac{\|\tilde{\pi}_{ij}(x)\|}{\|x_j\|} - c_{ij} \right| x_j \right\|^2 \\
&\geq \left\| \left(\frac{\|\tilde{\pi}_{ij}(x)\|}{\|x_j\|} - c_{ij} \right) x_j \right\|^2 \\
&= \|c_{ij} x_j\|^2 + \|\tilde{\pi}_{ij}(x)\|^2 - 2 \left\langle c_{ij} x_j, \frac{\|\tilde{\pi}_{ij}(x)\|}{\|x_j\|} x_j \right\rangle \\
&= \|c_{ij} x_j\|^2 + \|\tilde{\pi}_{ij}(x)\|^2 - 2 c_{ij} \|x_j\| \, \|\tilde{\pi}_{ij}(x)\| \\
&\geq \|c_{ij} x_j\|^2 + \|\tilde{\pi}_{ij}(x)\|^2 - 2 c_{ij} \left\langle \tilde{\pi}_{ij}(x), x_j \right\rangle \\
&= \|q_{ij} - c_{ij} x_j\|^2 .
\end{aligned}$$

By rearranging the above inequality, it can be concluded that $(x, q) \in \mathcal{S}(\xi)$.

For the necessary direction, we can construct $\tilde{\pi}(y) = q \frac{\langle y, x \rangle}{|x|^2}$ for all $y \in L^{n_s}$. This leads to $\tilde{\pi}(x) = q$, and the condition $\phi_{ij}(x, q) \geq 0$ is equivalent to $\|q_{ij} - c_{ij} x_j\| \leq \overline{c}_{ij} \|x_j\|$. Thus, we have $\tilde{\pi}_{ij}(x) = 0$ if $x_j = 0$. Without loss of generality, assume that $c_{ij} \leq 0$. Therefore, $\|q_{ij}\| \leq \overline{c}_{ij} \|x_j\| + c_{ij} \|x_j\|$ and $\|q_{ij}\| \geq -\overline{c}_{ij} \|x_j\| + c_{ij} \|x_j\|$, which are equivalent to the sector bound condition. □

The previous result indicates that the input and output pair of $\tilde{\pi}$ can be described by $\mathcal{S}(\xi)$. We show next that this set should be separated from the signal space of the dynamical system in order to ensure robust stability.

LEMMA 3.8. *If $(G, \tilde{\pi})$ is robustly stable, then there cannot exist a nonzero $q \in L_2$ such that $x = Gq$ and $(x, q) \in \mathcal{S}(\xi)$.*

Proof. We prove this lemma by contraposition. If there exists a nonzero $q \in L_2$ such that $(x, q) \in \mathcal{S}(\xi)$, then it follows from Lemma 3.7 that there exists a linear operator $\tilde{\pi}$ such that $q = \tilde{\pi}(x) = \tilde{\pi}(Gq)$. This implies that the operator $(I - \tilde{\pi} G)$ is singular, and therefore, $(I - G\tilde{\pi})$ is singular, implying that the interconnected system is not robustly stable. □

The path of examining the necessity of the SDP condition (3.14) has become clear. Consider the set generated by the LTI system:

$$\Psi = \left\{ (\phi_{ij}(x, q) : q \in L^{n_a n_s}, \|q\| = 1, x = Gq \right\}, \tag{3.28}$$

and the positive orthant

$$\Pi = \left\{ (r_{ij}) \in \mathbb{R}^{n_a n_s} : r_{ij} \geq 0, \quad \forall i \in [n_a], j \in [n_s] \right\}. \tag{3.29}$$

Lemma 3.8 implies that the two sets Ψ and Π are separated if $(G, \tilde{\pi})$ is robustly stable. The goal is to show that there exists a separating hyperplane, whose parameters are related to the solution of (3.14). For simplicity, define the matrices $\Omega_{ij,x} = \operatorname{diag}\left(\left\{ \{ \overline{c}_{ij}^2 - c_{ij}^2 \} \right) \right.$, $\Omega_{ij,q}$ and $\Omega_{ij,xq}$ with their (k,l)-th elements $[\Omega_{ij,q}]_{kl} = \begin{cases} 1 & \text{if } k = in + j \\ 0 & \text{otherwise} \end{cases}$, and $[\Omega_{ij,xq}]_{kl} = \begin{cases} c_{ij} & \text{if } k = j, l = in + j \\ 0 & \text{otherwise} \end{cases}$. To write $\phi_{ij}(x = Gq, q)$ as an inner product, define

$$T_{ij} = G^* \Omega_{ij,x} G - \Omega_{ij,q} + G^* \Omega_{ij,xq}^* + \Omega_{ij,xq} G.$$

It results from the definition (3.27) that

$$\phi_{ij}(x = Gq, q) = \|Gq\|^2_{\Omega_{ij,x}} + 2\text{Re}\left\langle \Omega_{ij,xq} Gq, q\right\rangle - \|q\|^2_{\Omega_{ij,q}} = \left\langle q, T_{ij} q\right\rangle . \tag{3.30}$$

Lemma 3.9. *For a given linear time-invariant operator G, the closure $\overline{\Psi}$ of Ψ defined in (3.28) is convex.*

Proof. Because G is time-invariant, by denoting D_τ as the delay operator at scale τ, we obtain $D^*_\tau T_{ij} D_\tau = T_{ij}$. Let $y = \phi(q)$ and $\tilde{y} = \phi(\tilde{q})$ be the elements of Ψ, with $\|q\| = \|\tilde{q}\| = 1$. By considering $q_\tau = \sqrt{\alpha} q + \sqrt{1-\alpha} D_\tau \tilde{q}$, one can write

$$\begin{aligned}\phi_{ij}(q_\tau) &= \alpha \left\langle T_{ij} q, q\right\rangle + (1-\alpha)\left\langle T_{ij} D_\tau \tilde{q}, D_\tau \tilde{q}\right\rangle + 2\alpha\sqrt{1-\alpha}\text{Re}\left\langle T_{ij} q, D_\tau \tilde{q}\right\rangle \\ &= \alpha\phi_{ij}(q) + (1-\alpha)\phi_{ij}(\tilde{q}) + 2\alpha\sqrt{1-\alpha}\text{Re}\left\langle T_{ij} q, D_\tau \tilde{q}\right\rangle . \qquad \square\end{aligned}$$

By letting $\tau \to \infty$, we obtain $\text{Re}\left\langle T_{ij} q, D_\tau \tilde{q}\right\rangle \to 0$, where $\text{Re}(x)$ denotes the real part of a complex vector x. Thus,

$$\lim_{\tau\to\infty} \phi_{ij}(q_\tau) = \alpha\phi_{ij}(q) + (1-\alpha)\phi_{ij}(\tilde{q})$$

and $\lim_{\tau\to\infty} \|q_\tau\|^2 = \alpha\|q\|^2 + (1-\alpha)\|\tilde{q}\|^2 = 1$. Therefore,

$$\lim_{\tau\to\infty} \phi\left(\frac{q_\tau}{\|q_\tau\|}\right) = \alpha y + (1-\alpha)\tilde{y} \in \overline{\Psi}.$$

Now, we show that strict separation occurs when the system is robustly stable.

Lemma 3.10. *Suppose that $I - G\tilde{\pi}$ is nonsingular. Then, the sets Π and Ψ are strictly separated, namely $D(\Pi, \Psi) = \inf_{r\in\Pi, y\in\Psi} |r - y| > 0$.*

To prove this result, we need the following lemma.

Lemma 3.11. *Suppose that $D(\Pi, \Psi) = \inf_{r\in\Pi, y\in\Psi} |r - y| = 0$. Given any $\epsilon > 0$ and $t_0 \geq 0$, there exist a closed interval $[t_0, t_1]$ and two signals $x \in L^{n_s}$ and $q \in L^{n_a n_s}$ with $\|q\| = 1$ such that*

$$\phi_{ij}(x, q) \geq 0, \qquad \forall i \in [n_a], j \in [n_s] \tag{3.31}$$

$$\epsilon^2 > \|(I - \Gamma_{[t_0,t_1]})Gq\| \tag{3.32}$$

$$\epsilon = \|x - \Gamma_{[t_0,t_1]} Gq\|_{\Omega_{ij,x}}, \tag{3.33}$$

where $\|q\|_\Omega = \sqrt{q^*\Omega q}$ is the scaled norm and $\Gamma_{[t_0,t_1]}$ projects the signal onto the support of $[t_0, t_1]$. With the above choice of q, x and $[t_0, t_1]$, there exists an operator $\tilde{\pi} \in \tilde{P}(\xi)$ such that $\|(I - \tilde{\pi}\Gamma_{[t_0,t_1]} G)q\| \leq C\epsilon$ for some constant $C > 0$ that depends on the sector bounds ξ.

Proof. For a given $\epsilon > 0$, by hypothesis, there exists $q \in L^{n_a n_s}$ with $\|q\| = 1$ satisfying $\phi_{ij}(x, q) > -\epsilon^2$ for all $i \in [n_a]$ and $j \in [n_s]$, i.e.,

$$\epsilon^2 + \|Gq\|^2_{\Omega_{ij,x}} + 2\text{Re}\left\langle \Omega_{ij,xq} Gq, q\right\rangle > \|q\|^2_{\Omega_{ij,q}},$$

where $\Omega_{ij,x}$ and $\Omega_{ij,xq}$ are defined previously. Clearly, if q is truncated to a sufficiently long interval, and q is rescaled to have a unit norm, the above inequality will still hold. Since $Gq \in L^{n_s}$, by possibly enlarging the truncation interval to $[t_0, t_1]$, we obtain (3.32), and

$$\epsilon^2 + \|\Gamma_{[t_0,t_1]} Gq\|^2_{\Omega_{ij,x}} + 2\text{Re}\left\langle \Omega_{ij,xq}\Gamma_{[t_0,t_1]} Gq, q\right\rangle > \|q\|^2_{\Omega_{ij,q}},$$

Next, we choose $\eta \in L^{n_s}$ such that $\|\eta\|^2_{\Omega_{ij,x}} = \epsilon^2$, and that η is orthogonal to $\Gamma_{[t_0,t_1]}Gq$ and $\Omega^*_{ij,xq}q$ for all $i \in [n_a]$ and $j \in [n_s]$. Then, by considering $x = \Gamma_{[t_0,t_1]}Gq + \eta$, we obtain

$$\|x\|^2_{\Omega_{ij,x}} = \|\Gamma_{[t_0,t_1]}Gq + \eta\|^2_{\Omega_{ij,x}} = \epsilon^2 + \|\Gamma_{[t_0,t_1]}Gq\|^2_{\Omega_{ij,x}},$$

which leads to $\phi_{ij}(x,q) \geq 0$ and (3.33). Now, we can invoke Lemma 3.7 to construct $\tilde{\pi} \in \mathcal{P}(\xi)$ based on (3.31) such that $\tilde{\pi}$ becomes sector bounded and $q = \tilde{\pi}x$. Then,

$$(I - \tilde{\pi}\Gamma_{[t_0,t_1]}G)q = \tilde{\pi}(x - \Gamma_{[t_0,t_1]}Gq).$$

Let $\|\tilde{\pi}\| \leq C$ (which depends on the sector bounds). Then,

$$\|(I - \tilde{\pi}\Gamma_{[t_0,t_1]}G)q\| \leq C\epsilon$$

. □

We are now ready to prove the strict separation result.

Proof of Lemma 3.10. Assume that $D(\Pi, \Psi) = \inf_{r\in\Pi, y\in\Psi} |r - y| = 0$. Consider a sequence $\epsilon_n \to 0$ as n tends to ∞. For each ϵ_n, construct signals $q^{(n)}$ with a bounded support on $[t_n, t_{n+1}]$, and $\tilde{\pi}^{(n)}$ according to Lemma 3.11. Define

$$\tilde{\pi} = \sum_{n=1}^{\infty} \tilde{\pi}^{(n)}\Gamma_{[t_n,t_{n+1}]}.$$

We have

$$\tilde{\pi}Gq^{(n)} = \tilde{\pi}^{(n)}\Gamma_{[t_n,t_{n+1}]}Gq^{(n)} + \tilde{\pi}(I - \Gamma_{[t_n,t_{n+1}]})Gq^{(n)},$$

and

$$\begin{aligned}\|(I - \pi G)q^{(n)}\| &\leq \|(I - \tilde{\pi}^{(n)}\Gamma_{[t_n,t_{n+1}]}G)q^{(n)}\| + \|(I - \Gamma_{[t_n,t_{n+1}]})Gq^{(n)}\| \\ &\leq C\epsilon_n + \epsilon_n^2\end{aligned}$$

Because $\epsilon_n \to 0$, the right-hand side approaches 0, and so does the left-hand side. Therefore, since $\|q^{(n)}\| = 1$, the mapping $I - \tilde{\pi}G$ cannot be invertible, which contradicts the robust stability assumption. This implies that Π and Ψ are strictly separable. □

To draw the connection to the SDP problem (3.14), observe that

$$\phi_{ij}(x,q) = \left\langle \begin{bmatrix} x \\ q \end{bmatrix}, M_\pi^{ij} \begin{bmatrix} x \\ q \end{bmatrix} \right\rangle, \tag{3.34}$$

where

$$[M_\pi^{ij}]_{kl} = \begin{cases} \bar{c}_{ij}^2 - c_{ij}^2 & (k,l) = (j,j) \\ c_{ij} & (k,l) = (i, i*n+j) \text{ or } (i*n+j, i) \\ -1 & (k,l) = (i*n+j, i*n+j) \end{cases},$$

and $M(\lambda;\xi) = \sum_{i\in[n_a], j\in[n_s]} \lambda_{ij} M_\pi^{ij}$ as defined in Lemma 3.2.

PROPOSITION 3.12. *The SDP condition (3.14) is feasible if and only if there exist multipliers $\lambda_{ij} \geq 0$ and $\epsilon > 0$ such that*

$$\sum_{i\in[n_a], j\in[n_s]} \lambda_{ij}\phi_{ij}(x,q) \leq -\epsilon\|q\|^2 \tag{3.35}$$

for all $q \in L^{n_a n_s}$ and $x = Gq$.

Proof. Since $\phi_{ij}(x,q) = \left\langle \begin{bmatrix} x \\ q \end{bmatrix}, M_\pi^{ij} \begin{bmatrix} x \\ q \end{bmatrix} \right\rangle$, the condition (3.35) is equivalent to

$$\begin{bmatrix} G \\ I \end{bmatrix}^* M(\lambda;\xi) \begin{bmatrix} G \\ I \end{bmatrix} \prec 0. \tag{3.36}$$

By the KYP lemma, this is equivalent to the existence of $P \succeq 0$ such that:

$$\begin{bmatrix} A^\top P + PA & PBW \\ W^\top B^\top P & 0 \end{bmatrix} + M(\lambda;\xi) \prec 0. \tag{3.37}$$

By Schur complement, P satisfies the KYP condition if and only if it satisfies (3.14). Thus, the claim is shown. □

THEOREM 3.13. *Let $\tilde{\pi} : L^{n_s} \to L^{n_a n_s}$ be a bounded causal controller such that $\tilde{\pi} \in \tilde{\mathcal{P}}(\xi)$. Assume that the interconnection of G and $\tilde{\pi}$ is well-posed. Then, the input-output stability of the feedback interconnection of system* (3.23) *implies that there exist $P \succeq 0$, $\gamma > 0$ and $\lambda \geq 0$ such that SDP$(P, \lambda, \gamma, \xi)$ in* (3.14) *is feasible.*

Proof. Since the system is input-output stable, the sets Π and $\overline{\Psi}$ are strictly separable due to Lemma 3.10. Since both Π and $\overline{\Psi}$ are convex (Lemma 3.9), there exist a strictly separating hyperplane parametrized by $\lambda \in \mathbb{R}^{mn}$ and scalars α, β, such that

$$\langle \lambda, \phi \rangle \leq \alpha < \beta \leq \langle \lambda, y \rangle$$

for all $\phi \in \overline{\Psi}$ and $y \in \Pi$. Since $\langle \lambda, y \rangle$ is bounded from below, we must have $\lambda \geq 0$, and without loss of generality, we can set $\beta = 0$ and $\alpha < 0$. This condition is equivalent to (3.35), and by Proposition 3.12, this implies that the SDP condition is feasible. □

4. Numerical examples.

In this section, we empirically study the stability-certified reinforcement learning in real-world problems such as flight formation [23] and power grid frequency regulation [18]. Designing an optimal controller for these systems is challenging, because they consist of many interconnected subsystems that have limited information sharing, and also their underlying models are typically nonlinear and even time-varying and uncertain. Indeed, for the case of decentralized control, which aims at designing a set of local controllers whose interactions are specified by physical and informational structures, it has been long known that it amounts to an NP-hard optimization problem in general [8]. End-to-end reinforcement learning comes in handy, because it does not require model information by simply interacting with the environment while collecting rewards.

In a multi-agent setting, each agent explores and learns its own policy independently without knowing about other agents' policies [12]. For the simplicity of implementation, we consider the synchronous and cooperative scenario, where agents conduct an action at each time step and observe the reward for the whole system. Their goal is to collectively maximize the rewards (or minimize the costs) shared equally among them. The present analysis aims at *offering safety certificates of existing RL algorithms when applied to real-world dynamical systems*, by simply monitoring the gradients information of the neural network policy. This is orthogonal to the line of research that aims at improving the performance of the existing RL algorithms. The examples are taken from [23, 18, 17], but we deal directly with the underlying *nonlinear* physics rather than a linearized model.

4.1. Multi-agent flight formation. Consider the multi-agent flight formation problem [23], where each agent can only observe the relative distance from its neighbors, as illustrated in Figure 2. The goal is to design a local controller for each aircraft such a predefined pattern is formed as efficiently as possible. The physical model[3] for each aircraft is given by:

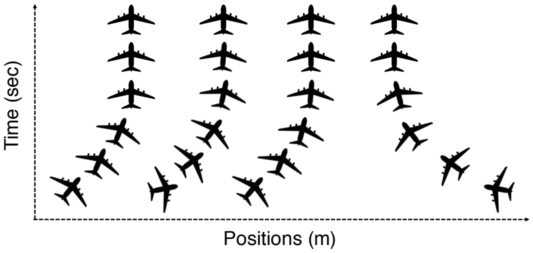


Fig. 2: Illustration of the multi-agent flight formation problem.

$$\begin{aligned}\ddot{z}^i(t) &= v^i(t)\\ \ddot{\theta}^i(t) &= \frac{1}{\delta}\left(\sin\theta^i(t) + v^i(t)\right),\end{aligned}$$

where z^i and θ^i denote the horizontal position and angle of aircraft i, respectively, and $\delta > 0$ characterizes the physical coupling of rolling moment and lateral acceleration. To stabilize the system, a simple feedback rule is proposed in [6],

$$v^i(t) = \alpha\dot{z}^i(t) + \beta\theta^i(t) + \gamma\dot{\theta}^i(t) + u^i(t) \tag{4.1}$$

where the parameters of the first three terms are designed to maintain the internal stability of the horizontal speed and angle of each aircraft (specifically, $\alpha = 90.62$, $\beta = -42.15$, $\gamma = -13.22$, $\delta = 0.1$ as explained in [6]), and the last term is an external input optimized for performance (e.g., to move the aircraft to a target state as fast as possible). For each agent, by defining the state $x^i(t) = \begin{bmatrix}\dot{z}^i(t) & \theta^i(t) & \dot{\theta}(t)\end{bmatrix}^\top$, the above dynamics can be written as

$$\dot{x}^i(t) = \underbrace{\begin{bmatrix}\alpha & \beta & \gamma\\ 0 & 0 & 1\\ \frac{\alpha}{\delta} & \frac{\beta+1}{\delta} & \frac{\gamma}{\delta}\end{bmatrix}}_{A^i} x^i(t) + \underbrace{\begin{bmatrix}1\\ 0\\ \frac{1}{\delta}\end{bmatrix}}_{B^i} u^i(t) + \underbrace{\begin{bmatrix}0\\ 0\\ \sin\theta^i(t) - \theta^i(t)\end{bmatrix}}_{g^i(x^i(t))}, \tag{4.2}$$

where $g^i(x^i(t))$ is a nonlinear function of $x^i(t)$ that is neglected for a linearized model [6, 17]. In a distributed control setting, each agent only has access to the relative distance from its neighbors; therefore, for agents $i = 1, 2, 3$, define

$$\widetilde{x}^i(t) = \begin{bmatrix}z^i(t) - z^{i+1}(t) - d & x^i(t)^\top\end{bmatrix}^\top, \tag{4.3}$$

[3] The cosine term in the original formulation is omitted for simplicity, though it can be incorporated in a more comprehensive treatment.

where d is the desired distance between agents. The state-space model of the interconnected system can be written in the form of (1.1):

(4.4)
$$\begin{bmatrix} \dot{\tilde{x}}^1 \\ \dot{\tilde{x}}^2 \\ \dot{\tilde{x}}^3 \\ \dot{x}^4 \end{bmatrix} = \underbrace{\begin{bmatrix} \widetilde{A}^1 & H_4 & 0 & 0 \\ 0 & \widetilde{A}^2 & H_4 & 0 \\ 0 & 0 & \widetilde{A}^3 & H_3 \\ 0 & 0 & 0 & A^4 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \widetilde{x}^1 \\ \widetilde{x}^2 \\ \widetilde{x}^3 \\ x^4 \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} \widetilde{B}^1 & 0 & 0 & 0 \\ 0 & \widetilde{B}^2 & 0 & 0 \\ 0 & 0 & \widetilde{B}^3 & 0 \\ 0 & 0 & 0 & B^4 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} u^1 \\ u^2 \\ u^3 \\ u^4 \end{bmatrix}}_{u(t)} + \underbrace{\begin{bmatrix} \widetilde{g}^1(x^1) \\ \widetilde{g}^2(x^2) \\ \widetilde{g}^3(x^3) \\ g^4(x^4) \end{bmatrix}}_{g(x(t))},$$

where H_3 (or H_4) is a 4×3 (or 4×4) matrix whose $(i,j)^{\text{th}}$ entry is equal to -1 if $(i,j) = (1,1)$ (or $(i,j) = (1,2)$) and is zero otherwise, and where $\widetilde{A}^i$, $\widetilde{B}^i$ and $\widetilde{g}^i(x^i(t))$ for $i = 1,2,3$ are augmented to account for the state of relative positions, given by

$$\widetilde{A}^i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \alpha & \beta & \gamma \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\alpha}{\delta} & \frac{\beta+1}{\delta} & \frac{\gamma}{\delta} \end{bmatrix}, \quad \widetilde{B}^i = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \frac{1}{\delta} \end{bmatrix}, \quad \widetilde{g}^i(x^i(t)) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sin\theta^i(t) - \theta^i(t) \end{bmatrix}. \tag{4.5}$$

One particular strength of RL is that the reward function can be highly nonconvex, nonlinear, and arbitrarily designed; however, since quadratic costs are widely used in the control literature, consider the case $r(x(t), u(t)) = x(t)^\top Q x(t) + u(t)^\top R u(t)$. For the following experiments, assume that $Q = 1000 \times I_{15}$ and $R = I_4$. In addition, because the original system A has its largest eigenvalue at 0, we need a nominal distributed linear controller K_d, whose primary goal is to make the largest eigenvalue of $A + BK_n$ negative. Such controller could be designed using methods such as robust control synthesis for the linearized system [16, 55]. With the nominal controller in place, we can define the new system matrix $A_G = A + BK_n$ and replace A in (4.4).

The task for multi-agent RL is to learn the controller $u^i(t)$, which only takes inputs of the relative distances of agent i to its neighbors. For example, agent 1 can only observe $z^1(t) - z^2(t) - d$ (i.e., the 1st entry of $x(t)$); similarly, agent 2 can only observe $z^1(t) - z^2(t) - d$ and $z^2(t) - z^3(t) - d$ (i.e., the 1st and 5th entries of $x(t)$).

Stability certificate: To obtain the stability certificate of (4.4), we apply the method in Section 3.3. The nonzero entries of the nonlinear component $g(x(t))$ are in the form of $\sin(\theta) - \theta$, which can be treated as an uncertainty block with the slope restricted to $[-1, 0]$ for $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$; therefore, the Zames-Falb IQCs can be employed to construct (3.19) [53, 29]. As for the RL agents u^i, their gradient bounds can be certified according to Theorem 3.4. Specifically, we assume that each agent u^i is l-Lipschitz continuous, and solve (3.21) for a given set of γ and l. The certified gradient bounds (Lipschitz constants) are plotted in Figure 3 using different constraints. The conservative L_2 constraint (3.3) is only able to certify stability for Lipschitz constants up to 0.8. By incorporating the sparsity of distributed controller, we can increase the margin to 1.2, which is satisfied throughout the learning process.

In order to further increase the set of certifiable stable controllers, we monitor the partial gradient information for each agent and encode them as non-homogeneous gradient bounds. For instance, if $\frac{\partial \pi_i(x)}{\partial x_j}$ has been consistently positive for latest iterations, we will set $\overline{\xi}_{ij} = l$ and $\underline{\xi}_{ij} = -\epsilon l$, where $\epsilon > 0$ is a small margin, such as 0.1, to allow explorations. By performing this during learning, it would be possible to significantly enlarge the certified Lipschitz bound to up to 2.5, as shown in Figure 3.

Policy gradient RL: To perform multi-agent reinforcement learning, we employ trust region policy optimization with natural gradients and smoothness policies. During

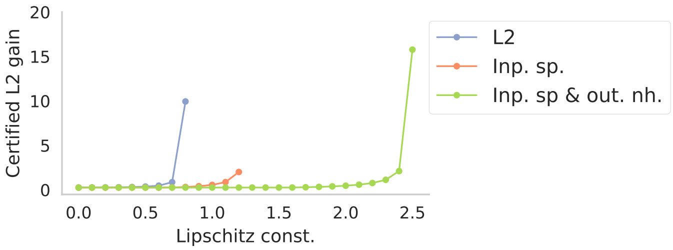


Fig. 3: Stability-certified Lipschitz constants obtained by the standard L_2 bound (L2) in (3.3) and the method proposed in Lemma 3.2, which considers input sparsity (inp. sp.) and output non-homogeneity (out. nh.).

learning, we employ the hard-thresholding step introduced in Section 2.1 to ensure that the gradient bounds are satisfied. The trajectories of rewards averaged over three independent experiments are shown in Figure 4. In this example, agents with a 1-layer neural network (each with 5 hidden units) can learn most efficiently when employed with the smoothness penalties (coefficients are set to be $\omega_1 = \omega_2 = 0.01$) in (2.8). Without the guidance of these penalties, the linear controller and 1-layer neural network apparently cannot effectively explore the parameter space.

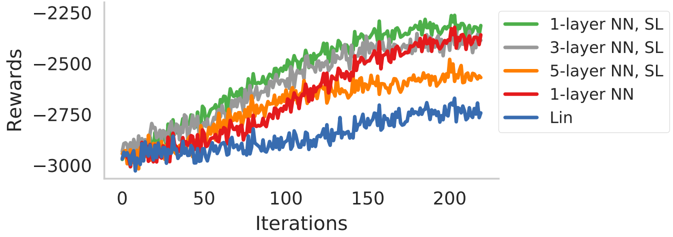
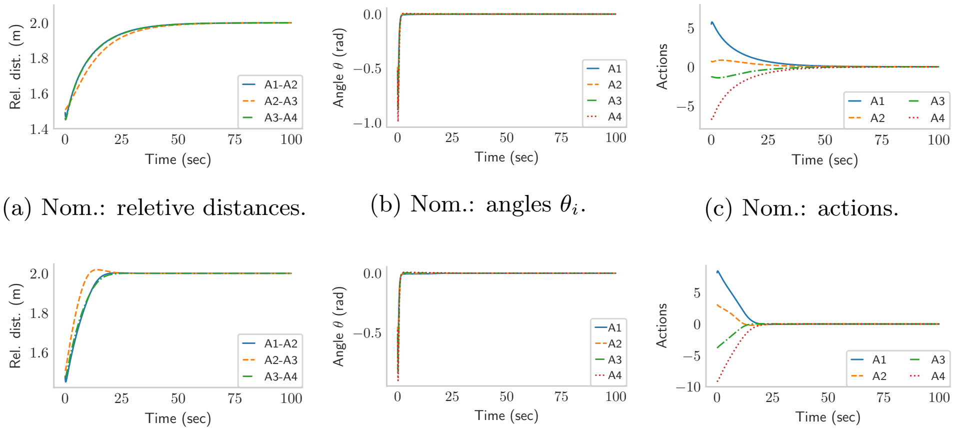


Fig. 4: Learning performance of different control structures (1-layer neural network, 5-layer neural network, and linear controller). By the inclusion of a smoothness loss (SL) in the learning objective (2.8), the exploration becomes more effective.

The learned 5-layer neural network policy is employed in an actual control task, as shown in Figure 5. Compared to the nominal controller, the flights can be maneuvered more efficiently in this case with only local information. In terms of the actual cost, the RL agents achieve the cost 41.0, which is about 30% lower than that of the nominal controller (58.3). This result can be examined both in the actual state-action trajectories in Figure 5 or the control behaviors in Figure 6. The results indicate that RL is able to improve a given controller when the underlying system is nonlinear and unknown.

4.2. Power system frequency regulation. In this case study, we focus on the problem of distributed control for power system frequency regulation [18]. The IEEE 39-Bus New England Power System under analysis is shown in Figure 7. In a distributed control setting, each generator can only share its rotor angle and frequency information with a pre-specified set of counterparts that are geographically distributed.



(a) Nom.: reletive distances.

(b) Nom.: angles θ_i.

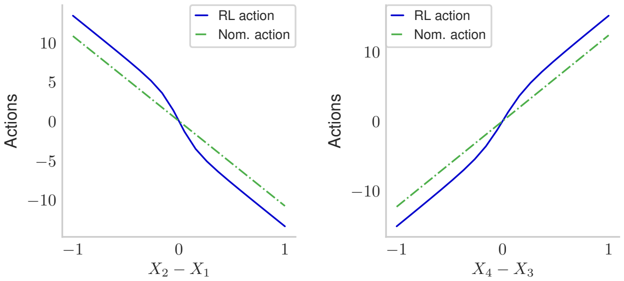
(c) Nom.: actions.

(d) NN: relative distances.

(e) NN: angles θ_i.

(f) NN: actions.

Fig. 5: State and action trajectories in a typical contral task, where the nominal controller (Nom) and the RL agents achieve costs of 58.3 and 41.0, respectively.



(a) Controller of Agent 1.

(b) Controller of Agent 4.

Fig. 6: Demonstration of control outputs for the nominal action and RL agents.

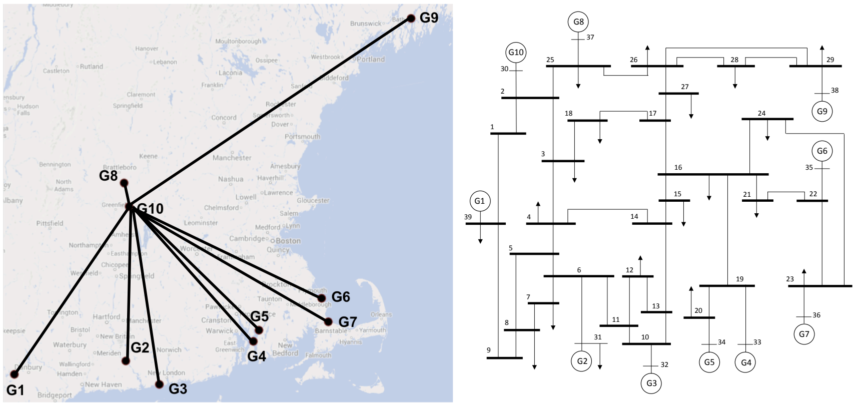
The main goal is to optimally adjust the mechanical power input to each generator such that the phase and frequency at each bus can be restored to their nominal values after a possible perturbation. Let θ_i denote the voltage angle at a generator bus i (in rad). The physics of power systems are modeled by the per-unit swing equation:

$$m_i\ddot{\theta}_i + d_i\dot{\theta} = p_{m_i} - p_{e_i} \tag{4.6}$$

where p_{m_i} is the mechanical power input to the generator at bus i (in p.u.), p_{e_i} is the electrical active power injection at bus i (in p.u.), m_i is the inertia coefficient of the generator at bus i (in p.u.-sec^2/rad), and d_i is the damping coefficient of the generator at bus i (in p.u.-sec/rad). The electrical real power injection p_{e_i} depends on the voltage angle difference in a nonlinear way, as governed by the AC power flow equation:

$$p_{e_i} = \sum_{j=1}^{n} |v_i||v_j| \left(g_{ij} \cos(\theta_i - \theta_j) + b_{ij} \sin(\theta_i - \theta_j) \right) \tag{4.7}$$

where n is the number of buses in the system, g_{ij} and b_{ij} are the conductance and susceptance of the transmission line that connects buses i and j, v_i is the voltage phasor at bus i, and $|v_i|$ is its voltage magnitude. Because the conductance g_{ij} is typically several orders of magnitude smaller than the susceptance b_{ij}, for the simplicity of mathematical treatment, we omit the cosine term and only keep the sine term that accounts for the majority of nonlinearity. Each generator needs to make decisions on the value of the mechanical power p_{m_i} to inject in order to maintain the stability of the power system.



(a) Star-connected structure.

(b) IEEE 39-bus system.

Fig. 7: Illustration of the frequency regulation problem for the New England power system. The communication among generators follows a star topology.

Let the rotor angles and the frequency states be denoted as $\theta = \begin{bmatrix} \theta_1 & \cdots & \theta_n \end{bmatrix}^\top$ and $\omega = \begin{bmatrix} \omega_1 & \cdots & \omega_n \end{bmatrix}^\top$, and the generator mechanical power injections be denoted as $p_m = \begin{bmatrix} p_{m_1} & \cdots & p_{m_n} \end{bmatrix}^\top$. Then, the state-space representation of the nonlinear system is given by:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ -M^{-1}L & -M^{-1}D \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \theta \\ \omega \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ M^{-1} \end{bmatrix}}_{B} p_m + \underbrace{\begin{bmatrix} \mathbf{0} \\ g(\theta) \end{bmatrix}}_{g(x)} \tag{4.8}$$

where $g(\theta) = \begin{bmatrix} g_1(\theta) & \cdots & g_n(\theta) \end{bmatrix}^\top$ with $g_i(\theta) = \sum_{j=1}^n \frac{b_{ij}}{m_j} \left((\theta_i - \theta_j) - \sin(\theta_i - \theta_j) \right)$, and $M = \operatorname{diag}\left(\{m_i\}_{i=1}^n \right)$, $D = \operatorname{diag}\left(\{d_i\}_{i=1}^n \right)$, and L is a Laplacian matrix whose entries are specified in [18, Sec. IV-B]. For linearization (also known as DC approximation), the nonlinear part $g(x)$ is assumed to be zero when the phase differences are small [18, 17]. On the contrary, we deal with this term in the stability certification to demonstrate its capability of producing non-conservative results even for nonlinear systems. Similar to the flight formation case, we assume that there exists a distributed nominal controller that stablizes the system. To conduct multi-agent RL, each controller p_{m_i} is a neural network that takes the available phases and frequencies as the input and determines the mechanical power injection at bus i. The main focus is to study the certified-gradient bounds for each agent policy in this large-scale setting.

Stability certificate: Similar to the flight formation problem, the nonlinearities in $\mathbf{g}(x)$ are in the form of $\Delta\theta_{ij} - \sin\Delta\theta_{ij}$, where $\Delta\theta_{ij} = \theta_i - \theta_j$ represents the phase difference, which has its slope restricted to $[0, 1 - \cos(\overline{\theta})]$ for every $\Delta\theta_{ij} \in [-\overline{\theta}, \overline{\theta}]$ and thus can be treated using the Zames-Falb IQC. In the smoothness margin analysis, assume that $\overline{\theta} = \frac{\pi}{3}$, which requires the phase angle difference to be within $[-\frac{\pi}{3}, \frac{\pi}{3}]$. This is a large set of uncertainties that includes both normal and abnormal operational conditions. To study the stability of the multi-agent policies, we adopt a black-box approach by simply considering the input-output constraint. By simply applying the L_2 constraint in (3.3), we can only certify stability for Lipschitz constants up to 0.4, as shown in Figure 8. Because the distributed control is sparse, we can leverage it by setting the lower and upper bounds $\underline{\xi}_{ij} = \overline{\xi}_{ij} = 0$ for each agent i that does not utilize observation j, and $\overline{\xi}_{ij} = -\underline{\xi}_{ij} = l$ otherwise, where l is the Lipschitz constant to be certified. This information can be encoded in $\underline{\text{SDP}}(P, \lambda, \gamma, \xi)$ in (3.21), which can be solved for L up to 0.6 (doubling the certificate provided by the L_2 constraint).

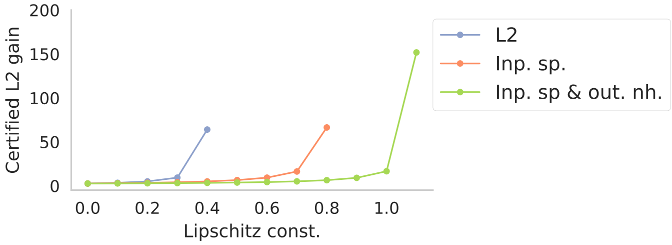
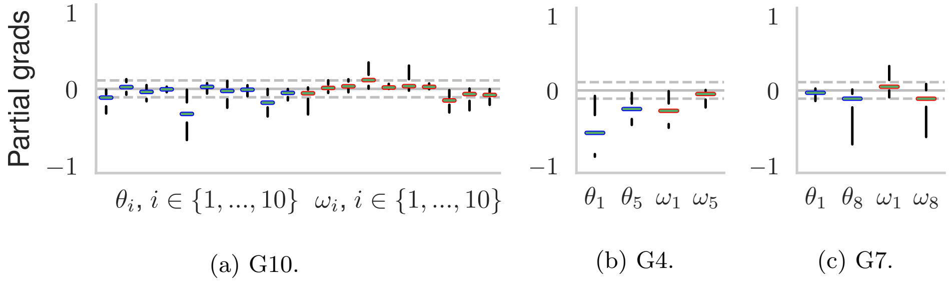


Fig. 8: Certified Lipschitz constants for the power system regulation task.

Due to the problem nature, we further observe that for each agent, the partial gradient of the policy with respect to certain observations is primarily one-sided, as shown in Figure 9. With a band of ±0.1, the partial gradients remain within either $[-0.1, 1]$ or $[-1, 0.1]$ throughout the learning process. This information is gleaned during the learning phase, and we can incorporate it into the partial gradient bounds (e.g., $\overline{\xi}_{ij} = -0.1l$ and $\underline{\xi}_{ij} = l$ for agent i which exhibits positive gradient with respect to observation j) to extend the certificate up to 1.1.

Policy gradient RL: Similar to the flight formation task, we perform multi-agent policy gradient RL. The learned neural network controller is implemented in a typical control case, whose trajectories are shown in Figure 10. As can be seen, the RL policies can regulate the frequencies more efficiently than the nominal controller, with a significantly lowered cost (50.8 vs. 23.9). More importantly, we compare the cases of RL with and without regulating the Lipschitz constants in Figure 11. Without regulating the gradients, the RL is able to reach a performance slightly higher than its stability-certified counterpart. However, after about iteration 500, the performance starts to deteriorate (due to a possibly large gradient variance and high sensitivity to step size) until it completely loses the previous gains and starts to break the system.

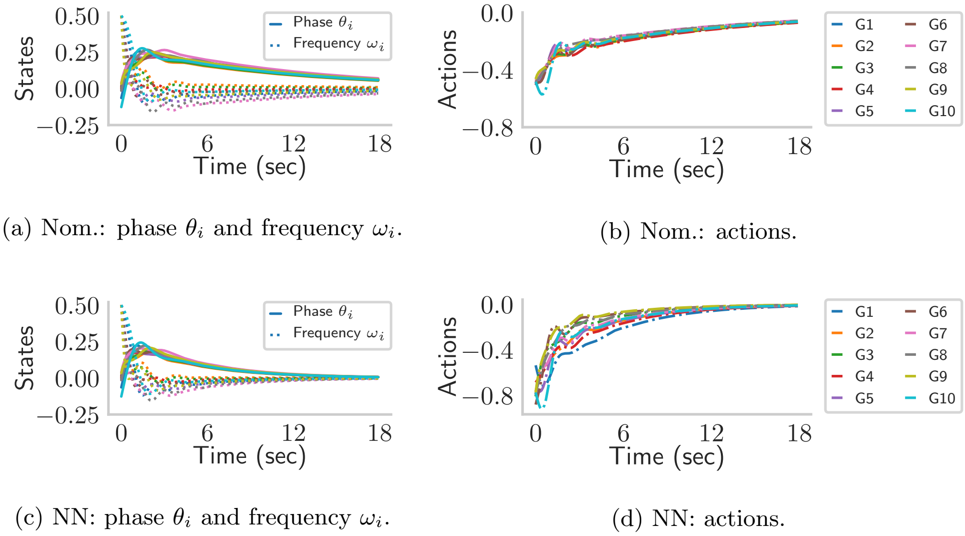


(a) G10.

(b) G4.

(c) G7.

Fig. 9: Box plots of partial gradients of individual generators (G10, G4, G7) with respect to local information. Grey dashed lines indicate ±0.1.



(a) Nom.: phase θ_i and frequency ω_i.

(b) Nom.: actions.

(c) NN: phase θ_i and frequency ω_i.

(d) NN: actions.

Fig. 10: State and action trajectories of the nominal and neural network controllers for power system frequency regulation, with costs of 50.8 and 23.9, respectively.

This intolerable behavior is due to the large Lipschitz gains that grow unboundedly, as shown in Figure 12. In comparison, RL with regulated gradient bounds is able to make a substantial improvement, and also exhibits a more stable behavior.

5. Conclusions. In this paper, we focused on the challenging task of ensuring the stability of reinforcement learning in real-world dynamical systems. By solving the proposed SDP feasibility problem, we can offer a preventative certificate of stability for a broad class of neural network controllers with bounded gradients. Furthermore, we analyzed the (non)conservatism of the certificate, which was demonstrated in the empirical investigation of decentralized nonlinear control tasks, including multi-agent flight formation and power grid frequency regulation. Results indicated that the set of stability-certified controllers was significantly larger than what the existing approaches

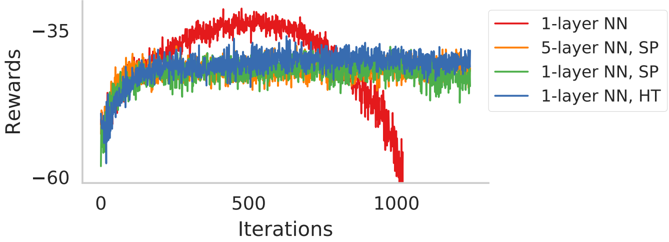
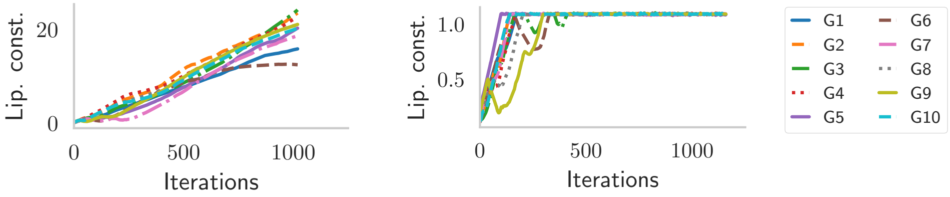


Fig. 11: Long-term performance of RL for agents with regulated gradients by soft penalty (SP), which adaptively adjusts the coefficients ω_2 in (2.8), and hard thresholding (HT), which shrinks the network last layer to satisfy the gradient bounds. The RL agents without regulating the gradients exhibit "dangerous" behaviors in the long run.



(a) No gradient regulation.

(b) Gradient regulation.

Fig. 12: Trajectories of Lipschitz constants with and without regulation.

can offer, and that the RL agents can substantially improve the performance of nominal controllers while staying within the safe set. Most importantly, regulation of gradient bounds was able to improve on-policy learning stability and avoid "catastropic" effects caused by the unregulated high gains. The present study represents a key step towards safe deployment of reinforcement learning in mission-critical real-world systems.

REFERENCES

[1] J. Achiam, D. Held, A. Tamar, and P. Abbeel, *Constrained policy optimization*, in Proc. of the International Conference on Machine Learning, 2017, pp. 22–31.

[2] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, *Reachability-based safe learning with Gaussian processes*, in Proc. of the IEEE Conference on Decision and Control, 2014, pp. 1424–1431.

[3] S.-I. Amari, *Natural gradient works efficiently in learning*, Neural computation, 10 (1998), pp. 251–276.

[4] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, *Concrete problems in AI safety*, arXiv preprint arXiv:1606.06565, (2016).

[5] C. W. Anderson, P. M. Young, M. R. Buehner, J. N. Knight, K. A. Bush, and D. C. Hittle, *Robust reinforcement learning control using integral quadratic constraints for recurrent neural networks*, IEEE Transactions on Neural Networks, 18 (2007), pp. 993–1002.

[6] M. Arcak, *Synchronization and pattern formation in diffusively coupled systems*, in Proc. of the IEEE Conference on Decision and Control, 2012, pp. 7184–7192.

[7] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, *Provably safe and robust learning-based model predictive control*, Automatica, 49 (2013), pp. 1216–1226.

[8] L. Bakule, *Decentralized control: An overview*, Annual reviews in control, 32 (2008), pp. 87–98.

[9] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, *Safe model-based reinforcement learning with stability guarantees*, in Advances in Neural Information Processing Systems, 2017, pp. 908–919.

[10] R. Bobiti and M. Lazar, *A sampling approach to finding lyapunov functions for nonlinear discrete-time systems*, in Proc. of the IEEE European Control Conference, 2016, pp. 561–566.

[11] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, vol. 15, SIAM, 1994.

[12] L. Buşoniu, R. Babuška, and B. De Schutter, *Multi-agent reinforcement learning: An overview*, in Innovations in multi-agent systems and applications-1, Springer, 2010, pp. 183–221.

[13] F. H. Clarke, *Optimization and nonsmooth analysis*, vol. 5, SIAM, 1990.

[14] S. P. Coraluppi and S. I. Marcus, *Risk-sensitive and minimax control of discrete-time, finite-state Markov decision processes*, Automatica, 35 (1999), pp. 301–309.

[15] H. Drucker and Y. Le Cun, *Improving generalization performance using double backpropagation*, IEEE Transactions on Neural Networks, 3 (1992), pp. 991–997.

[16] G. E. Dullerud and F. Paganini, *A course in robust control theory: a convex approach*, vol. 36, Springer Science & Business Media, 2013.

[17] S. Fattahi, G. Fazelnia, J. Lavaei, and M. Arcak, *Transformation of optimal centralized controllers into near-globally optimal static distributed controllers*, IEEE Transactions on Automatic Control, (2018), pp. 1–1, https://doi.org/10.1109/TAC.2018.2829473.

[18] G. Fazelnia, R. Madani, A. Kalbat, and J. Lavaei, *Convex relaxation for optimal distributed control problems*, IEEE Transactions on Automatic Control, 62 (2017), pp. 206–221.

[19] J. M. Fry, M. Farhood, and P. Seiler, *IQC-based robustness analysis of discrete-time linear time-varying systems*, International Journal of Robust and Nonlinear Control, 27 (2017), pp. 3135–3157.

[20] J. Garcia and F. Fernández, *A comprehensive survey on safe reinforcement learning*, Journal of Machine Learning Research, 16 (2015), pp. 1437–1480.

[21] P. Geibel and F. Wysotzki, *Risk-sensitive reinforcement learning applied to control under constraints*, Journal of Artificial Intelligence Research, 24 (2005), pp. 81–108.

[22] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, *Improved training of wasserstein gans*, in Advances in Neural Information Processing Systems, 2017, pp. 5767–5777.

[23] J. Hauser, S. Sastry, and G. Meyer, *Nonlinear control design for slightly non-minimum phase systems: Application to v/stol aircraft*, Automatica, 28 (1992), pp. 665–679.

[24] W. P. Heath and A. G. Wills, *Zames-Falb multipliers for quadratic programming*, in Proc. of the IEEE Conference on Decision and Control, 2005, pp. 963–968.

[25] S. Kakade and J. Langford, *Approximately optimal approximate reinforcement learning*, in Proc. of the International Conference on Machine Learning, 2002, pp. 267–274.

[26] S. M. Kakade, *A natural policy gradient*, in Advances in neural information processing systems, 2002, pp. 1531–1538.

[27] H. K. Khalil, *Noninear systems*, Prentice-Hall, New Jersey, 2 (1996), pp. 5–1.

[28] R. M. Kretchmara, P. M. Young, C. W. Anderson, D. C. Hittle, M. L. Anderson, and C. Delnero, *Robust reinforcement learning control*, in Proc. of the IEEE American Control Conference, vol. 2, 2001, pp. 902–907.

[29] L. Lessard, B. Recht, and A. Packard, *Analysis and design of optimization algorithms via integral quadratic constraints*, SIAM Journal on Optimization, 26 (2016), pp. 57–95.

[30] S. Levine and P. Abbeel, *Learning neural network policies with guided policy search under unknown dynamics*, in Advances in Neural Information Processing Systems, 2014, pp. 1071–1079.

[31] Y. Li, *Deep reinforcement learning: An overview*, arXiv preprint arXiv:1701.07274, (2017).

[32] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, *Continuous control with deep reinforcement learning*, arXiv preprint arXiv:1509.02971, (2015).

[33] A. Megretski and A. Rantzer, *System analysis via integral quadratic constraints*, IEEE Transactions on Automatic Control, 42 (1997), pp. 819–830.

[34] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, *Asynchronous methods for deep reinforcement learning*, in Proc. of the International Conference on Machine Learning, 2016, pp. 1928–1937.

[35] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, *Playing atari with deep reinforcement learning*, arXiv preprint arXiv:1312.5602, (2013).

[36] T. M. Moldovan and P. Abbeel, *Safe exploration in Markov decision processes*, in Proc. of the International Conference on Machine Learning, 2012, pp. 1451–1458.

[37] A. G. Ororbia II, D. Kifer, and C. L. Giles, *Unifying adversarial training algorithms with data gradient regularization*, Neural computation, 29 (2017), pp. 867–887.

[38] A. Packard and J. Doyle, *The complex structured singular value*, Automatica, 29 (1993), pp. 71–109.

[39] T. J. Perkins and A. G. Barto, *Lyapunov design for safe reinforcement learning*, Journal of Machine Learning Research, 3 (2002), pp. 803–832.

[40] M. G. Safonov and V. V. Kulkarni, *Zames-Falb multipliers for MIMO nonlinearities*, in Proc. of the American Control Conference, vol. 6, 2000, pp. 4144–4148.

[41] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, *Trust region policy optimization*, in Proc. of the International Conference on Machine Learning, 2015, pp. 1889–1897.

[42] P. Seiler, *Stability analysis with dissipation inequalities and integral quadratic constraints*, IEEE Transactions on Automatic Control, 60 (2015), pp. 1704–1709.

[43] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., *Mastering the game of go with deep neural networks and tree search*, Nature, 529 (2016), p. 484.

[44] I. Stoica, D. Song, R. A. Popa, D. Patterson, M. W. Mahoney, R. Katz, A. D. Joseph, M. Jordan, J. M. Hellerstein, J. E. Gonzalez, et al., *A Berkeley view of systems challenges for AI*, arXiv preprint arXiv:1712.05855, (2017).

[45] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, *Safe exploration for optimization with Gaussian processes*, in Proc. of the International Conference on Machine Learning, 2015, pp. 997–1005.

[46] R. S. Sutton, *Integrated architecture for learning, planning, and reacting based on approximating dynamic programming*, in Proc. of the International Conference on Machine Learning, 1990, pp. 216–224.

[47] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1, MIT press Cambridge, 1998.

[48] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, *Intriguing properties of neural networks*, in Proc. of the International Conference on Learning Representations, 2014.

[49] C. Watkins and P. Dayan, *Q-learning*, Machine learning, 8 (1992), pp. 279–292.

[50] W. Wiesemann, D. Kuhn, and B. Rustem, *Robust Markov decision processes*, Mathematics of Operations Research, 38 (2013), pp. 153–183.

[51] J. C. Willems, *Dissipative dynamical systems, Part II: Linear systems with quadratic supply rates*, Archive for rational mechanics and analysis, 45 (1972), pp. 352–393.

[52] R. J. Williams, *Simple statistical gradient-following algorithms for connectionist reinforcement learning*, Machine learning, 8 (1992), pp. 229–256.

[53] G. Zames and P. Falb, *Stability conditions for systems with monotone and slope-restricted nonlinearities*, SIAM Journal on Control, 6 (1968), pp. 89–108.

[54] A. Zemouche and M. Boutayeb, *On LMI conditions to design observers for lipschitz nonlinear systems*, Automatica, 49 (2013), pp. 585–591.

[55] K. Zhou, J. C. Doyle, K. Glover, et al., *Robust and optimal control*, vol. 40, Prentice hall New Jersey, 1996.