Control-Theoretic Analysis of Smoothness for Stability-Certified Reinforcement Learning

Ming Jin and Javad Lavaei

***Abstract*— It is critical to obtain stability certificate before deploying reinforcement learning in real-world mission-critical systems. This study justifies the intuition that smoothness (i.e., small changes in inputs lead to small changes in outputs) is an important property for stability-certified reinforcement learning from a control-theoretic perspective. The smoothness margin can be obtained by solving a feasibility problem based on semi-definite programming for both linear and nonlinear dynamical systems, and it does not need to access the exact parameters of the learned controllers. Numerical evaluation on nonlinear and decentralized frequency control for large-scale power grids demonstrates that the smoothness margin can certify stability during both exploration and deployment for (deep) neural-network policies, which substantially surpass nominal controllers in performance. The study opens up new opportunities for robust Lipschitz continuous policy learning.**

I. Introduction

Reinforcement learning (RL) is a powerful tool for real-world control, which aims at guiding an agent to perform a task as efficiently and skillfully as possible through interactions with the environment [1], [2]. This work investigates the important role of *smoothness* to certify stability for neural-network based reinforcement learning when deployed in real-world control tasks (illustrated in Fig. 1). Consider a deterministic, continuous-time dynamical system $\dot{\mathbf{x}}(t) = \mathbf{f}_t(\mathbf{x}(t), \mathbf{u}(t))$, with the state $\mathbf{x}(t) \in \mathbb{R}^n$ and the control action $\mathbf{u}(t) \in \mathbb{R}^m$. In general, $\mathbf{f}_t$ can be a time-varying, nonlinear function, but for the purpose of stability analysis, we focus on the important case

$$\mathbf{f}_t(\mathbf{x}, \mathbf{u}) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{g}_t(\mathbf{x}(t)), \tag{1}$$

where $\mathbf{f}_t$ comprises of a linear time-invariant (LTI) component $\mathbf{A} \in \mathbb{R}^{n\times n}$, a control matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$, and a slowly time-varying component $\mathbf{g}_t$ that is allowed to be nonlinear and even uncertain.[1] For feedback control, we also allow the controller to obtain observations of the form $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \in \mathbb{R}^n$ that are a linear function of the state, where $\mathbf{C} \in \mathbb{R}^{n\times n}$ can have any prescribed sparsity pattern to account for partial observations in the context of decentralized control [3].

Suppose that $\mathbf{u}(t) = \boldsymbol{\pi}_t(\mathbf{y}(t)) + \mathbf{e}(t)$ is given by a neural network output with the exploration $\mathbf{e}(t)$ that has bounded energy over time ($\|\mathbf{e}\|_2^2 = \int \|\mathbf{e}(t)\|_2^2 dt \le \infty$). The neural network can be learned by a reinforcement learning agent to optimize some reward $r(\mathbf{x}, \mathbf{u})$ revealed through interactions with the environment. The main goal is to analyze the stability of the system under the policy $\boldsymbol{\pi}_t$ in the sense of finite $\mathcal{L}_2$ gain [4].

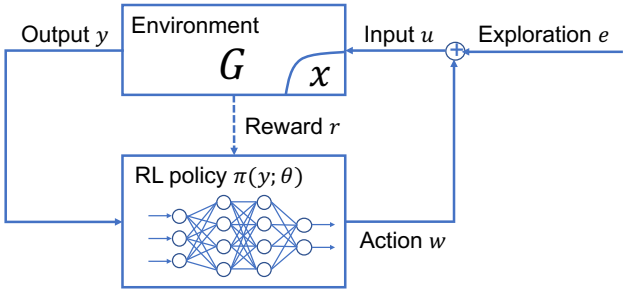


Fig. 1: End-to-end reinforcement learning in real-world dynamical system $\mathcal{G}$. The agent optimizes policy $\boldsymbol{\pi}(\mathbf{y})$ through exploration while receiving rewards r.

Definition 1.1 (Input-output stability): The $\mathcal{L}_2$ gain of the system $\mathcal{G}$ controlled by $\boldsymbol{\pi}$ is the worst-case ratio:

$$\gamma(\mathcal{G}, \boldsymbol{\pi}) = \sup_{\mathbf{e}\in\mathcal{L}_2} \frac{\|\mathbf{y}\|_2^2}{\|\mathbf{e}\|_2^2}, \tag{2}$$

where $\mathcal{L}_2$ is the set of square-summable signals, and $\mathbf{u}(t) = \boldsymbol{\pi}_t(\mathbf{y}(t)) + \mathbf{e}(t)$ is the control input with the exploration $\mathbf{e}(t)$. If $\gamma(\mathcal{G}, \boldsymbol{\pi}) < \infty$ is finite, then the interconnected system is said to have input-output stability (or finite $\mathcal{L}_2$ gain).

Let $L(\boldsymbol{\pi}_t)$ be the Lipschitz constant of $\boldsymbol{\pi}_t(\cdot)$ [5]. The main result of this paper can be stated as follows (a formal statement can be found in Theorem 4.4):

> *If there exists a constant L° such that the convex program SDP$(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L^\circ)$ defined in* (SDP-NL) *is numerically feasible, then the interconnected system (Fig. 1) is certifiably stable for any smooth controllers (i.e., $L(\boldsymbol{\pi}_t) \le L^\circ$).*

This theoretical result is based on the intuition that a real-life stable controlled system should be smooth, in the sense that small input changes lead to small output changes. To compute L°, we borrow powerful ideas from the framework of integral quadratic constraint (IQC) [6] and dissipation theory [7].

Even though IQC is celebrated for its non-conservativeness in robustness analysis, existing libraries for multi-input multi-output Lipschitz functions are very limited. One major obstacle is the derivation of non-trivial bounds on smoothness. To this end, we introduce *a new quadratic constraint* on

†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.

*M. Jin and J. Lavaei are with the Department of Industrial Engineering and Operations Research and the Tsinghua-Berkeley Shenzhen Institute, University of California, Berkeley. Emails: {jinming,lavaei}@berkeley.edu

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

smooth functions by exploiting the input sparsity and output non-homogeneity inherent in specific problems (Sec. IV-A). An overview of smooth reinforcement learning is provided in Sec. III. The method to compute stability-certified smoothness margin is presented in Sec. IV-B, which is evaluated in Sec. V for learning-based nonlinear decentralized control. The bounds are shown to be non-trivial and satisfied by performance-optimizing neural networks. Concluding remarks are provided in Sec. VI.

II. Related Work

This paper is closely related to the body of works on *safe reinforcement learning*, defined in [8] as "the process of learning policies that maximize performance in problems where safety is required during the learning and/or deployment." 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 [9], [10]. Robust MDP has been designed to maximize rewards while safely exploring the discrete state space [11], [12]. For continuous states and actions, robust MPC can be employed [13]. These methods require models for policy learning. Recently, *model-free policy optimization* has been successfully demonstrated in real-world tasks such as robotics, smart grid and transportation [2]. Existing approaches to guarantee safety are based on constraint satisfaction that holds with high probability [14].

The present analysis approaches safe reinforcement learning from a robust control perspective [4]. Lyapunov function and region of convergence have been widely used to analyze and verify stability when the system and its controller are known [4], [15]. Recently, learning-based Lyapunov stability verification has been employed for physical systems [16]. The main challenge of these methods is to find a suitable non-conservative Lyapunov function to conduct the analysis.

The framework of IQC has been widely used to analyze large-scale complex systems due to its computational efficiency, non-conservativeness, and unified treatment of a variety of nonlinearities and uncertainties [6]. It has also been employed to analyze stability of small-sized neural networks in reinforcement learning [17], [18]; however, in these analyses, 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 quickly becomes intractable when the neural network size increases. On the contrary, the present analysis is based on *a broad characterization of smoothness* of the control function, and it does not need to access the coefficients of the neural network. We are able to reduce conservativeness of results by introducing more informative quadratic constraints, which has not been proposed before in the IQC literature to the best of the knowledge of the authors. *This significantly extends the possibilities of stability-certified reinforcement learning to large and deep neural networks in nonlinear large-scale real-world systems.*

III. Smooth Reinforcement Learning

The goal of reinforcement learning is to maximize the expected return over horizon T:

$$\eta(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \mathbb{E}\left[\sum\nolimits_{t=0}^{T} \rho^t r(\mathbf{x}_t, \mathbf{u}_t)\right], \tag{3}$$

where $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x})$ is the policy (e.g., neural network parameterized by $\boldsymbol{\theta}$), $\rho \in (0, 1]$ is the factor to discount future rewards, $r(\mathbf{x}, \mathbf{u})$ is the reward at state $\mathbf{x}$ and action $\mathbf{u}$, and $\mathbb{E}[\cdot]$ is the expectation operator. For continuous control, the actions follow a multivariate normal distribution, where $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x})$ is the mean, and the standard deviation in each action dimension is set to be a diminishing number during exploration/learning and 0 during actual deployment. With a slight abuse of notations, we will also use $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})$ to denote this normal distribution over actions. Thus, the expectation is taken over the policy, the initial state distribution and the system dynamics (1).

Trust region policy optimization is an end-to-end policy gradient learning that constrains the step length to be within a "trust region" for guaranteed improvement. By manipulating the expected return $\eta(\boldsymbol{\pi})$, the "surrogate loss" can be estimated with trajectories sampled from $\boldsymbol{\pi}_{\text{old}}$:

$$\widehat{L}_{\boldsymbol{\pi}_{\text{old}}}(\boldsymbol{\pi}) = \sum_t \frac{\boldsymbol{\pi}(\mathbf{u}_t|\mathbf{x}_t)}{\boldsymbol{\pi}_{\text{old}}(\mathbf{u}_t|\mathbf{x}_t)} \widehat{\Lambda}^{\boldsymbol{\pi}_{\text{old}}}(\mathbf{x}, \mathbf{u}), \tag{4}$$

where the ratio is also known as the importance weight, and $\widehat{\Lambda}^{\boldsymbol{\pi}_{\text{old}}}(\mathbf{x}, \mathbf{u})$ is the advantage function that measures the improvement of taking action $\mathbf{u}$ at state $\mathbf{x}$ over the old policy in terms of the value functions $V^{\boldsymbol{\pi}_{\text{old}}}$ [19].

Natural gradient is defined by a metric in the probability manifold induced by the Kullback–Leibler (KL) divergence, and it makes a step invariant to reparametrization of parameter coordinates [20]:

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \lambda \mathbf{M}_{\boldsymbol{\theta}}^{-1} \mathbf{g}_t, \tag{5}$$

where $\mathbf{g}_t$ is the standard gradient, λ is the step size, and $\mathbf{M}_{\boldsymbol{\theta}}$ defined as

$$\frac{1}{T}\sum_t \left(\frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\log \mathbf{u}_t|\mathbf{x}_t)\right)\left(\frac{\partial}{\partial \boldsymbol{\theta}} \log \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{u}_t|\mathbf{x}_t)\right)^{\top}$$

is the Fisher information matrix estimated with the trajectory data. Since the Fisher information matrix coincides with the second-order approximation of the KL divergence, one can perform back-tracking line search on the step size λ to ensure that the updated policy stays within the trust region.

Smoothness penalty (SP) is employed in this study to control the Lipschitz constants of $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot)$ during RL:

$$L_{\text{smooth}} = \sum\nolimits_{t=1}^{T} \left\| \frac{\partial}{\partial \mathbf{x}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}_t) \right\|_2^2, \tag{6}$$

which is added to $\widehat{L}_{\boldsymbol{\pi}_{\text{old}}}(\boldsymbol{\pi})$ (with a weight that yields this term roughly 1/100 of the surrogate loss) to regularize the gradient of the policy with respect to its inputs along the trajectories. This term was first proposed in "double backpropagation" [21], and recently rediscovered in [22], [23]. In addition,

we incorporate a hard threshold (HT) approach that rescales the weight matrices at each layer W_l by $(L^\circ/L(\boldsymbol{\pi_\theta}))^{1/n_L}$ if $L(\boldsymbol{\pi_\theta}) > L^\circ$, where n_L is the number of layers of the neural network. This ensures that the Lipschitz constant of the policy is bounded by a constant L°.

IV. Analysis of Stability-Certified Smoothness Margin

In this section, we introduce a new quadratic constraint on Lipschitz functions and describe the computation of smoothness margins for both linear and nonlinear systems.

A. Quadratic constraint on smooth functions

We start by recalling the definition of a Lipschitz continuous function:

Definition 4.1 (Lipschitz continuous function): Consider a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$:

(a) The function $\mathbf{f}$ is *locally Lipschitz continuous* on a set $\mathcal{B}$ if there exists a constant $L > 0$ (a.k.a., Lipschitz constant) such that

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \forall\, \mathbf{x}, \mathbf{y} \in \mathcal{B}. \quad (7)$$

(b) If $\mathbf{f}$ is Lipschitz continuous on $\mathbb{R}^n$ with constant L (i.e., $\mathcal{B} = \mathbb{R}^n$ in (7)), then $\mathbf{f}$ is called *globally Lipschitz continuous* with Lipschitz constant L.

For the purpose of stability analysis, we can express (7) as a point-wise quadratic constraint (where we use $\star$ to denote the symmetric component):

$$\begin{bmatrix} \mathbf{x} - \mathbf{y} \\ \mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y}) \end{bmatrix}^\top \begin{bmatrix} L^2\mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_m \end{bmatrix} [\star] \geq 0, \forall\, \mathbf{x}, \mathbf{y} \in \mathcal{B}. \quad (8)$$

The above constraint, nevertheless, can be conservative, because it does not explore the *inherent structure* of the problem. To illustrate this fact, consider the function

$$\mathbf{f}(x_1, x_2) = \big[\tanh(0.5x_1) - ax_1, \sin(x_2)\big]^\top, \quad (9)$$

where $x_1, x_2 \in \mathbb{R}$ and $|a| \leq 0.1$ is a deterministic but *unknown* parameter with bounded magnitude. Clearly, to satisfy (7) on $\mathbb{R}^2$ for all possible a, x_1, x_2, we need to specify the Lipschitz constant to be 1. However, this characterization is too general, because it ignores the *non-homogeneity* of f_1 and f_2, as well as the *sparsity* of the inputs x_1 and x_2. Indeed, f_1 only depends on x_1 with its slope restricted to $[-0.1, 0.6]$ for all possible values $|a| \leq 0.1$, 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 the input is common in many problems such as decentralized control. To explicitly address these requirements, we state the following quadratic constraint.

Lemma 4.2: For a vector-valued function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ that is differentiable with bounded partial derivatives on $\mathcal{B}$ (i.e., $\underline{b}_{ij} \leq \partial_j f_i(\mathbf{x}) \leq \overline{b}_{ij}$, for all $\mathbf{x} \in \mathcal{B}$),[2] the following quadratic constraint is satisfied for all $\lambda_{ij} \geq 0$, $i \in [m]$, $j \in [n]$,[3] and $\mathbf{x}, \mathbf{y} \in \mathcal{B}$,

$$\begin{bmatrix} \mathbf{x} - \mathbf{y} \\ \mathbf{q} \end{bmatrix}^\top \mathbf{M}_\pi(\boldsymbol{\lambda}) [\star] \geq 0, \quad (10)$$

where $\mathbf{M}_\pi(\boldsymbol{\lambda})$ is given by

$$\begin{bmatrix} \operatorname{diag}\Big(\Big\{\sum_i \lambda_{ij}(\bar{c}_{ij}^2 - c_{ij}^2)\Big\}\Big) & \boldsymbol{\Lambda}(\{\lambda_{ij}, c_{ij}\})^\top \\ \boldsymbol{\Lambda}(\{\lambda_{ij}, c_{ij}\}) & \operatorname{diag}\big(\{-\lambda_{ij}\}\big) \end{bmatrix},$$

and $\mathbf{q} = \big[q_{11}, ..., q_{1n}, ..., q_{m1}, ..., q_{mn}\big]^\top \in \mathbb{R}^{mn}$ is a function of $\mathbf{x}$ and $\mathbf{y}$, $\{-\lambda_{ij}\}$ follows the same index order as $\mathbf{q}$, $\boldsymbol{\Lambda}(\{\lambda_{ij}, c_{ij}\})^\top = \Big[\operatorname{diag}\big(\{\lambda_{1j}c_{1j}\}\big)\ ...\ \operatorname{diag}\big(\{\lambda_{mj}c_{mj}\}\big)\Big] \in \mathbb{R}^{n\times mn}$, $c_{ij} = \frac{1}{2}\Big(\underline{b}_{ij} + \overline{b}_{ij}\Big)$, $\bar{c}_{ij} = \overline{b}_{ij} - c_{ij}$, and $\mathbf{q}$ is related to the output of $\mathbf{f}$ by the constraint:

$$\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y}) = \big[\mathbf{I}_m \otimes \mathbf{1}_{1\times n}\big]\mathbf{q} = \mathbf{W}\mathbf{q}, \quad (11)$$

where $\otimes$ denotes the Kronecker product.

Proof: See Appendix A. ∎

This bound is a direct consequence of standard tools in real analysis, partially inspired by [25]. To understand this result, note that (10) is equivalent to:

$$\sum_{ij} \lambda_{ij}\Big((\bar{c}_{ij}^2 - c_{ij}^2)(x_j - y_j)^2 + 2c_{ij}q_{ij}(x_j - y_j) - q_{ij}^2\Big) \geq 0 \quad (12)$$

for all nonnegative numbers $\lambda_{ij} \geq 0$, with $f_i(\mathbf{x}) - f_i(\mathbf{y}) = \sum_{j=1}^n q_{ij}$. Since (12) 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 $i \in [m]$, $j \in [n]$, which is a direct result of the bounds imposed on the partial derivatives of f_i. To illustrate its usage, let us apply it to example (9), where $\underline{b}_{11} = -0.1, \overline{b}_{11} = 0.6, \underline{b}_{22} = -1, \overline{b}_{22} = 1$, and all the other bounds $(\underline{b}_{12}, \overline{b}_{12}, \underline{b}_{21}, \overline{b}_{21})$ are zero. This yields a more informative constraint than simply relying on the Lipschitz constraint (8). In fact, for Lipschitz functions, we have $\overline{b}_{ij} = -\underline{b}_{ij} = L$, and by limiting the choice of $\lambda_{ij} = \begin{cases} \lambda & \text{if } i = 1 \\ 0 & \text{if } i \neq 1 \end{cases}$, (12) is reduced to (8). Nonetheless, Lemma 4.2 can incorporate richer information about *input sparsity* and *output structures*, thus it can yield non-trivial stability bounds in practice.

The constraint introduced above is *not* a standard IQC, since it involves an intermediate variable $\mathbf{q}$ that relates to the output $\mathbf{f}$ through a set of linear equalities. In relation to existing IQCs, it has wider applications to characterize smooth functions. The Zames-Falb IQC introduced in [26] has been widely used for single-input single-output function $f : \mathbb{R} \to \mathbb{R}$, but it requires the function to be monotone with slope restricted to $[\alpha, \beta]$ and $\alpha \geq 0$, i.e., $0 \leq \alpha \leq \frac{f(x)-f(y)}{x-y} \leq \beta$ for $x \neq y$. The multi-input multi-output extension holds true only if the nonlinearity $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is restricted to be the *gradient of a convex real-valued function* [27]. The sector IQC is in fact (8). By contrast, the quadratic constraint in Lemma 4.2 can be applied to non-monotone, vector-valued Lipschitz functions.

[2]The analysis can be extended to non-differentiable but Lipschitz continuous functions (e.g., ReLU $\max\{0, x\}$) using the notion of generalized gradient [24, Chap. 2].

[3]We use the set notation $[n] = \{1, ..., n\}$.

B. Computation of smoothness margin

We illustrates the computation of smoothness margin for an LTI system $\mathcal{G}$ with the state-space representation:

$$\begin{cases} \dot{\mathbf{x}}_G = \mathbf{A}\mathbf{x}_G + \mathbf{B}\mathbf{u} \\ \mathbf{y} \quad = \mathbf{x}_G \end{cases}, \tag{13}$$

where $\mathbf{x}_G \in \mathbb{R}^n$ is the state and $\mathbf{y} \in \mathbb{R}^n$ is the output. We can connect this linear system in *feedback* with a Lipschitz-continuous controller $\boldsymbol{\pi} : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$\begin{cases} \mathbf{u} \ = \mathbf{e} + \mathbf{w} \\ \mathbf{w} = \boldsymbol{\pi}(\mathbf{C}_\pi \mathbf{y}) \end{cases}, \tag{14}$$

where $\mathbf{e} \in \mathbb{R}^m$ is the exploration vector introduced in reinforcement learning, $\mathbf{w} \in \mathbb{R}^m$ is the policy action, and $\mathbf{C}_\pi \in \mathbb{R}^{n\times n}$ is an observation matrix that determines the set of states observable for the reinforcement agent (this matrix is absorbed into the partial gradient specifications in Lemma 4.2). Assume that the policy $\boldsymbol{\pi}$ satisfies the conditions in Lemma 4.2, then we can express $\mathbf{w} = \mathbf{W}\mathbf{q}$ using the internal signal $\mathbf{q} \in \mathbb{R}^{mn}$, which satisfies the quadratic constraint (10).

We are interested in certifying the largest Lipschitz constant L° (i.e., smoothness margin) of $\boldsymbol{\pi}(\cdot)$ such that the interconnected system is input-output stable at all time $T \geq 0$, i.e.,

$$\int_0^T \left\|\mathbf{y}(t)\right\|_2^2 dt \leq \gamma^2 \int_0^T \left\|\mathbf{e}(t)\right\|_2^2 dt, \tag{15}$$

where γ^2 is a finite upper bound for the $\mathcal{L}_2$ gain. To this end, define the SDP$(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L)$ as follows:

$$\text{SDP}(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L) : \begin{bmatrix} \mathcal{O}(\mathbf{P}, \boldsymbol{\lambda}, L) & \mathcal{S}(\mathbf{P}) \\ \mathcal{S}(\mathbf{P})^\top & -\gamma \mathbf{I}_m \end{bmatrix} \preceq 0, \tag{16}$$

where $\preceq$ indicates negative semi-definite, $\mathbf{P} = \mathbf{P}^\top \succeq 0$, L is the Lipschitz upper bound of $\boldsymbol{\pi}$, $\mathcal{O}(\mathbf{P}, \boldsymbol{\lambda}, L)$ is given by

$$\begin{bmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} + \frac{1}{\gamma}\mathbf{I}_n & \mathbf{P}\mathbf{B}\mathbf{W} \\ \mathbf{W}^\top \mathbf{B}^\top \mathbf{P} & \mathbf{0}_{mn,mn} \end{bmatrix} + \mathbf{M}_\pi(\boldsymbol{\lambda}, L),$$

and

$$\mathcal{S}(\mathbf{P}) = \begin{bmatrix} \mathbf{P}\mathbf{B} \\ \mathbf{0}_{mn,m} \end{bmatrix},$$

where $\mathbf{M}_\pi(\boldsymbol{\lambda}, L)$ is defined in (10) with $|\underline{b}_{ij}| \leq L, |\bar{b}_{ij}| \leq L$ (and 0 if the j-th observation is not used for the i-th action) and multipliers $\boldsymbol{\lambda} = \{\lambda_{ij}\}$ for $i \in [m], j \in [n]$. The next theorem can be used to certify stability of the interconnected system.

Theorem 4.3: Let $\boldsymbol{\pi} \in \mathbb{R}^n \to \mathbb{R}^m$ be a bounded causal controller. Assume that:

(i) the interconnection of $\mathcal{G}$ and $\boldsymbol{\pi}$ is well-posed;
(ii) $\boldsymbol{\pi}$ is L-Lipschitz with bounded partial derivatives on $\mathcal{B}$ (i.e., $\underline{b}_{ij} \leq \partial_j \pi_i(\mathbf{x}) \leq \bar{b}_{ij}$, and $|\underline{b}_{ij}|, |\bar{b}_{ij}| \leq L$ for all $\mathbf{x} \in \mathcal{B}$, $i \in [m]$ and $j \in [n]$);
(iii) there exist $\mathbf{P} = \mathbf{P}^\top \succeq 0$ and a scaler $\gamma > 0$ such that SDP$(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L)$ is feasible.

Then the interconnection of $\mathcal{G}$ and $\boldsymbol{\pi}$ is stable.

Proof: See Appendix B. ∎

The above result offers a computational approach to certify the maximal Lipschitz constant of a generic nonlinear controller. Given an LTI system (13), the first step is to represent the reinforcement policy as a "black box" in a feedback interconnection. Because the controller parameters can not be known *a priori* and will be continuously updated during learning, we use the smoothness property and some high-level domain knowledge in the form of refined partial gradient bounds. A simple but conservative choice is a $\mathcal{L}_2$-gain bound IQC; nevertheless, to achieve a less conservative result, we can employ Lemma 4.2 to exploit both the sparsity of the inputs and the non-homogeneity of the outputs. For a given Lipschitz constant L, we find the smallest γ such that SDP$(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L)$ is feasible, which also corresponds to the upper bound on the $\mathcal{L}_2$ gain of the interconnected system both during learning (with exploration excitation $\mathbf{e}$) and actual deployment. If γ is finite, then the system is provably stable.

We remark that SDP$(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L)$ is quasiconvex, in the sense that it reduces to a standard LMI with fixed γ and L [28]. To solve it numerically, we start with a small Lipschitz constant L and gradually increase γ until a solution $(\mathbf{P}, \boldsymbol{\lambda})$ is found. Then, we increase L and repeat the process. Each iteration (i.e., LMI for a given set of γ and L) can be solved efficiently by interior-point methods.

C. Extension to nonlinear systems with uncertainty

The analysis for LTI systems can be extended to a generic nonlinear system described in (1). The key idea is to model the nonlinear and potentially time-varying part $\mathbf{g}_t(\mathbf{x}(t))$ as an uncertain block with IQC constraints on its behavior. Specifically, consider the LTI component $\underline{\mathcal{G}}$:

$$\begin{cases} \dot{\mathbf{x}}_G = \mathbf{A}\mathbf{x}_G + \mathbf{B}\mathbf{u} + \mathbf{v} \\ \mathbf{y} \quad = \mathbf{x}_G \end{cases}, \tag{17}$$

where $\mathbf{x}_G \in \mathbb{R}^n$ is the state and $\mathbf{y} \in \mathbb{R}^n$ is the output. The nonlinear part is connected in feedback:

$$\begin{cases} \mathbf{u} \ = \mathbf{e} + \mathbf{w} \\ \mathbf{w} = \boldsymbol{\pi}(\mathbf{C}_\pi \mathbf{y}) \\ \mathbf{v} \ = \mathbf{g}_t(\mathbf{y}) \end{cases}, \tag{18}$$

where $\mathbf{e} \in \mathbb{R}^m$, $\mathbf{w} \in \mathbb{R}^m$ and $\mathbf{C}_\pi \in \mathbb{R}^{n\times n}$ are defined as before, and $\mathbf{g}_t : \mathbb{R}^n \to \mathbb{R}^n$ is the nonlinear and time-varying component. In addition to characterizing $\boldsymbol{\pi}(\cdot)$ using the Lipschitz property (10), we assume that $\mathbf{g}_t : \mathbb{R}^n \to \mathbb{R}^n$ satisfies the IQC defined by $(\boldsymbol{\Psi}, \mathbf{M}_g)$ (see [29] for more details). The system $\boldsymbol{\Psi} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ has the state-space representation:

$$\begin{cases} \dot{\boldsymbol{\psi}} = \mathbf{A}_\psi \boldsymbol{\psi} + \mathbf{B}_\psi^v \mathbf{v} + \mathbf{B}_\psi^y \mathbf{y} \\ \mathbf{z} \ = \mathbf{C}_\psi \boldsymbol{\psi} + \mathbf{D}_\psi^v \mathbf{v} + \mathbf{D}_\psi^y \mathbf{y} \end{cases}, \tag{19}$$

where $\boldsymbol{\psi} \in \mathbb{R}^n$ is the internal state and $\mathbf{z} \in \mathbb{R}^n$ is the *filtered* output. By denoting $\mathbf{x} = \begin{bmatrix} \mathbf{x}_G^\top & \boldsymbol{\psi}^\top \end{bmatrix}^\top \in \mathbb{R}^{2n}$ as the new state, we can combine (17) and (19) by reducing $\mathbf{y}$ and letting

$\mathbf{w} = \mathbf{W}\mathbf{q}$:

$$\begin{cases} \dot{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0}_{n,n} \\ \mathbf{B}_\psi^y & \mathbf{A}_\psi \end{bmatrix}}_{\underline{\mathbf{A}}} \mathbf{x} + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0}_{n,m} \end{bmatrix}}_{\underline{\mathbf{B}}_e} \mathbf{e} + \underbrace{\begin{bmatrix} \mathbf{B}\mathbf{W} \\ \mathbf{0}_{n,mn} \end{bmatrix}}_{\underline{\mathbf{B}}_q} \mathbf{q} + \underbrace{\begin{bmatrix} \mathbf{I}_n \\ \mathbf{B}_\psi^v \end{bmatrix}}_{\underline{\mathbf{B}}_v} \mathbf{v} \\ \mathbf{z} = \underbrace{\begin{bmatrix} \mathbf{D}_\psi^y & \mathbf{C}_\psi \end{bmatrix}}_{\underline{\mathbf{C}}} \mathbf{x} + \mathbf{D}_\psi^v \mathbf{v} \end{cases}, \tag{20}$$

where $\underline{\mathbf{A}}$, $\underline{\mathbf{B}}_e$, $\underline{\mathbf{B}}_q$, $\underline{\mathbf{B}}_v$ and $\underline{\mathbf{C}}$ are matrices of proper dimensions. Similar to the case of LTI systems, the objective is to find an upper bound L° on the Lipschitz constant of $\boldsymbol{\pi}(\cdot)$ such that the system is stable. In the same vein, we define $\underline{\text{SDP}}(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L)$:

$$\begin{bmatrix} \mathcal{O}(\mathbf{P}, \boldsymbol{\lambda}, L) & \mathcal{O}_v(\mathbf{P}) & \mathcal{S}(\mathbf{P}) \\ \mathcal{O}_v(\mathbf{P})^\top & \mathbf{D}_\psi^{v\top} \mathbf{M}_q \mathbf{D}_\psi^v & \mathbf{0}_{n,n} \\ \mathcal{S}(\mathbf{P})^\top & \mathbf{0}_{n,n} & -\gamma \mathbf{I}_m \end{bmatrix} \preceq 0, \quad \text{(SDP-NL)}$$

where $\mathbf{P} = \mathbf{P}^\top \succeq 0$, L is the Lipschitz upper bound of $\boldsymbol{\pi}$, and

$$\mathcal{O}(\mathbf{P}, \boldsymbol{\lambda}, L) = \begin{bmatrix} \underline{\mathbf{A}}^\top \mathbf{P} + \mathbf{P}\underline{\mathbf{A}} + \underline{\mathbf{C}}^\top \mathbf{M}_g \underline{\mathbf{C}} & \mathbf{P}\underline{\mathbf{B}}_q \\ \underline{\mathbf{B}}_q^\top \mathbf{P} & \mathbf{0}_{mn,mn} \end{bmatrix} + \mathbf{M}_\pi(\boldsymbol{\lambda}, L) + \frac{1}{\gamma} \begin{bmatrix} \mathbf{I}_n & \\ & \mathbf{0}_{(m+1)n \times (m+1)n} \end{bmatrix},$$

$$\mathcal{O}_v(\mathbf{P}) = \begin{bmatrix} \underline{\mathbf{C}}^\top \mathbf{M}_q \mathbf{D}_\psi^v + \mathbf{P}\underline{\mathbf{B}}_v \\ \mathbf{0}_{mn,n} \end{bmatrix}, \mathcal{S}(\mathbf{P}) = \begin{bmatrix} \mathbf{P}\underline{\mathbf{B}}_e \\ \mathbf{0}_{mn,m} \end{bmatrix},$$

where $\mathbf{M}_\pi(\boldsymbol{\lambda}, L)$ is defined in (10). The next theorem provides stability certificate for the nonlinear time-varying system (1).

Theorem 4.4: Let $\boldsymbol{\pi} \in \mathbb{R}^n \to \mathbb{R}^m$ be a bounded causal controller. Assume that:

(i) the interconnection of $\mathcal{G}$, $\boldsymbol{\pi}$, and $\mathbf{g}_t$ is well-posed;
(ii) $\boldsymbol{\pi}$ is L-Lipschitz with bounded partial derivatives on $\mathcal{B}$ (i.e., $\underline{b}_{ij} \leq \partial_j \pi_i(\mathbf{x}) \leq \bar{b}_{ij}$, and $|\underline{b}_{ij}|, |\bar{b}_{ij}| \leq L$ for all $\mathbf{x} \in \mathcal{B}$, $i \in [m]$ and $j \in [n]$);
(iii) $\mathbf{g}_t \in \text{IQC}(\boldsymbol{\Psi}, \mathbf{M}_g)$, where $\boldsymbol{\Psi}$ is stable;
(iv) there exist $\mathbf{P} = \mathbf{P}^\top \succeq 0$ and a scaler $\gamma > 0$ such that $\underline{\text{SDP}}(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L)$ is feasible.

Then, the feedback interconnection of the nonlinear system (1) and $\boldsymbol{\pi}$ is stable (i.e., it satisfies (15)).

Proof: See Appendix C. ∎

V. Case Study

In this section, we empirically study the smoothness margin for reinforcement learning agents in a real-world problem, namely power grid frequency regulation [30], [31]. The IEEE 39-Bus New England Power System under analysis is shown in Fig. 2. Under the star-connected information structure, each generator can only share its rotor angle and frequency information with a pre-specified set of geographically separated counterparts. Decentralized control has been long known to be an NP-hard problem in general [3]. End-to-end multi-agent reinforcement learning comes in handy, because it does not require model information [32]. The main task is to adjust the mechanical power inputs to each generator such that the phases and frequencies at each bus stabilizes after possible perturbation. If θ_i denotes the voltage angle at a generator bus i (in rad), the physics of power systems can be modeled by the per-unit swing equation:

$$Q_i \ddot{\theta}_i + K_i \dot{\theta} = P_{M_i} - P_{E_i},$$

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.), Q_i is the inertia coefficient of the generator at bus i (in p.u.-sec^2/rad), and K_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:

$$P_{E_i} = \sum_{j=1}^{n} |V_i||V_j| \left(G_{ij} \cos(\theta_i - \theta_j) + S_{ij} \sin(\theta_i - \theta_j) \right),$$

where n is the number of buses in the system, G_{ij} and S_{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 magnitudes smaller than the susceptance S_{ij}, for the simplicity of mathematical treatment, we omit the $\cos(\cdot)$ term and only keep the $\sin(\cdot)$ term. This leads to a less conservative approximation compared to the well-known DC model.

Let the rotor angle states and the frequency states be denoted as $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \cdots & \theta_n \end{bmatrix}^\top$ and $\boldsymbol{\omega} = \begin{bmatrix} \omega_1 & \cdots & \omega_n \end{bmatrix}^\top$, and the generator mechanical power injections be denoted as $\mathbf{P}_M = \begin{bmatrix} P_{M_1} & \cdots & P_{M_n} \end{bmatrix}^\top$. Then, the state-space representation of the nonlinear system is given by:

$$\begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{Q}^{-1}\mathbf{L} & -\mathbf{Q}^{-1}\mathbf{K} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\omega} \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{Q}^{-1} \end{bmatrix}}_{\mathbf{B}} \mathbf{P}_M + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{g}(\boldsymbol{\theta}) \end{bmatrix}}_{\mathbf{g}(\mathbf{x})}$$

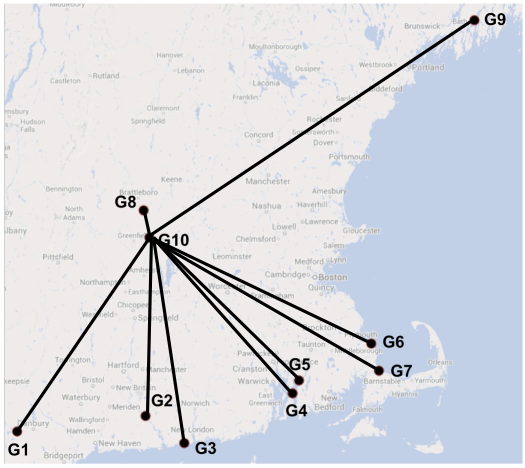


Fig. 2: New England Power System with a star-connected information structure.

where $\mathbf{g}(\boldsymbol{\theta}) = \begin{bmatrix} g_1(\boldsymbol{\theta}) & \cdots & g_n(\boldsymbol{\theta}) \end{bmatrix}^\top$ with $g_i(\boldsymbol{\theta}) = \sum_{j=1}^n \frac{S_{ij}}{Q_i}\big((\theta_i - \theta_j) - \sin(\theta_i - \theta_j)\big)$, and $\mathbf{Q} = \operatorname{diag}\big(\{Q_i\}_{i=1}^n\big)$, $\mathbf{K} = \operatorname{diag}\big(\{K_i\}_{i=1}^n\big)$, and $\mathbf{L}$ is a Laplacian matrix whose entries are specified in [30, Sec. IV-B]. For linearization (also known as DC approximation), the nonlinear part $\mathbf{g}(\mathbf{x})$ is assumed to be zero when the phase differences are small [30], [31]. On the contrary, we deal with this term in the smoothness margin analysis to demonstrate its capability of producing non-conservative certificates even for nonlinear systems. We assume that there exists a distributed nominal controller that stablizes the system, which may be designed by H_∞-controller synthesis [4] and is out of the scope of this paper.

Smoothness margin analysis: The nonlinearities in $\mathbf{g}(\mathbf{x})$ 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 a slope restricted to $[0, 1-\cos(\overline{\theta})]$ for $\Delta\theta_{ij} \in [-\overline{\theta}, \overline{\theta}]$ and thus can be treated using the Zames-Falb IQC. In the smoothness margin analysis, we assume $\overline{\theta} = \frac{\pi}{3}$, which requires the phase angle difference to be within $[-\frac{\pi}{3}, \frac{\pi}{3}]$. To study the stability of the multi-agent policies, we adopt a *black-box* approach by simply considering the input-output constraint. By applying the $\mathcal{L}_2$ constraint in (8), we can only certify stability for Lipschitz constants up to 0.4. Because the distributed control has natural structures of input sparsity, we can characterize it by setting the lower and upper bounds $\underline{b}_{ij} = \overline{b}_{ij} = 0$ for agent i that does not utilize observation j, and $\overline{b}_{ij} = -\underline{b}_{ij} = L$ otherwise, where L is the Lipschitz constant to be certified. This information can be encoded in $\underline{\text{SDP}}(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L)$ in (SDP-NL), which can be solved for L up to 0.8 (doubling the certificate provided by $\mathcal{L}_2$ constraint).

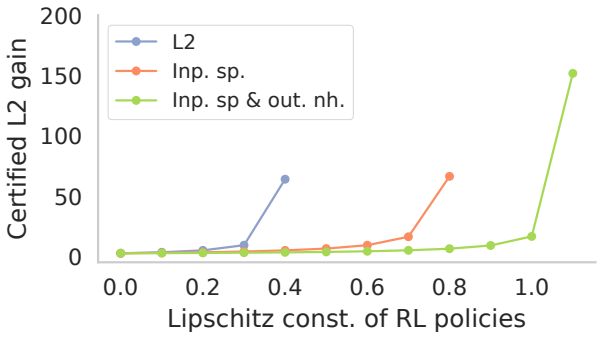


Fig. 3: Certified $\mathcal{L}_2$ gain (γ in (2)) for smoothness margins in nonlinear decentralized power frequency stabilization, given by the constraint (8) and Lemma 4.2 with input sparsity and output nonhomogeneity.

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. 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 revealed at the early stage, typically after several iterations, when the Lipschitz constants of the agents are far less than 0.8 (the certificate provided by Theorem 4.4). When we incorporate this characterization into the partial gradient bounds (e.g., $\overline{b}_{ij} = -0.1L$ and $\underline{b}_{ij} = L$ for each agent i that exhibits a positive gradient with respect to observation j), we can extend the certificate up to 1.1, as shown in Fig. 3.

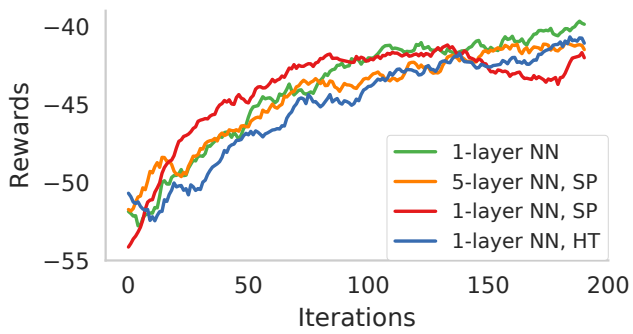


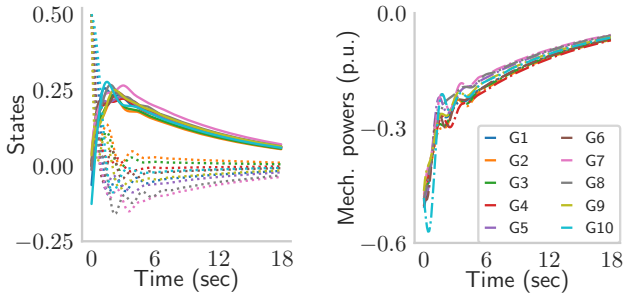
Fig. 4: Trajectories of rewards during reinforcement learning for various neural networks (each hidden layer consists of 3 neurons). The plot shows a running average of rewards for every 10 iterations.

Policy gradient reinforcement learning: To conduct multi-agent reinforcement learning, each controller P_{M_i} is considered to be a neural network that takes inputs of observed phases and frequencies to determine the mechanical power injection at bus i. In this experiment, the *unknown* reward is a quadratic function that weighs the square of each state variable $\mathbf{x}$ by 10 and the square of each control input by 0.1. Since we aim at designing a generic controller that allows the initial state to vary in a large operating region (between -0.5 and 0.5), and we do not assume the knowledge of the true reward, the methods proposed in [30], [31] for linear distributed controller design cannot be employed. We employ TRPO [19] with natural gradient [20] as the baseline, in addition to smooth RL methods with SP and HT in Sec. III.

The reward trajectories are shown in Fig. 4. The SP method has higher initial learning rates, and all methods significantly improve the performance after 150 iterations (each iteration includes 100 independent policy evaluations, which amounts to 25 minutes of data if deployed in real power systems). The learned policy demonstrates faster stabilization of power grid frequencies compared to the nominal (cost 23.9 for neural network versus 50.8 for nominal controller). More importantly, we are able to certify stability of the policies throughout the exploration and deployment phases by monitoring the Lipschitz constants (Fig. 6 demonstrates the case of HT). This comprises a key step towards safe deployment of reinforcement learning in real-world environments.

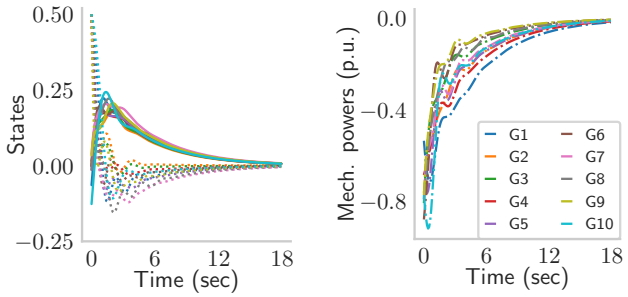
VI. Conclusion

We proposed a method to certify stability of reinforcement learning in real-world dynamical systems. The analysis is based on a general characterization of smoothness measured by Lipschitz constants, and is applicable to a large class of nonlinear controllers such as (deep) neural networks. A numerical evaluation on decentralized power grid frequency regulation demonstrated that the learned policies significantly



(a) Nom.: states θ_i, ω_i.

(b) Nom.: actions.



(c) RL: states θ_i, ω_i.

(d) RL: actions.

Fig. 5: Typical examples of system behaviors under the nominal controller (cost: 50.8) and neural network given by reinforcement learning (cost: 23.9).

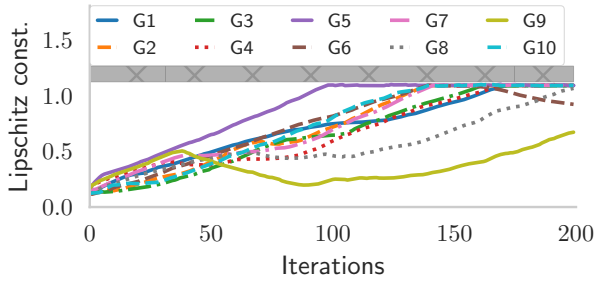


Fig. 6: Monitoring of Lipschitz constants of agent policies during learning. With hard thresholding, they remain bounded below the certified margin (grey band).

surpass nominal controllers in performance while maintaining strong stability certificates. The results are parallel to the study of security and robustness of neural networks to adversarial data injections.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[2] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017.

[3] L. Bakule, "Decentralized control: An overview," *Annual reviews in control*, vol. 32, no. 1, pp. 87–98, 2008.

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

[5] 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.

[6] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE Transactions on Automatic Control*, vol. 42, no. 6, pp. 819–830, 1997.

[7] J. C. Willems, "Dissipative dynamical systems part ii: Linear systems with quadratic supply rates," *Archive for rational mechanics and analysis*, vol. 45, no. 5, pp. 352–393, 1972.

[8] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[9] S. P. Coraluppi and S. I. Marcus, "Risk-sensitive and minimax control of discrete-time, finite-state Markov decision processes," *Automatica*, vol. 35, no. 2, pp. 301–309, 1999.

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

[11] 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.

[12] W. Wiesemann, D. Kuhn, and B. Rustem, "Robust Markov decision processes," *Mathematics of Operations Research*, vol. 38, no. 1, pp. 153–183, 2013.

[13] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[14] 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.

[15] T. J. Perkins and A. G. Barto, "Lyapunov design for safe reinforcement learning," *Journal of Machine Learning Research*, vol. 3, no. Dec, pp. 803–832, 2002.

[16] 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.

[17] 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.

[18] 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*, vol. 18, no. 4, pp. 993–1002, 2007.

[19] 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.

[20] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural computation*, vol. 10, no. 2, pp. 251–276, 1998.

[21] H. Drucker and Y. Le Cun, "Improving generalization performance using double backpropagation," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 991–997, 1992.

[22] A. G. Ororbia II, D. Kifer, and C. L. Giles, "Unifying adversarial training algorithms with data gradient regularization," *Neural computation*, vol. 29, no. 4, pp. 867–887, 2017.

[23] 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. 5769–5779.

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

[25] A. Zemouche and M. Boutayeb, "On LMI conditions to design observers for lipschitz nonlinear systems," *Automatica*, vol. 49, no. 2, pp. 585–591, 2013.

[26] G. Zames and P. Falb, "Stability conditions for systems with monotone and slope-restricted nonlinearities," *SIAM Journal on Control*, vol. 6, no. 1, pp. 89–108, 1968.

[27] 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.

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

[29] P. Seiler, "Stability analysis with dissipation inequalities and integral quadratic constraints," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1704–1709, 2015.

[30] G. Fazelnia, R. Madani, A. Kalbat, and J. Lavaei, "Convex relaxation for optimal distributed control problems," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 206–221, 2017.

[31] S. Fattahi, G. Fazelnia, and J. Lavaei, "Transformation of optimal centralized controllers into near-globally optimal static distributed controllers," *IEEE Transactions on Automatic Control*, 2017.

[32] D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers, "Evolutionary dynamics of multi-agent learning: A survey," *Journal of Artificial Intelligence Research*, vol. 53, pp. 659–697, 2015.

APPENDIX

A. Proof of Lemma 4.2

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

$$\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y}) = \begin{bmatrix} \sum_{j=1}^n \delta_{1j}(\mathbf{x}, \mathbf{y})(x_j - y_j) \\ \vdots \\ \sum_{j=1}^n \delta_{mj}(\mathbf{x}, \mathbf{y})(x_j - y_j) \end{bmatrix}. \tag{21}$$

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

$$\begin{bmatrix} x_j - y_j \\ q_{ij} \end{bmatrix}^\top \begin{bmatrix} \bar{c}_{ij}^2 - c_{ij}^2 & c_{ij} \\ c_{ij} & -1 \end{bmatrix} \begin{bmatrix} \star \end{bmatrix} \geq 0. \tag{22}$$

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

B. Proof of Theorem 4.3

By multiplying $\begin{bmatrix} \mathbf{x}_G^\top & \mathbf{q}^\top & \mathbf{e}^\top \end{bmatrix}^\top$ to the left and its transpose to the right of the augmented matrix in (16), and using the constraints $\mathbf{w} = \mathbf{W}\mathbf{q}$ and $\mathbf{y} = \mathbf{x}_G$, SDP$(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L)$ can be written as a dissipation inequality:

$$\dot{V}(\mathbf{x}_G) + \begin{bmatrix} \mathbf{x}_G \\ \mathbf{q} \end{bmatrix}^\top \mathbf{M}_\pi \begin{bmatrix} \mathbf{x}_G \\ \mathbf{q} \end{bmatrix} \leq \gamma \mathbf{e}^\top \mathbf{e} - \frac{1}{\gamma} \mathbf{y}^\top \mathbf{y},$$

where $V(\mathbf{x}_G) = \mathbf{x}_G^\top \mathbf{P} \mathbf{x}_G$ 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 4.2, if SDP$(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L)$ is feasible with a solution $(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L)$, we have:

$$\dot{V}(\mathbf{x}_G) + \frac{1}{\gamma} \mathbf{y}^\top \mathbf{y} - \gamma \mathbf{e}^\top \mathbf{e} \leq 0, \tag{23}$$

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 $\mathbf{P} \succeq 0$ that:

$$\int_0^T \|\mathbf{y}(t)\|^2 dt \leq \gamma^2 \int_0^T \|\mathbf{e}(t)\|^2 dt. \tag{24}$$

Hence, the interconnected system with L-Lipschitz reinforcement policy is stable.

C. Proof of Theorem 4.4

The proof is in the same vein as that of Theorem 4.3. The main technical difference is the consideration of filtered states $\boldsymbol{\psi}$ and outputs $\mathbf{z}$ to impose IQC constraints on the nonlinearities $\mathbf{g}_t(\mathbf{y})$ in the dynamical system [6]. The dissipation inequality follows by multiplying both sides of the matrix in (SDP-NL) by $\begin{bmatrix} \mathbf{x}^\top & \mathbf{q}^\top & \mathbf{v}^\top & \mathbf{e}^\top \end{bmatrix}^\top$ and its transpose:

$$\dot{V}(\mathbf{x}) + \mathbf{z}^\top \mathbf{M}_g \mathbf{z} + \begin{bmatrix} \mathbf{x}_G \\ \mathbf{q} \end{bmatrix}^\top \mathbf{M}_\pi \begin{bmatrix} \mathbf{x}_G \\ \mathbf{q} \end{bmatrix} \leq \gamma \mathbf{e}^\top \mathbf{e} - \frac{1}{\gamma} \mathbf{y}^\top \mathbf{y},$$

where $\mathbf{x}$ and $\mathbf{z}$ are defined in (20), and $V(\mathbf{x}) = \mathbf{x}^\top \mathbf{P} \mathbf{x}$ is the storage function with $\dot{V}(\cdot)$ as its time derivative. The first term is non-negative because $\mathbf{g}_t \in \text{IQC}(\boldsymbol{\Psi}, \mathbf{M}_g)$, and the second term is non-negative due to the smoothness quadratic costraint in Lemma 4.2. Thus, integrating the inequality from $t = 0$ to $t = T$, and if there exists a feasible solution $\mathbf{P} \succeq 0$ to $\underline{\text{SDP}}(\mathbf{P}, \boldsymbol{\lambda}, \gamma, L)$, it yields that:

$$\int_0^T \|\mathbf{y}(t)\|^2 dt \leq \gamma^2 \int_0^T \|\mathbf{e}(t)\|^2 dt. \tag{25}$$

Hence, the nonlinear system interconnected with L-Lipschitz continuous reinforcement policies is certifiably stable in the sense of finite $\mathcal{L}_2$ gain.