Revisiting PGD Attacks for Stability Analysis of High-Dimensional Nonlinear Systems and Perception-Based Control

Aaron Havens, Darioush Kevian, Peter Seiler, *Fellow, IEEE*, Geir Dullerud, *Fellow, IEEE*, and Bin Hu

***Abstract*—Many existing region-of-attraction (ROA) analysis tools find difficulty in addressing feedback systems with large-scale neural network (NN) policies and/or high-dimensional sensing modalities such as cameras. In this letter, we tailor the projected gradient descent (PGD) attack method as a general-purpose ROA analysis tool for high-dimensional nonlinear systems and end-to-end perception-based control. We show that the ROA analysis can be approximated as a constrained maximization problem such that PGD-based iterative methods can be directly applied. In the model-based setting, we show that the PGD updates can be efficiently performed using back-propagation. In the model-free setting, we propose a finite-difference PGD estimate which is general and only requires a black-box simulator for generating the trajectories of the closed-loop system given any initial state. Finally, we demonstrate the scalability and generality of our analysis tool on several numerical examples with large state dimensions or complex image observations.**

***Index Terms*—Region of attraction, nonlinear systems, perception-based control.**

I. Introduction

RECENTLY, deep reinforcement learning (DRL) techniques have gained popularity in control [1], [2]. For control applications, DRL has two main advantages. First, DRL provides a general-purpose framework for addressing complex nonlinear dynamics (e.g., contact force, etc). Second, DRL can be applied to train pixel-based control systems in an end-to-end manner [3], [4]. Despite these advantages, applications of DRL in real-world control systems are still rare. One issue is that the stability and robustness properties of such DRL-based controllers have not been fully understood. There is an urgent need to develop new analysis tools for addressing the stability and robustness of DRL-based control systems.

In this letter, we are interested in estimating the region of attraction (ROA) of nonlinear control systems with high-dimensional states (e.g., up to 1000 states) and/or rich observations (e.g., images). There are three main technical difficulties. First, the state dimension can be high, and the NN policies can have a large number of hidden neurons, leading to scalability issues. Second, the feed-forward dynamics of perception-based control systems are typically not fully known due to the complex mapping from the plant state to the image. Third, in general, the control action may depend on the past output measurements, and thus the coupling of system states at different time steps can be complicated. The first two issues will cause trouble for existing Lyapunov-based ROA analysis methods using semidefinite programming [5]–[8] or mixed-integer programs [9]–[11]. Due to the last issue, the methods of Lyapunov neural networks [12]–[14] or other stability certificate learning methods [15]–[17] may also be not applicable since these methods typically require the control action to depend on the current state. Our goal is to develop a ROA analysis method which can address the above three issues simultaneously.

To achieve our goal, we will borrow the method of the projected gradient descent (PGD) attack developed in the adversarial learning literature [18]–[21] and tailor it as a general-purpose ROA analysis tool. Originally, the PGD attack was developed to find the worst-case perturbation that can shift the output of neural networks significantly and degrade the performances of classifiers in computer vision. In our paper, we build a connection between PGD attack and the ROA analysis. We show that the ROA analysis can be approximated as a constrained maximization problem whose goal is to find the worst-case initial condition which shifts the terminal state most significantly. Then PGD can be directly

Manuscript received 21 March 2022; revised 28 May 2022; accepted 15 June 2022. Date of publication 4 July 2022; date of current version 18 July 2022. The work of Aaron Havens and Bin Hu was supported in part by the NSF Award CAREER under Grant 2048168 and in part by 2020 Amazon Research Award. The work of Darioush Kevian and Geir Dullerud was supported by NSF under Grant ECCS 19-32735. The work of Peter Seiler was supported by the U.S. Office of Naval Research (ONR) under Grant N00014-18-1-2209. Recommended by Senior Editor C. Prieur. *(Aaron Havens and Darioush Kevian contributed equally to this work.) (Corresponding author: Aaron Havens.)*

Aaron Havens and Bin Hu are with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Champaign, IL 61820 USA (e-mail: ahavens2@illinois.edu; binhu7@illinois.edu).

Darioush Kevian and Geir Dullerud are with the Coordinated Science Laboratory and the Department of Mechanical Science and Engineering, University of Illinois at Urbana–Champaign, IL 61820 USA (e-mail: dk12@illinois.edu; dullerud@illinois.edu).

Peter Seiler is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: pseiler@umich.edu).

Digital Object Identifier 10.1109/LCSYS.2022.3188016

applied to solve the resultant maximization problem. Such a maximization formulation is not based on Lyapunov theory, and hence does not require any particular structures for the underlying dynamical system. Similar to the applications in computer vision, we find that PGD scales well and can address high-dimensional system states and large-scale NN policies. When the unknown mapping from the plant state to the image pixels and complex coupling between states at different time steps are involved, we propose a finite difference PGD estimation which is general in the sense that it only requires a black-box simulator for generating the trajectories of the closed-loop systems given any initial state. Consequently, the proposed method can address the three challenges mentioned above simultaneously. Finally, we present some numerical experiments as well as some concluding remarks.

II. Problem Formulation

In this letter, we are interested in ROA analysis of the following nonlinear dynamical system

$$\begin{aligned} x_{t+1} &= f(x_t, u_t) \\ y_t &= h(x_t) \end{aligned} \tag{1}$$

where $x_t \in \mathbb{R}^{n_x}$ is the state, $u_t \in \mathbb{R}^{n_u}$ is the control input, and $y_t \in \mathbb{R}^{n_y}$ is the output observation. Here, u_t is determined by a complex nonlinear mapping K from the history of observation-action pairs over a time window, i.e., $u_t = K(y_t, y_{t-1}, u_{t-1}, \ldots, y_{t-N+1}, u_{t-N+1})$ where N is the window length. We allow the analytical form of the function h to be unknown and hence y_t is allowed to be a high-dimensional, rich observation from a camera. In this case, y_t is just a vector augmented from the image pixel values obtained at time t, and we make the assumption that the environment for the image generation is relatively static such that h is deterministic. For perception-based control systems, the analytical form of h is unknown. However, it is reasonable to assume that we can query a simulator which can render the perception-based observation output of h for a given state (e.g., a simulated rgb depth camera used in [22]).

We assume that (1) is posed in a way that the equilibrium state is 0. Our goal is to estimate the ROA of the feedback interconnection of (1) and K. For any fixed t, the state x_t of the closed-loop control system (1) with policy K will be uniquely determined by a mapping from the initial state x_0. We denote such a mapping as g_t. Once f, h, and K are fixed, g_t is determined. Then the state trajectory generated by the closed-loop feedback control system satisfies $x_t = g_t(x_0)$ for any t. Now we can define ROA as follows.

Definition 1: The ROA of the feedback control system (1) with the policy K is defined as

$$\mathcal{R} = \{x_0 : \lim_{t\to\infty} g_t(x_0) = 0\}. \tag{2}$$

We are interested in finding convex approximations of $\mathcal{R}$ and addressing two difficult cases. First, system states and DRL-based NN policies can be high-dimensional, causing a scalability issue for performing ROA analysis. Second, for perception-based control systems, it is difficult to figure out the internal mechanism of image generation, and hence the mapping h is typically unknown. Notice that the output of h is typically a high-dimensional signal, and fitting a function to estimate such h for ROA analysis is also difficult. In general, it is very difficult to obtain tight rigorous approximations of $\mathcal{R}$ for high-dimensional nonlinear/perception-based control systems. We will borrow the idea of PGD attack to generate initial conditions which do not belong in $\mathcal{R}$ and then construct ROA approximations using these initial conditions.

III. Main Analysis Framework

A. Approximating ROA via Constrained Maximization

In this letter, we are interested in approximating $\mathcal{R}$ as the following parameterized convex set

$$\hat{\mathcal{R}}(p, r, C) = \{\xi : \|C\xi\|_p \le r\} \tag{3}$$

where C is some prescribed transformation matrix, r quantifies the size of the approximated ROA, and p can be 1, 2, or ∞. Notice that C should be full rank such that $C^\top C$ is a positive definite matrix. When $p = 2$, we will have ellipsoidal approximations. Clearly, the set $\hat{\mathcal{R}}(p, r, C)$ is always convex, and hence projection to $\hat{\mathcal{R}}(p, r, C)$ can be easily done. For convenience, we will drop the subscript "2" in the notation of the ℓ_2 norm and just use $\|\cdot\|$ instead.

If $\hat{\mathcal{R}}(p, r, C) \subset \mathcal{R}$, then we have $\lim_{t\to\infty} g_t(\xi) = 0$ for any $\xi \in \hat{\mathcal{R}}$. Therefore, a necessary and sufficient condition for $\hat{\mathcal{R}}(p, r, C) \subset \mathcal{R}$ is given as follows

$$\max_{\xi\in\hat{\mathcal{R}}(p,r,C)} \left(\limsup_{t\to\infty} \|g_t(\xi)\|^2 \right) = 0. \tag{4}$$

Checking the above condition numerically will lead to a finite-horizon approximation:

$$\max_{\xi\in\hat{\mathcal{R}}(p,r,C)} \|g_T(\xi)\|^2 \le \delta \tag{5}$$

where T is a prescribed large number and δ is some fixed small number. We will use (5) as our main criterion for approximating $\mathcal{R}$. Specifically, given C, p, r, T, and δ, we will calculate $\max_{\xi\in\hat{\mathcal{R}}(p,r,C)} \|g_T(\xi)\|^2$ and compare the resultant solution with δ. For a fixed C and p, we will perform bisection on the radius r to find the maximum of r such that (5) is satisfied. Then the resultant $\hat{\mathcal{R}}(p, r, C)$ will be our ROA approximation. We formalize the above discussion by defining the (T, δ)-approximated region of attraction (AROA) as follows.

Definition 2: The (T, δ)-AROA is defined as

$$\tilde{\mathcal{R}}(T, \delta) = \{x_0 : \|g_T(x_0)\|^2 \le \delta\}.$$

The following lemma gives a precise characterization of the relation between $\tilde{\mathcal{R}}(T, \delta)$ and $\mathcal{R}$.

Proposition 1: For any fixed T and $\delta > 0$, we have

$$\bigcap_{t\ge T} \tilde{\mathcal{R}}(t, \delta) = \{x_0 : \|g_t(x_0)\|^2 \le \delta, \ \ \forall t \ge T\}. \tag{6}$$

The sequence of sets $\{\bigcap_{t\ge T} \tilde{\mathcal{R}}(t, \delta)\}_{T=0}^{\infty}$ is monotonically increasing to $\liminf_{T\to\infty} \tilde{\mathcal{R}}(T, \delta)$. In addition, $\liminf_{T\to\infty} \tilde{\mathcal{R}}(T, \delta)$ is monotonically decreasing in δ, and the following limit holds

$$\mathcal{R} = \lim_{\delta\to 0} \left(\liminf_{T\to\infty} \tilde{\mathcal{R}}(T, \delta) \right). \tag{7}$$

Proof: Most statements in the above lemma can be verified trivially. The proof of (7) is less straightforward and hence included here. First, we will show $\mathcal{R} \subset \lim_{\delta\to 0}(\liminf_{T\to\infty}\tilde{\mathcal{R}}(T,\delta))$. Suppose $x_0 \in \mathcal{R}$. By definition, we have $\lim_{t\to\infty} g_t(x_0) = 0$. Hence for any $\delta_0 > 0$, there exists T such that $\|g_t(x_0)\|^2 \le \delta_0$ for all $t \ge T$. This means $\mathcal{R} \subset \liminf_{T\to\infty}\tilde{\mathcal{R}}(T,\delta_0)$ for any δ_0. Then we can let δ_0 approach 0 and have $\mathcal{R} \subset \lim_{\delta\to 0}(\liminf_{T\to\infty}\tilde{\mathcal{R}}(T,\delta))$. Next, we will show $\lim_{\delta\to 0}(\liminf_{T\to\infty}\tilde{\mathcal{R}}(T,\delta)) \subset \mathcal{R}$. Suppose $x_0 \in \lim_{\delta\to 0}(\liminf_{T\to\infty}\tilde{\mathcal{R}}(T,\delta))$. For any $\delta_0 > 0$, it is straightforward to verify $\lim_{\delta\to 0}(\liminf_{T\to\infty}\tilde{\mathcal{R}}(T,\delta)) \subset \liminf_{T\to\infty}\tilde{\mathcal{R}}(T,\delta_0)$. This means that for any arbitrary $\delta_0 > 0$, there exists $T > 0$ such that $\|g_t(x_0)\|^2 \le \delta_0$ for $t \ge T$. Therefore, we have $\lim_{t\to\infty} g_t(x_0) = 0$. This leads to the desired conclusion and completes the proof. ∎

Based on (7), it is reasonable to estimate $\mathcal{R}$ from $\tilde{R}(T,\delta)$ with some small δ and large T. From a practical point of view, Definition 2 also makes sense since engineering systems are run on finite-time windows. Stability and ROAs defined on infinite horizons provide meaningful abstractions for quantifying the resilience property of many feedback control systems. However, we will show that the finite-horizon notion of ROA will provide complementary benefits from a computational perspective. Our finite-horizon approach will not give a rigorous inner approximation of $\mathcal{R}$. However, if we choose T and δ carefully, our approach will lead to scalable solutions for estimating ROA of complex nonlinear systems and perception-based control. We will elaborate on this later.

B. PGD Attack for ROA Approximation

From the above discussion, the ROA analysis can be formulated as the following maximization problem

$$\underset{\xi\in\hat{\mathcal{R}}(p,r,C)}{\text{maximize}} \|g_T(\xi)\|^2 \tag{8}$$

Denote $\xi^* = \arg\max_{\xi\in\hat{\mathcal{R}}(p,r,C)}\|g_T(\xi)\|^2$. We will perform bisection on r over the domain $[0, r_{max}]$, iteratively solving (8), to find the largest r such that $\|g_T(\xi^*)\|^2 \le \delta$. We will then use the resultant set $\hat{\mathcal{R}}(p,r,C)$ to approximate the ROA. The key to our analysis is that we can apply PGD to solve ξ_r^*. Denote $L_T(\xi) = \|g_T(\xi)\|^2$. In addition, for any convex set S, we use Π_S to denote the projection onto S. Then PGD iterates as[1]

$$\xi^{k+1} = \Pi_{\hat{\mathcal{R}}(p,r,C)}\left(\xi^k + \alpha\nabla L_T(\xi^k)\right). \tag{9}$$

One way to interpret (9) is that at every k, it recursively minimizes $(-L_T(\xi^k) - \nabla L_T(\xi^k)^\top(\xi-\xi^k) + \frac{1}{2\alpha}\|\xi-\xi^k\|^2)$ over the feasible set $\xi \in \hat{\mathcal{R}}(p,r,C)$, where an ℓ_2 regularizer is added to the first-order expansion of $(-L_T)$ around ξ^k.

Another way to solve (8) is to approximate (8) as

$$\xi^{k+1} = \underset{\|C\xi\|_p^2=r^2}{\arg\max}\left\{L_T(\xi^k) + \nabla L_T(\xi^k)^\top(\xi-\xi^k)\right\} \tag{10}$$

where the ℓ_2 regularizer is removed and the inequality constraint $\|C\xi\|_p^2 \le r^2$ is replaced with an equality condition $\|C\xi\|_p^2 = r^2$. Intuitively, using an equality constraint makes sense for the ROA analysis since we will perform a bisection on r. When $p = 2$, the constraint is just $\|C\xi\|^2 = r^2$, and a closed-form solution for (10) is given as

$$\xi^{k+1} = \frac{r}{\left\|C^{-\top}\nabla L_T(\xi^k)\right\|}(C^\top C)^{-1}\nabla L_T(\xi^k) \tag{11}$$

To see this, we apply the Lagrange multiplier theorem to (10) and obtain $-\nabla L_T(\xi^k) + 2\lambda C^\top C\xi = 0$ and $\|C\xi\| = r$, where λ is the Lagrange multiplier. This leads to (11). The above PGD update rule (10) can be viewed as a special case of the so-called Frank-Wolfe (or conditional gradient) algorithm [23]. There is also a connection between (11) and the alignment condition in the controls literature [24], since the initial condition can be viewed as an input applied at $t = 0$.

For $p = 2$, the implementation of (11) is straightforward. For other values of p, (10) can also be applied. Due to the page limit, we only discuss the update rules for these other cases in our `arXiv` report [25]. Another subtle issue is how to choose T. Notice that T cannot be too large. Otherwise the gradient $\nabla L_T(\xi)$ may be too small for any $\xi \in \mathcal{R}$ and this makes finding ξ^* more difficult. Decreasing T is actually smoothing the cost function L_T and makes the optimization easier. Decreasing T will also significantly shorten the computational time. However, we also cannot make T be too small. Otherwise $\tilde{\mathcal{R}}(T,\delta)$ is no longer a good estimate for $\mathcal{R}$. In general, T will be highly dependent on the class of system and may have to be chosen through domain knowledge or experiments. Also note that the choice of C in the update (11) is fixed and can significantly impact the resulting volume of the ROA approximation. If nothing is known a priori about the shape of the ROA, one may tune C heuristically by fitting the shape to sampled points.

Remark 1: Our proposed analysis only provides an approximation for the true ROA. There is a gap between $\mathcal{R}$ and $\tilde{\mathcal{R}}(T,\delta)$. If T and δ are well chosen, the approximation error induced by such a gap will be small. Then we can just verify whether $\hat{\mathcal{R}}(p,r,C) \subset \tilde{\mathcal{R}}(T,\delta)$, and the optimization error in solving (8) will become a more dominant factor. In general, (8) is a non-concave maximization problem. If $\hat{\mathcal{R}}(p,r,C) \not\subset \tilde{\mathcal{R}}$, we can verify this by finding one initial condition satisfying $L_T(\xi) > \delta$. This is relatively easy. However, to verify $\hat{\mathcal{R}}(p,r,C) \subset \tilde{\mathcal{R}}$, we need to find the global maximum of (8) and compare it with δ. There lacks strong guarantees for finding such global solutions. One useful heuristic fix is to run PGD with different random initial conditions. Despite the lack of strong global guarantees, our simulation study shows that the ROA approximations from PGD are good estimates of the true ROAs in many situations.

Both (9) and (11) are inspired by existing attack methods in the adversarial learning literature [19], [21]. For both (9) and (11), the key step is to evaluate $\nabla L_T(\xi^k)$. Next, we discuss how to perform such gradient evaluation for nonlinear systems with NN policies and/or image observations.

C. Model-Based Analysis: PGD With Back-Propagation

As a sanity check, we consider the relative simple case where $u_t = K \circ h(x_t)$ with the analytical forms of both K and h

[1]Technically speaking, projected gradient ascent is needed for maximization problems. However, the terminology PGD is still used here such that our paper is consistent with the adversarial learning literature.

being known apriori. Here the operation $\circ$ denotes the composition of two maps. In this case, we can use back-propagation to evaluate $\nabla L_T(\xi)$ efficiently. The feedback system reduces to the autonomous form $x_{t+1} = f(x_t, K \circ h(x_t))$. For simplicity, we denote $\tilde{f}(x) = f(x, K \circ h(x))$. We also denote $\tilde{f}^{(0)}$ to be the identity map and set $\tilde{f}^{(n+1)} = \tilde{f} \circ \tilde{f}^{(n)}$. Then we have $x_t = \tilde{f}^{(t)}(x_0)$, and $L_T(x_0) = \|\tilde{f}^{(t)}(x_0)\|^2$. We introduce the costate p_t, and define the Hamiltonian as $H(x, p) = p\tilde{f}(x)$. The following result holds.

Proposition 2: Set $x_0 = \xi$ and generate the state sequence as $x_{t+1} = \tilde{f}(x_t)$ for $t = 0, 1, \ldots, T-1$. Next, set $p_T = 2x_T$ and generate the costate in a backward manner, i.e., $p_t = \nabla_x H(x_t, p_{t+1})$ for $0 \le t \le T-1$. Then $\nabla L_T(\xi) = p_0$.

Proof: The above result is a special case of [26, Proposition 5]. It can also be verified using the chain rule. ■

Therefore, one can just run the forward dynamics and then make a back-propagation to calculate the $\nabla L_T(\xi)$. When f, h, and K are known, one can write out explicit expressions for $\nabla_x H$ and perform the gradient calculation accordingly. For perception-based control systems, the analytical form of the mapping h is typically not known, and the dimension of h is high. This motivates us to use a model-free approach.

D. Model-Free ROA Analysis

Now we present a model-free approach to address the case where the analytical forms of f and h are unknown, but a black-box simulator for g_T is available. We assume that we can simulate the closed-loop system and generate $g_T(\xi)$ for any given T and ξ. For perception-based control systems, the dimension of y_t is high. However, a key fact is that $g_T(\xi)$ is a function of ξ which lives in a space of much lower dimension. This motivates us to apply the finite difference estimation for the gradient evaluation. Specifically, we have $\nabla L_T(\xi^k) \approx \Gamma$ where the j-th entry of Γ can be estimated using the following finite difference scheme:

$$\Gamma(j) = \frac{L_T(\xi^k + \epsilon e_j) - L_T(\xi^k)}{\epsilon}. \tag{12}$$

Notice e_j denotes a vector whose entries are all 0 except the j-th entry which is 1. Therefore, we need to simulate the trajectories for (n_x+1) times. For systems with a reasonable state dimension (e.g., up to a thousand), such a gradient evaluation is scalable, in fact, linearly in the horizon T.

The above finite difference method is extremely general. In principle, as long as we can have a black-box simulator for the closed-loop system, we can apply the finite difference method without knowing any underlying dynamic structures. Such an approach is mostly useful for perception-based control systems with a complex mapping h. It is also possible to apply auto-differentiation to obtain the gradient estimate. In general, it is unclear whether the finite-difference technique or the auto-differentiation method gives a better gradient in the context of control, and this topic is still being studied actively [27]. We will investigate this issue in the future.

IV. Experimental Results

We now present numerical results demonstrating the scalability and effectiveness of our proposed PGD-based analysis. We apply the PGD update rule (11) with the finite difference gradient estimator (12) to perform ROA analysis. More discussion on the alternative PGD update rule (9) is given in our `arXiv` report [25]. Finally, we will also justify our approach via a comparison with a Monte Carlo sampling baseline.

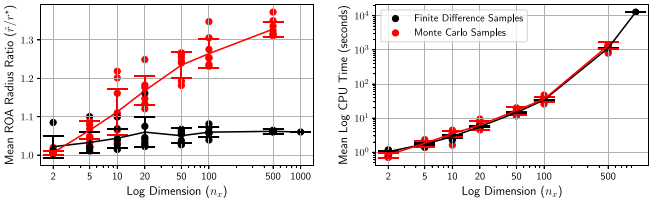

Fig. 1. In this figure we compare the results of our PGD approach and Monte Carlo over random cubic systems over a range of state dimensions. **left**: For each state dimension n_x, we report the mean and one standard deviation over $N = 10$ samples for the spherical ROA approximation found by our method. Here $\hat{r}$ is the estimate and $r^\star$ is the true largest spherical region about the origin that is a subset of the true ellipsoidal ROA. **right**: The mean CPU time is reported with a log-log scaling. Note that only a single sample for $n_x = 1000$ for PGD was taken and no statistics are reported.

A. Cubic Systems With High-Dimensional States

First, we test our proposed method on high-dimensional polynomial systems. We revisit the cubic system example from [28] to showcase the scalability of our proposed method when the system state is high-dimensional. Specifically, we consider the system $x_{t+1} = \tilde{f}(x_t)$, which is discretized from the continuous-time cubic ODE $\dot{x}(t) = (1 - x(t)^\top M x(t)) F x(t)$, where M can be any positive definite (PD) matrix, and $F = -I$. We choose the sampling time as $dt = 10^{-3}$. Therefore, $\tilde{f}$ generates x_{t+1} by solving for the state of the cubic continuous-time ODE at dt starting from initial time 0 and initial state x_t. One can simulate $\tilde{f}$ using the Runge-Kutta method. This cubic example has a known ROA given by the ellipsoid $\mathcal{R} = \{x_0 \in \mathbb{R}^{n_x} : x_0^\top M x_0 < 1\}$, and was used to study the scalability of the sum-of-squares (SOS) technique [28]. It is found that SOS has difficulty when the state dimension exceeds 10, as the number of variables scales exponentially with dimension. In fact, the original results in [28] only cover the case when n_x is up to 8. As shown in Figure 1 (which will be explained in the next paragraph), we demonstrate that our PGD analysis can be applied for state dimensions greater than 1000.

Now we present the details for our scalability study. If we let M be an arbitrary PD matrix and choose $C = I$ in our ROA analysis, then the best inner approximation of the true ROA is given by a sphere of radius $r^\star := \lambda_{\max}^{-1/2}$, where $\lambda_{\max}$ is the largest eigenvalue of M. In this way, we can obtain a spherical ROA ground-truth for systems of arbitrarily high dimension. We can then benchmark our PGD approach by computing the radius $\hat{r}$ of our spherical ROA approximation which can be compared with $r^\star$. Our ROA approximation will be good if $\frac{\hat{r}}{r^*}$ is close to 1. If $\frac{\hat{r}}{r^*}$ is smaller than 1, then we have obtained an inner-approximation of the true spherical ROA. Otherwise our analysis results will be outer approximations of the true ROA. This will allow us to demonstrate high-dimensional state cases that can not be solved by SOS but still have a known ground-truth ROA to compare against.

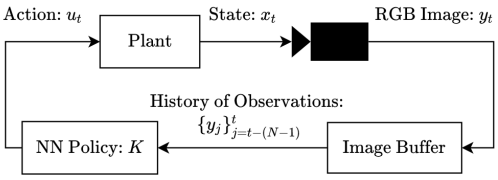

Fig. 2. A diagram depicting the image-based feedback setup.

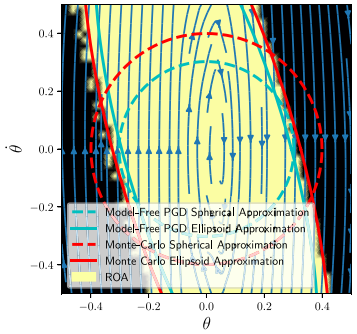

Fig. 3. We compare the ROA spherical and ellipsoidal approximations found by our PGD approach as well as the Monte Carlo baseline on the perception-based single cart-pole system. Shown is a two-dimensional slice of the four-dimensional ROA is determined by fixing the cart position and velocity to be zero. Our approach produces a more accurate ROA estimate when allowing Monte Carlo to use the same number of simulation time-steps.

In principle, any Hurwitz matrix F and PD matrix M may be used for this example as long as (F, M) satisfy the Lyapunov equation $F^\top M + MF = -Q$ for some PD matrix Q. For simplicity and freedom to choose arbitrary ROA parameterization M, we use $F = -I$. For this experiment, we apply our PGD analysis with finite-difference gradient estimates to randomly generated cases with different state dimensions, i.e., $n_x \in \{2, 5, 10, 20, 50, 100, 500, 1000\}$. We fix $T = 10^5$, integrating for 100 seconds with $dt = 10^{-3}$ to ensure the system can converge and $\delta = 10^{-2}$. For each case, we compute the radius $\hat{r}$ and plot the ratio $\frac{\hat{r}}{r^\star}$ in Figure 1, which demonstrates that our approach scales reasonably well as the state dimension increases. For the state-dimensions up to 500, we take 10 randomly generated samples for each dimension. We then take the mean and standard deviations of the approximate radius found by our PGD method, $\hat{r}$, normalized by the best inner approximation of the true ROA $r^\star$. We also run the PGD method for the case of $n_x = 1000$, but only take a single sample for the sake of exposition. This data, along with the mean CPU time taken to run the PGD method on local laptop machine can be found in Figure 1. It is impressive that the analysis for this case can be run on a laptop within four hours.

We also compare our approach with a Monte Carlo-based baseline algorithm which performs the same ROA analysis with a uniform sampling approach. This approach still performs bisection on the ROA radius, but uses the same simulation time-step budget as allowed by PGD for each bisection step. For the cubic system example, we find that our PGD method scales favorable to simply Monte Carlo sampling especially for high-dimensions (> 50).

B. Perception-Based Control Systems

Next, we consider several examples where control actions are directly determined from image pixels (see Figure 2). We present our ROA analysis results for perception-based control of nonlinear cartpole systems (with single and double links). For the full-state feedback case, comparison to an existing quadratic-constraint approach can be found in our `arXiv` report [25], however it is not applicable to this setting. The perception-based feedback control loop is visualized in Figure 2. The plant can be any nonlinear system (e.g., inverted pendulum, cartpole, etc). A camera is used to measure the system output and environment is assumed to be static such that the mapping from the system state to the images is time-invariant. This is a reasonable assumption and similar settings have been adopted previously [22], [29]. The controller uses the last N 84×84 RGB images, which are generated by the *Deepmind Control Suite* [29]. We train each perception-based controller using the novel image-augmentation training procedure from [4], and utilize the model-free RL algorithm Soft-Actor-Critic (SAC) [30] as the policy optimizer. Within the controller K, a policy network is prepended by a four-layer CNN encoder with 3×3 kernels and 32 channels, applying ReLU activations at each CNN layer. The output of the CNN encoder is then fed to the fully-connected four-layer ReLU policy network with 1024 neurons. Despite the complexity of the policy network, we can apply the PGD update rule (11) with the finite difference gradient estimator (12) of ∇L_T to perform ROA analysis.

The k-link cartpole is a useful example since there is an obvious equilibrium point at the upright position which can be studied. For convenience, we use the *Deepmind Control Suite* with its default model parameters to simulate this cartpole example with image observations. To capture the physical limit of the actuators, we set the saturation limit of the control action to be 1.

In this situation, the dimension of the decision variable $\xi = x_0$ is $2(n_{poles} + 1)$ which will be much lower than the dimension of the output variable ($= 2 \times 84^2 \times 3 = 42336$). The model-free approach allows us to estimate ∇L_T without concerning the complexity of the image map h. As long as one can simulate the closed-loop dynamics and generate outputs from any initial state, ∇L_T may be evaluated efficiently in a model-free manner.

The controller is trained to balance the cartpole starting near the equilibrium point. Initializing C by fitting an ellipsoid to 50 ROA sample points, we obtain ROA estimates via the PGD approach (11) for both the resultant perception-based controllers. For both the double and single-link cases, we use $T = 20$ and $\delta = 10^{-1}$ as the PGD hyper-parameters. For the single-link case, we obtain an approximate ROA by running both the model-free PGD method and Monte Carlo to obtain ellipsoid ROA radii of $\hat{r} = 0.96$ and $\hat{r} = 1.07$ respectively. The spherical shape yields a more drastic relative difference of $\hat{r} = 0.3$ and $\hat{r} = 0.4$ respectively. A slice of the ROA's for this experiment can be seen in Figure 3. For the double-link cartpole case, we obtain the spherical ROA radius of $\hat{r} = 0.077$.

Both values are reasonable when compared with the true ROA radius. Though our analysis tends to slightly over-estimate the ROA due to our choice of δ, the performance of the estimate in terms of mean and variance is consistent. This experiment shows that our method is reasonably effective for such perception-based tasks with failure cases that may be addressed with more careful hyper-parameter tuning.

Comparison to the Monte Carlo baseline is also made. One advantage of our PGD approach is that we may take gradient estimates using shorter time horizon trajectories, effectively smoothing the optimization process. This can be seen from Figure 3. For this example, with larger T, Monte Carlo sampling starts to generate comparable results to our PGD approach. This is not that surprising since the system dimension is below 10. However, our PGD approach allows the use of small T, reducing the computational efforts.

V. Concluding Remarks and Future Work

In this letter, we tailor the PGD attack as a general-purpose ROA analysis tool for high-dimensional nonlinear and/or perception-based systems. We reformulate the ROA analysis as a constrained maximization problem, and show that PGD and the model-free variant based on finite difference estimation can be directly applied to solve the resultant optimization problem. An important future task is to extend the PGD attack for input-output gain analysis which is crucial for robust control. Recently, the $\mathcal{H}_\infty$ input-output gain has been used in robust reinforcement learning [31]–[36]. A general-purpose input-output gain analysis will play a crucial role for the developments of robust DRL methods.

References

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

[2] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2015, *arXiv:1506.02438*.

[3] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," 2019, *arXiv:1912.01603*.

[4] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, "Mastering visual continuous control: Improved data-augmented reinforcement learning," 2021, *arXiv:2107.09645*.

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

[6] H. Hu, M. Fazlyab, M. Morari, and G. J. Pappas, "Reach-SDP: Reachability analysis of closed-loop systems with neural network controllers via semidefinite programming," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, 2020, pp. 5929–5934.

[7] M. Jin and J. Lavaei, "Stability-certified reinforcement learning: A control-theoretic perspective," *IEEE Access*, vol. 8, pp. 229086–229100, 2020.

[8] A. Aydinoglu, M. Fazlyab, M. Morari, and M. Posa, "Stability analysis of complementarity systems with neural network controllers," in *Proc. 24th Int. Conf. Hybrid Syst. Comput. Control*, 2021, pp. 1–10.

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

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

[11] S. Chen, M. Fazlyab, M. Morari, G. J. Pappas, and V. M. Preciado, "Learning region of attraction for nonlinear systems," 2021, *arXiv:2110.00731*.

[12] S. M. Richards, F. Berkenkamp, and A. Krause, "The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems," in *Proc. Conf. Robot Learn.*, 2018, pp. 466–476.

[13] Y.-C. Chang, N. Roohi, and S. Gao, "Neural Lyapunov control," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–9.

[14] W. Jin, Z. Wang, Z. Yang, and S. Mou, "Neural certificates for safe control policies," 2020, *arXiv:2006.08465*.

[15] J. Kenanian, A. Balkan, R. M. Jungers, and P. Tabuada, "Data driven stability analysis of black-box switched linear systems," *Automatica*, vol. 109, Nov. 2019, Art. no. 108533.

[16] P. Giesl, B. Hamzi, M. Rasmussen, and K. Webster, "Approximation of Lyapunov functions from noisy data," *J. Comput. Dyn.*, vol. 7, no. 1, p. 57, 2020.

[17] H. Ravanbakhsh and S. Sankaranarayanan, "Learning control Lyapunov functions from counterexamples and demonstrations," *Auton. Robots*, vol. 43, no. 2, pp. 275–307, 2019.

[18] C. Szegedy *et al.*, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[20] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial Intelligence Safety and Security*. Chapman & Hall, 2018, pp. 99–112.

[21] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[22] J. Xu, B. Lee, N. Matni, and D. Jayaraman, "How are learned perception-based controllers impacted by the limits of robust control?" in *Proc. Learn. Dyn. Control*, 2021, pp. 954–966.

[23] J. C. Dunn and S. Harshbarger, "Conditional gradient algorithms with open loop step size rules," *J. Math. Anal. Appl.*, vol. 62, no. 2, pp. 432–444, 1978.

[24] J. E. Tierno, R. M. Murray, J. C. Doyle, and I. M. Gregory, "Numerically efficient robustness analysis of trajectory tracking for nonlinear systems," *J. Guid. Control Dyn.*, vol. 20, no. 4, pp. 640–647, 1997.

[25] A. Havens, D. Keivan, P. Seiler, G. Dullerud, and B. Hu, "Revisiting PGD attacks for stability analysis of large-scale nonlinear systems and perception-based control," 2022, *arXiv:2201.00801*.

[26] Q. Li, L. Chen, and C. Tai, "Maximum principle based algorithms for deep learning," *J. Mach. Learn. Res.*, vol. 18, pp. 1–29, Apr. 2018.

[27] H. Suh, M. Simchowitz, K. Zhang, and R. Tedrake, "Do differentiable simulators give better policy gradients?" 2022, *arXiv:2202.00817*.

[28] W. Tan and A. Packard, "Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming," *IEEE Trans. Autom. Control*, vol. 53, no. 2, pp. 565–571, Mar. 2008.

[29] S. Tunyasuvunakool *et al.*, "DM_control: Software and tasks for continuous control," *Softw. Impacts*, vol. 6, Jun. 2020, Art. no. 100022.

[30] T. Haarnoja *et al.*, "Soft actor–critic algorithms and applications," 2018, *arXiv:1812.05905*.

[31] K. Zhang, B. Hu, and T. Başar, "Policy optimization for $\mathcal{H}_2$ linear control with $\mathcal{H}_\infty$ robustness guarantee: Implicit Regularization and global convergence," in *Proc. Learn. Dyn. Control*, 2020, pp. 179–190.

[32] M. Han, Y. Tian, L. Zhang, J. Wang, and W. Pan, "$\mathcal{H}_\infty$ model-free reinforcement learning with robust stability guarantee," 2019, *arXiv:1911.02875*.

[33] K. Zhang, B. Hu, and T. Başar, "On the stability and convergence of robust adversarial reinforcement learning: A case study on linear quadratic systems," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 22056–22068.

[34] P. L. Donti, M. Roderick, M. Fazlyab, and J. Z. Kolter, "Enforcing robust control guarantees within neural network policies," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–9.

[35] K. Zhang, X. Zhang, B. Hu, and T. Başar, "Derivative-free policy optimization for linear risk-sensitive and robust control design: Implicit Regularization and sample complexity," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 2949–2964.

[36] D. Keivan, A. Havens, P. Seiler, G. Dullerud, and B. Hu, "Model-free μ synthesis via adversarial reinforcement learning," 2021, *arXiv:2111.15537*.