2022 American Control Conference (ACC)
Atlanta, USA, June 8-10, 2022

Computationally Efficient Safe Reinforcement Learning for Power Systems

Daniel Tabas and Baosen Zhang

Abstract— **We propose a computationally efficient approach to safe reinforcement learning (RL) for frequency regulation in power systems with high levels of variable renewable energy resources. The approach draws on set-theoretic control techniques to craft a neural network-based control policy that is guaranteed to satisfy safety-critical state constraints, without needing to solve a model predictive control or projection problem in real time. By exploiting the properties of robust controlled-invariant polytopes, we construct a novel, closed-form "safety-filter" that enables end-to-end safe learning using any policy gradient-based RL algorithm. We then apply the safety filter in conjunction with the deep deterministic policy gradient (DDPG) algorithm to regulate frequency in a modified 9-bus power system, and show that the learned policy is more cost-effective than robust linear feedback control techniques while maintaining the same safety guarantee. We also show that the proposed paradigm outperforms DDPG augmented with constraint violation penalties.**

I. INTRODUCTION

Power systems are a quintessential example of safety-critical infrastructure, in which the violation of operational constraints can lead to large blackouts with high economic and human cost. As variable renewable energy resources are integrated into the grid, it becomes increasingly important to ensure that the system states, such as generator frequencies and bus voltages, remain within a "safe" region defined by the operators [1].

The design of safe controllers concerns the ability to ensure that an uncertain dynamical system will satisfy hard state and action constraints during execution of a control policy [2], [3]. Recently, set-theoretic control [4] has been applied to a wide range of safety-critical problems in power system operation [5], [6]. This approach involves computing a *robust controlled-invariant set* (RCI) along with an associated control policy which is guaranteed to keep the system state inside the RCI [4], [7], [8]. If the RCI is contained in the feasible region of the (safety-critical) state constraints, then the associated control policy is considered to be safe.

However, the set-theoretic approach requires several simplifying assumptions for tractability, leading to controllers with suboptimal performance. First, the disturbances to the system are assumed to be bounded in magnitude but otherwise arbitrary [4], [6]. Second, the RCIs must be restricted to simple geometric objects such as polytopes or ellipsoids [9]. Third, many approaches select an RCI and control policy in tandem, which usually requires the control policy to be linear and forces a tradeoff between performance and robustness [10], [11], [12]. Fourth, nonlinear systems must be treated as linear systems plus an unknown-but-bounded linearization error [5].

Once an RCI is generated using the conservative assumptions listed above, data-driven approaches can use learning to improve performance with respect to the true behavior of the disturbances and nonlinearities without risk of taking unsafe actions [13], [14], [15], [16]. However, these techniques require solving a model predictive control (MPC) or projection problem each time an action is executed, which may be too computationally expensive. Several approaches that avoid repeatedly solving an optimization problem have also been proposed. One such approach involves tracking the vertices of the set of safe actions, and using a neural network to specify an action by choosing convex weights on these vertices. However, this is only possible when the RCI has exceedingly simple geometry [17]. Other strategies only guarantee safety in expectation, and do not rule out constraint violations in every situation [18], [19]. Controllers with Lyapunov stability or robust control guarantees have also been proposed [20], [21], [22], but stability does not always translate to constraint satisfaction.

In this paper, we present a method to design safe, data-driven, and closed-form control policies for frequency regulation in power systems. Our approach combines the advantages of set-theoretic control and learning. In particular, we use simple linear controllers to find a maximal RCI, and then use reinforcement learning (RL) to train a neural network-based controller that improves performance while maintaining safety. The safety of this control policy is accomplished by constraining the output of the neural network to the present set of safe actions. By leveraging the structure of polytopic RCIs, we construct a closed-form *safety filter* to map the neural network's output into the safe action set without solving an MPC or projection problem. The safety filter is differentiable, allowing end-to-end training of the neural network using any policy gradient-based RL algorithm. We demonstrate our proposed control design on a frequency regulation problem in a 9-bus power system model consisting of several generators, loads, and inverter-based resources (IBRs). The simulation results demonstrate that our proposed policy maintains safety and outperforms safe linear controllers without repeatedly solving an optimization problem in real time.

This work is partially supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1762114 and NSF grant ECCS-1930605. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Authors are with the department of Electrical and Computer Engineering, University of Washington, Seattle, WA, United States. {dtabas, zhangbao}@uw.edu.

We focus on applying our algorithm to the problem of primary frequency control in power systems. Frequency is a signal in the grid that indicates the balance of supply and demand. Generators typically respond to the change in frequency by adjusting their power output to bring the frequency back to nominal (e.g., 60 Hz in the North American system) [23], [24]. For conventional generators, these responses are limited to be linear (possibly with a dead-band). In contrast, IBRs such as solar, wind and battery storage can provide almost any desired response to frequency changes, subject to some actuation constraints [25]. Currently, however, these resources still use linear responses, largely because of the difficulty in designing nonlinear control laws. Recently, RL based methods have been introduced in the literature (see, e.g. [26] and the references within). However, most approaches treat safety and constraint satisfaction as soft penalties, and cannot provide any guarantees [26], [27], [28].

The rest of the paper is organized as follows. Section II introduces the power system model and formulates the problem of safety-critical control from a set-theoretic perspective. Section III describes the proposed controller design. Section IV presents simulation results for the modified 9-bus power system.

II. MODEL AND PROBLEM FORMULATION

A. Model assumptions

In this paper we are interested in a linear system with control inputs and disturbances. We write the system evolution as

$$x_{t+1} = Ax_t + Bu_t + Ed_t, \tag{1}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$ and $d_t \in \mathbb{R}^p$ are vectors of the state variables, control inputs, and disturbances at time t. We assume the disturbance d_t is bounded but otherwise can take arbitrary values. More precisely, we assume that d_t lies in a compact set. This boundedness assumption on d_t is fairly general, since it allows the disturbances to capture uncontrolled input into the system, model uncertainties in A, B, and E, and linearization error. For more compact notation, we will sometimes summarize (1) as $x^+ = f(x_t, u_t, d_t)$.

The constraints on inputs are $u_t \in \boldsymbol{U} \subset \mathbb{R}^m$ and $d_t \in \boldsymbol{D} \subset \mathbb{R}^p$ for all t. The sets $\boldsymbol{U}$ and $\boldsymbol{D}$ are assumed to be polytopes, defined as the bounded intersection of a finite number of halfspaces or linear inequalities [29]. Specifically, $\boldsymbol{U}$ and $\boldsymbol{D}$ are defined as

$$\boldsymbol{U} = \{u \in \mathbb{R}^m \mid -\bar{u} \le V_u u \le \bar{u}\} \text{ and} \tag{2}$$

$$\boldsymbol{D} = \{d \in \mathbb{R}^p \mid -\bar{d} \le V_d d \le \bar{d}\}. \tag{3}$$

In safety-critical control problems such as frequency regulation, operators want to keep the system states within hard constraints. For example, frequencies are generally kept within a tenth of a hertz of the nominal frequency and rotor angle deviations are limited for stability considerations [23]. We use the set

$$\boldsymbol{X} = \{x \in \mathbb{R}^n \mid -\bar{x} \le V_x x \le \bar{x}\} \tag{4}$$

to denote the constraints that the state x must satisfy in real-time.

B. Safety-critical control

Because of the presence of disturbances, it may not be possible for the system state to always remain in $\boldsymbol{X}$. Some states close to the boundary of $\boldsymbol{X}$ could be pushed out by a disturbance no matter the control action, while for other states in $\boldsymbol{X}$, there may exist a control action such that no disturbance would push the state outside of the prescribed region. This motivates the definition of a *robust controlled-invariant set*.

Definition 1 (Robust controlled-invariant set (RCI) [4])**.** An RCI is a set $\boldsymbol{S}$ for which there exists a feedback control policy $u_t = \pi_0(x_t) \in \boldsymbol{U}$ ensuring that all system trajectories originating in $\boldsymbol{S}$ will remain in $\boldsymbol{S}$ for all time, under any disturbance sequence $d_t \in \boldsymbol{D}$.

If $\boldsymbol{S}$ is contained in $\boldsymbol{X}$, then π_0 is a safe policy. Often, the goal is to find the policy that maximizes the size of $\boldsymbol{S}$ while being contained in $\boldsymbol{X}$, since it corresponds to making most of the acceptable states safe [9]. In general, this is a difficult problem. Fortunately, if we restrict the policy to be linear, there are many well-studied techniques that have been shown to be successful at producing large safety sets [10], [30], [31], [32].

In this paper, we assume that $\boldsymbol{S}$ is a polytope described by $2r$ linear inequalities, and that π_0 is a linear feedback control policy. Specifically, we assume

$$\boldsymbol{S} = \{x \in \mathbb{R}^n \mid -\bar{s} \le V_s x \le \bar{s}\} \subseteq \boldsymbol{X} \text{ and} \tag{5}$$

$$\pi_0(x) = Kx \tag{6}$$

where $V_s \in \mathbb{R}^{r \times n}$, $\bar{s} \in \mathbb{R}^r$, and $Kx \in \boldsymbol{U}$ for all $x \in \boldsymbol{S}$. For robustness, we choose the largest RCI satisfying (5). The algorithm used for choosing $(\boldsymbol{S}, \pi_0)$ is described in [30]. The algorithm uses a convex relaxation to find an *approximately* maximal RCI $\boldsymbol{S}$ and an associated K as the solution to an SDP. The objective of the SDP is to maximize the volume of the largest inscribed ellipsoid inside $\boldsymbol{S}$.

Of course, a linear policy that maximizes the size of $\boldsymbol{S}$ may not lead to satisfactory control performance. The set $\boldsymbol{S}$ is chosen jointly with the policy π_0, but there could be many policies (not necessarily linear) that keep $\boldsymbol{S}$ robustly invariant. We want to optimize over this class of nonlinear policies to improve the performance of the system. To explore the full range of safe policies, we define the *safe action set at time* t as

$$\Omega(x_t) := \{u_t \in \boldsymbol{U} \mid x_{t+1} \in \boldsymbol{S},\ \forall\ d_t \in \boldsymbol{D}\} \tag{7}$$

where it is assumed that $x_t \in \boldsymbol{S}$. By induction, any policy that chooses actions from $\Omega(x_t)$ is a safe policy [4].

We define the set of safe policies with respect to the RCI $\boldsymbol{S}$ as

$$\Pi := \{\pi : \mathbb{R}^n \to \mathbb{R}^m \mid \pi(x_t) \in \Omega(x_t),\ \forall\ x_t \in \boldsymbol{S}\}. \tag{8}$$

Given $\boldsymbol{S}$, we search for a policy by optimizing over Π:

$$\min_{\pi\in\Pi} \mathbb{E}_{x_0\in\boldsymbol{S},d_t\in\boldsymbol{D}}\left[\frac{1}{T}\sum_{t=1}^{T} J(x_t,u_t)\right] \tag{9a}$$

$$\text{subject to: } x_{t+1} = Ax_t + Bu_t + Ed_t \tag{9b}$$

$$u_t = \pi(x_t) \tag{9c}$$

where $\mathbb{E}_{x_0\in\boldsymbol{S},d_t\in\boldsymbol{D}}$ is the expectation with respect to randomness in initial conditions and in the sequence of disturbances, and $J(x_t,u_t)$ is the cost associated with occupying state x_t and taking action u_t. To estimate (9a), d_t is sampled from $\boldsymbol{D}$ but treated as stochastic, so that standard RL algorithms can be used to solve (9) [13]. This relaxation does not require thorough sampling of $\boldsymbol{D}$ to preserve safety, since the constraint $\pi \in \Pi$ imposes state and input constraint satisfaction for *all* possible disturbances $d_t \in \boldsymbol{D}$. The solution of (9) depends on the distribution of d_t over $\boldsymbol{D}$ but safety is guaranteed for any $\pi \in \Pi$.

One example of a cost function that can be used in (9) is the classical LQR cost on state and control [33], but other non-quadratic cost functions can also be used. For example, for sparsity-promoting controllers, we may set $J(x_t,u_t) = x_t^TQx_t + c\|u_t\|_1$, where $Q \succeq 0$ and $c > 0$ [34].

III. CONTROLLER DESIGN

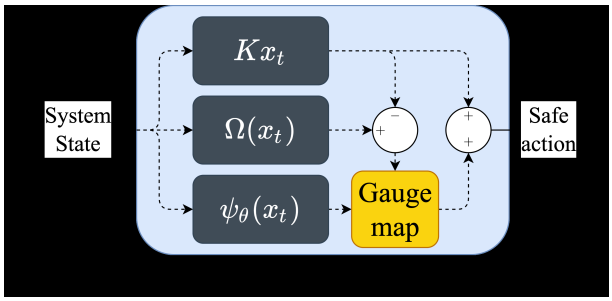

Fig. 1. Policy network architecture for safe learning. The components are: Kx_t, a safe linear feedback included for numerical stability; $\Omega(x_t)$, the set of safe actions from observed state x_t; $\psi_\theta(x_t)$, a neural network; and the closed-form *gauge map* which maps neural network outputs to the current set of safe actions $\Omega(x_t)$.

In this section, we describe how set-theoretic control techniques can be used to create a safety guarantee for data-driven controllers without solving an MPC or projection problem in real time. Since d_t is unknown, data-driven approaches for choosing π are appropriate if safety guarantees can be maintained. For control problems with continuous state and action spaces, one class of RL algorithms involves parameterizing π as a neural network or other function approximator and using stochastic optimization to search over the parameters of that function class for a (locally) optimal policy.

A common approach to safety-critical control with RL is to combine a model-predictive controller with a neural network providing an action recommendation or warm start [13], [14]. However, this makes it difficult to search over Π efficiently and leads to control policies with higher computational overhead. One optimization-free approach involves tracking the vertices of $\Omega(x_t)$ and using a neural network to choose convex weights on the vertices of $\Omega(x_t)$. However, this is only possible when $\boldsymbol{S}$ has exceedingly simple geometry [17]. While it is difficult to constrain the output of a neural network to arbitrary polytopes such as $\Omega(x_t)$, it is easy to constrain the output to $\mathbb{B}_\infty$, the ∞-norm unit ball in $\mathbb{R}^m$, using activation functions like sigmoid or hyperbolic tangent in the output layer. By establishing a correspondence between points in $\mathbb{B}_\infty$ and points in $\Omega(x_t)$, we will use neural network-based controllers to parameterize Π.

In particular, we construct a class of safe, differentiable, and closed-form policies π_θ, parameterized by θ, that can approximate any policy in Π. The policy first chooses a "virtual" action in $\mathbb{B}_\infty$ using a neural network ψ_θ. The policy then uses a novel, closed-form, differentiable "safety filter" to equate $\psi_\theta(x_t)$ with an action in $\Omega(x_t)$. Figure 1 illustrates the way ψ_θ, Ω, and π_0 are interconnected using a novel *gauge map* in order to form the policy π_θ. In order to efficiently map between $\mathbb{B}_\infty$ and $\Omega(x_t)$, we now introduce the concepts of *C-sets* and *gauge functions*.

Definition 2 (C-set [4])**.** A *C-set* is a set that is convex and compact and that contains the origin as an interior point.

Any C-set can be used as a "measuring stick" in a way that generalizes the notion of a vector norm [4]. In particular, the gauge function (or Minkowski function) of a vector $v \in \mathbb{R}^m$ with respect to a C-set $\boldsymbol{Q} \subset \mathbb{R}^m$ is given by

$$\gamma_{\boldsymbol{Q}}(v) = \inf\{\lambda \geq 0 \mid v \in \lambda\boldsymbol{Q}\}. \tag{10}$$

If $\boldsymbol{Q}$ is a polytopic C-set defined by $\{w \in \mathbb{R}^m \mid F_i^Tw \leq g_i, i = 1,\dots,r\}$, then the gauge function is given by

$$\gamma_{\boldsymbol{Q}}(v) = \max_i \left\{\frac{F_i^Tv}{g_i}\right\}, \tag{11}$$

which is easy to compute since it is simply the maximum over r elements. Equation (11) is derived in Appendix A. We will use (11) to construct a closed-form, differentiable bijection between $\mathbb{B}_\infty$ and $\Omega(x_t)$.

A. Gauge map

We will first show how to use the gauge function to construct a bijection from $\mathbb{B}_\infty$ to any C-set $\boldsymbol{Q}$, and will then generalize to the case when $\boldsymbol{Q}$ does not contain the origin as an interior point. For any $v \in \mathbb{B}_\infty$, we define the *gauge map from* $\mathbb{B}_\infty$ *to* $\boldsymbol{Q}$ as

$$G(v|\boldsymbol{Q}) = \frac{\|v\|_\infty}{\gamma_{\boldsymbol{Q}}(v)} \cdot v. \tag{12}$$

Lemma 1. *For any C-set* $\boldsymbol{Q}$*, the gauge map* $G : \mathbb{B}_\infty \to \boldsymbol{Q}$ *is a bijection. Specifically,* $w = G(v|\boldsymbol{Q})$ *if and only if* w *and* v *have the same direction and* $\gamma_{\boldsymbol{Q}}(w) = \|v\|_\infty$.

The proof of Lemma 1 is provided in Appendix C. By Lemma 1, choosing a point in $\mathbb{B}_\infty$ is equivalent to choosing a point in $\boldsymbol{Q}$. The action of the gauge map is illustrated in Figure 2.

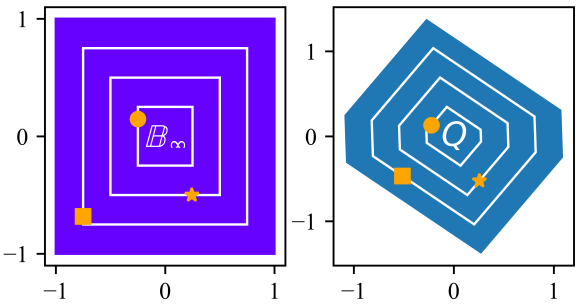

Fig. 2. Action of the gauge map from $\mathbb{B}_\infty$ to randomly generated $\boldsymbol{Q}$, with the $\frac{1}{4}$, $\frac{1}{2}$, and $\frac{3}{4}$ level sets of the respective gauge functions shown in white. For each point $v \in \mathbb{B}_\infty$ and its image $w \in \boldsymbol{Q}$, v and w have the same direction and $\gamma_{\boldsymbol{Q}}(w) = \|v\|_\infty$.

We cannot directly use the gauge map to convert between points in $\mathbb{B}_\infty$ and points in $\Omega(x_t)$, since $\Omega(x_t)$ may not contain the origin as an interior point. Instead, we must temporarily "shift" $\Omega(x_t)$ by one of its interior points, making it a C-set. Lemma 2 provides an efficient way to achieve this.

Lemma 2. *If $\pi_0(x) = Kx$ is a policy in Π, then for any x_t in the interior of $\boldsymbol{S}$, the set $\hat{\Omega}_t := [\Omega(x_t) - Kx_t]$ is a C-set.*

The proof of Lemma 2 is provided in Appendix D. Figure 3 illustrates the way the gauge map and Lemma 2 are used in the policy network as a safety filter, by transforming the output of the policy network from $\mathbb{B}_\infty$ to $\Omega(x_t)$.

B. Policy architecture

Theorem 1. *Assume the system dynamics and constraints are given by* (1), (2) *and* (3), *and there exists a choice of $(\boldsymbol{S}, \pi_0)$ conforming to* (5) *and* (6). *Let $\psi_\theta : \boldsymbol{S} \to \mathbb{B}_\infty$ be a neural network parameterized by θ. Then for any x_t in the interior of $\boldsymbol{S}$, the policy*

$$\pi_\theta(x_t) := G(\psi_\theta(x_t)|\hat{\Omega}_t) + Kx_t \tag{13}$$

has the following properties.

1) *π_θ is a safe policy.*
2) *π_θ can be computed in closed form.*
3) *π_θ is differentiable at x_t.*
4) *π_θ can approximate any policy in Π.*

We will comment briefly on the last property and leave the proof of Theorem 1 to Appendix E. The ability of π_θ to approximate any policy in Π given proper choice of θ is based on the function approximation properties of ψ_θ [35] and the ability of the gauge map to establish a one-to-one correspondence between points in $\mathbb{B}_\infty$ and actions in $\Omega(x_t)$.

C. Policy optimization through reinforcement learning

We parametrize the search over Π using (13) with parameter θ, and we choose θ to optimize (9) using policy gradient RL algorithms. The policy gradient theorem from reinforcement learning allows one to use past experience to estimate the gradient of the cost function (9a) with respect to θ [36]. This is a standard approach for RL in continuous control tasks [37]. Policy gradient methods require that it be possible to compute the gradient of π_θ with respect to θ. More specifically, G must be differentiable (Thm. 1, part 3) or else the safety filter would have to be treated as an uncertain influence whose behavior must be estimated from data. The parameter θ is randomly initialized at the beginning of the policy gradient algorithm.

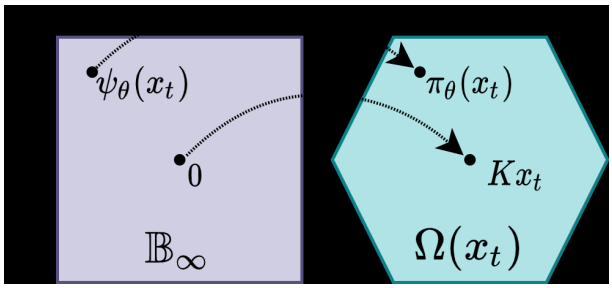

Fig. 3. In the policy network, the gauge map is used to map virtual actions to safe actions.

In addition to being differentiable, π_θ has two other noteworthy attributes. First, under the optimal choice of θ, the controller π_θ performs no worse than π_0. This is because π_0 is a feasible solution to (9), so the optimal solution to (9) can do no worse. Second, unlike projection-based methods [15], the structure of π_θ facilitates exploration of the interior of the safe action set. This is because smooth functions such as the sigmoid or hyperbolic tangent can be used as activation functions in the output layer of ψ_θ to constrain its output to $\mathbb{B}_\infty$. By tuning the steepness of the activation function, it is possible to bias the output of ψ_θ towards or away from the boundary of $\mathbb{B}_\infty$.

IV. SIMULATIONS

A. Power system model

The main application considered in this paper is frequency control in power systems. We consider a system with N synchronous electric generators. The standard linearized swing equation at generator i is:

$$\dot{\delta}_i = \omega_i \tag{14a}$$

$$M_i\dot{\omega}_i = -D_i\omega_i - \sum_{j=1}^{N} K_{ij}(\delta_i - \delta_j) + \sum_{k=1}^{m} b_{ik}u_k - \sum_{l=1}^{p} e_{il}d_l, \tag{14b}$$

where δ_i is the rotor angle, ω_i is the frequency deviation, and M_i and D_i are the inertia and damping coefficients of generator i. The coefficients K_{ij}, b_{ik}, and e_{il} are based on generator and transmission line parameters taken from a modified IEEE 9-bus test case, and are computed by solving the DC power flow equations. Thus, the size of the coefficient measures the influence of each element on the dynamics of generator i. The quantity u_k represents controller k, an IBR such as a battery energy storage system or wind turbine [38], [39], where the active power injections can be controlled in response to a change in system frequency. The feasible

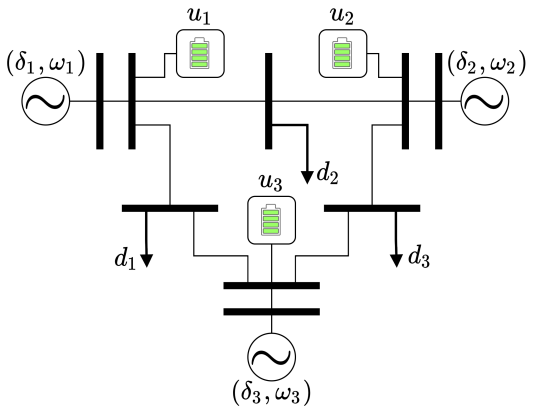

Fig. 4. Illustration of 9-bus power system model.

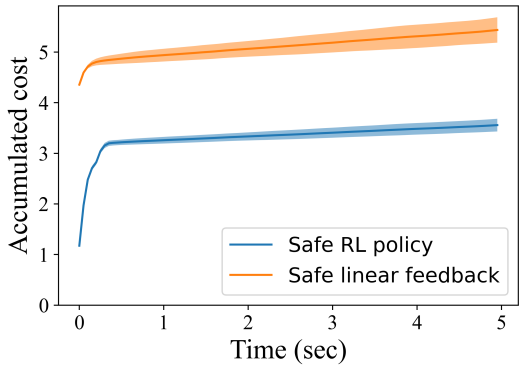

Fig. 5. Accumulated cost over several test trajectories with a fixed initial condition and randomly generated disturbance sequences, showing that the RCI policy network achieves better performance than the safe linear feedback.

control set $\boldsymbol{U} \subset \mathbb{R}^m$ represents limits on power output for each of the m IBRs.

The disturbance d_l captures the uncertainties both in load and in uncontrolled renewable resources. It is also possible to use d to account for parameteric uncertainties, linearization error associated with the linearized swing equation dynamics, or error associated with the DC power flow approximation, by adding virtual disturbances at every bus in the system [7], [40]. The disturbance set $\boldsymbol{D} \subset \mathbb{R}^p$ is conservatively estimated from the capacity of the p uncontrolled elements [41].

Discretizing the continuous-time system in (14) and assembling block components gives a system in the form of (1). Let δ and $\omega \in \mathbb{R}^N$ be vectors representing the rotor angles and frequency deviations of all generators in the system, and let the system state be represented by $x = \begin{bmatrix}\delta & \omega\end{bmatrix}^T \in \mathbb{R}^n$ where $n = 2N$. Using time step τ, the discrete-time system matrices are given by

$$A = \begin{bmatrix} I & \tau I \\ -\tau M^{-1}K & I - \tau M^{-1}D \end{bmatrix},$$
$$B = \begin{bmatrix} 0 \\ M^{-1}\hat{B} \end{bmatrix}, E = \begin{bmatrix} 0 \\ M^{-1}\hat{E} \end{bmatrix}$$

where $[M]_{ii} = M_i$, $[D]_{ii} = D_i$, $[K]_{ij} = K_{ij}$, $[\hat{B}]_{ik} = b_{ik}$, and $[\hat{E}]_{il} = e_{il}$.

We simulate the proposed policy network architecture on a 9-bus power system consisting of three synchronous electric generators, three controllable IBRs, and three uncontrolled loads. The time step for discretization is 0.05 seconds. The load is modeled as autoregressive noise defined by

$$d_{t+1} = \alpha d_t + (1 - \alpha)\hat{d} \tag{15}$$

where $\hat{d} \in \mathbb{R}^p$ is randomly generated from a uniform distribution over $\boldsymbol{D}$, and $\alpha \in (0, 1)$. The system is illustrated in Figure 4.

B. RL algorithm

To train the policy network, we used the Deep Deterministic Policy Gradient (DDPG) algorithm [37], an algorithm well-suited for RL in continuous control tasks. DDPG is an actor-critic algorithm, in which the "actor" or policy chooses actions based on the state of the system, and the "critic" predicts the value of state-action pairs in order to estimate the gradient of the cost function (9a) with respect to θ (the "policy gradient"). In our simulations, the cost was given by

$$J(x_t, u_t) = x^T Q x + u^T R u \tag{16}$$

where $Q = \textbf{block diag}\{1000I_N, 10I_N\}$, $R = 5I_m$, and I_N is the identity matrix in $\mathbb{R}^{N\times N}$. The actor was given by (13). The function ψ_θ was parameterized by a neural network with two hidden layers of 256 nodes each, with ReLU activation functions in the hidden layers. We use sigmoid functions in the last layer to limit the the outputs to be within $[-1, 1]$. The critic, or value network, had the same hidden layers as ψ_θ but a linear output layer. We trained the system for 200 episodes of 100 time steps each.

C. Benchmark comparisons

To demonstrate the advantages of the proposed policy architecture, we compare against two benchmarks. The first is the linear controller Kx, chosen to maximize the size of the associated RCI. Using the algorithm in [30], we choose $(\boldsymbol{S}, K)$ by solving the optimization problem

$$\max_{\boldsymbol{S}\in\mathcal{S}, K\in\mathbb{R}^{n\times m}} \text{vol}(\boldsymbol{S}) \tag{17a}$$
$$\text{s.t. Invariance: } (A + BK)\boldsymbol{S} \oplus E\boldsymbol{D} \subseteq \boldsymbol{S} \tag{17b}$$
$$\text{Safety: } \boldsymbol{S} \subseteq \boldsymbol{X} \tag{17c}$$
$$\text{Control bounds: } K\boldsymbol{S} \subseteq \boldsymbol{U} \tag{17d}$$

where $\oplus$ denotes Minkowski set addition and $\mathcal{S}$ is a class of polytopes described by (5). Figure 5 displays the accumulated cost during a number of test trajectories, showing that π_θ is a more cost-effective controller than Kx when the same $\boldsymbol{S}$ is used for each policy. This makes sense, since the nonlinear policy is afforded additional flexibility in balancing performance and robustness. Since π_θ and Kx are both policies in Π, the learned policy has the same safety guarantees as the linear policy.

The second benchmark is a policy network that is trained using DDPG augmented with a soft penalty on constraint

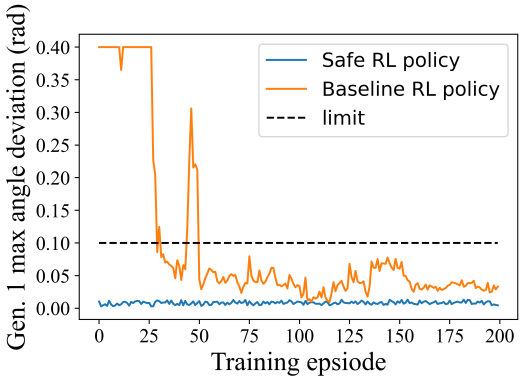

Fig. 6. Maximum angle deviation per training episode for the safe policy network (blue) and the baseline policy network with soft penalty (orange). The safe policy network guarantees safety during training, while soft penalties eventually drive the baseline policy towards constraint satisfaction.

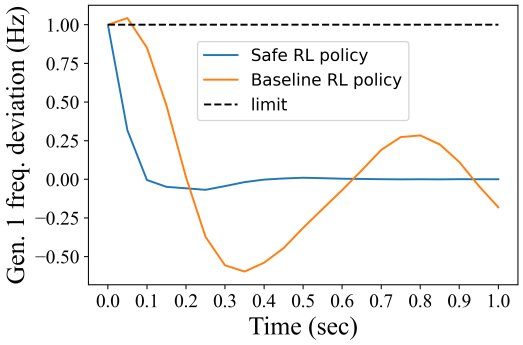

Fig. 7. Even though soft penalties succeed in driving policies to be safe during training, they do not necessarily provide safety during testing. In this example, a policy network that was safe during training (Fig. 6) still exhibits constraint violations during testing. In contrast, starting from the same initial conditions and subject to the same disturbance sequence, the proposed safe policy network guarantees constraint satisfaction.

violations, in order to incentivize remaining in $\boldsymbol{X}$. The policy network for this benchmark consists of two 256-node hidden layers with ReLU activation, and hyperbolic tangent activation functions in the output layer that clamp the output to the box-shaped set $\boldsymbol{U}$. The soft penalty is the total constraint violation, calculated as

$$\lambda \| \max\{V_x x_t - \bar{x}, 0\} - \min\{V_x x_t + \bar{x}, 0\} \|_1$$

where the max and min are taken elementwise, and $\lambda > 0$.

In Fig. 6, we plot an example of the maximum angle deviation during training. We place a hard constraint of 0.1 radians on this angle deviation. For the policy given by (13), the trajectory always stays within this bound, by the design of the the controller. For a policy trained with a soft penalty, trajectories initially exit the safe set. With enough training, the trajectories eventually satisfy the state constraints.

Figure 7 shows that safety in training does not imply safety in testing. The policy network trained using a soft penalty can still result in constraint violations, whereas the safe policy network does not. In some sense, this is not unexpected. Only a limited number of disturbances can be seen during training, and because of the nonlinearity of the neural network-based policy, it is difficult to provide guarantees from the cost alone. In addition, picking the right soft penalty parameter is nontrivial. If the penalty λ is too low, constraint satisfaction will not be incentivized, and if λ is too high, convergence issues may arise [42]. In our experiments, we tuned λ by hand to strike the middle ground, but even automatic, dynamic tuning of λ during training is not guaranteed to prevent constraint violations in all cases [42].

V. CONCLUSIONS

In this paper, we propose an efficient approach to safety-critical, data-driven control. The strategy relies on results from set-theoretic control and convex analysis to provide provable guarantees of constraint satisfaction. Importantly, the proposed policy chooses actions without solving an optimization problem, opening the door to safety-critical control in applications in which computational power may be a bottleneck. We apply the proposed controller to a frequency regulation problem in power systems, but the applications are much more wide-ranging. Future work includes investigating robustness to changes in power system topology and extending the proposed technique to decentralized control.

ACKNOWLEDGMENTS

D. Tabas would like to thank Liyuan Zheng for guidance and Sarah H.Q. Li for helpful discussions.

REFERENCES

[1] Y. Chen, J. Anderson, K. Kalsi, A. D. Ames, and S. H. Low, "Safety-critical control synthesis for network systems with control barrier functions and assume-guarantee contracts," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 487–499, 2021.

[2] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[3] J. Lygeros, "On reachability and minimum cost optimal control," *Automatica*, vol. 40, no. 6, pp. 917–927, 2004.

[4] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Birkhauser, 2015.

[5] A. El-Guindy, Y. C. Chen, and M. Althoff, "Compositional transient stability analysis of power systems via the computation of reachable sets," *Proceedings of the American Control Conference*, pp. 2536–2543, 2017.

[6] Y. Zhang, Y. Li, K. Tomsovic, S. Djouadi, and M. Yue, "Review on Set-Theoretic Methods for Safety Verification and Control of Power System," *IET Energy Systems Integration*, pp. 2–12, 2020.

[7] A. El-Guindy, Y. C. Chen, and M. Althoff, "Compositional transient stability analysis of power systems via the computation of reachable sets," *Proceedings of the American Control Conference*, pp. 2536–2543, 2017.

[8] A. El-Guindy, K. Schaab, B. Schurmann, O. Stursberg, and M. Althoff, "Formal LPV control for transient stability of power systems," *IEEE Power and Energy Society General Meeting*, vol. 2018-Janua, pp. 1–5, 2018.

[9] J. N. Maidens, S. Kaynama, I. M. Mitchell, M. M. Oishi, and G. A. Dumont, "Lagrangian methods for approximating the viability kernel in high-dimensional systems," *Automatica*, vol. 49, no. 7, pp. 2017–2029, 2013.

[10] C. Liu, F. Tahir, and I. M. Jaimoukha, "Full-complexity polytopic robust control invariant sets for uncertain linear discrete-time systems," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 11, pp. 3587–3605, 2019.

[11] F. Blanchini and A. Megretski, "Robust state feedback control of LTV systems: Nonlinear is better than linear," *IEEE Tranactions on Automatic Control*, vol. 44, no. 4, pp. 2347–2352, 1999.
[12] T. Nguyen and F. Jabbari, "Disturbance Attenuation for Systems with Input Saturation: An LMI Approach," Tech. Rep. 4, Department of Mechanical and Aerospace Engineering, University of California, Irvine, 1999.
[13] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.
[14] 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.
[15] S. Gros, M. Zanon, and A. Bemporad, "Safe reinforcement learning via projection on a safe set: How to achieve optimality?," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 8076–8081, 2020.
[16] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *33rd AAAI Conference on Artificial Intelligence*, pp. 3387–3395, 2019.
[17] L. Zheng, Y. Shi, L. J. Ratliff, and B. Zhang, "Safe reinforcement learning of control-affine systems with vertex networks," *arXiv preprint: arXiv 2003.09488v1*, 2020.
[18] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 22–31, PMLR, 06–11 Aug 2017.
[19] M. Yu, Z. Yang, M. Kolar, and Z. Wang, "Convergent policy optimization for safe reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 32, no. NeurIPS, 2019.
[20] W. Cui and B. Zhang, "Reinforcement Learning for Optimal Frequency Control: A Lyapunov Approach," *arXiv preprint: 2009.05654v3*, 2021.
[21] W. Cui and B. Zhang, "Lyapunov-regularized reinforcement learning for power system transient stability," *IEEE Control Systems Letters*, 2021.
[22] P. L. Donti, M. Roderick, M. Fazlyab, and J. Z. Kolter, "Enforcing robust control guarantees within neural network policies," in *International Conference on Learning Representations*, pp. 1–26, 2021.
[23] P. Kundur, N. Balu, and M. Lauby, *Power system stability and control*. New York: McGraw-Hill, 7 ed., 1994.
[24] C. Zhao, U. Topcu, N. Li, and S. Low, "Design and stability of load-side primary frequency control in power systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1177–1189, 2014.
[25] A. Ademola-Idowu and B. Zhang, "Frequency stability using mpc-based inverter power control in low-inertia power systems," *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 1628–1637, 2020.
[26] X. Chen, G. Qu, Y. Tang, S. Low, and N. Li, "Reinforcement Learning for Decision-Making and Control in Power Systems: Tutorial, Review, and Vision," *arXiv preprint: arXiv 2102.01168*, 2021.
[27] Z. Yan and Y. Xu, "Data-driven load frequency control for stochastic power systems: A deep reinforcement learning method with continuous action search," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1653–1656, 2018.
[28] A. Latif, S. S. Hussain, D. C. Das, and T. S. Ustun, "State-of-the-art of controllers and soft computing techniques for regulated load frequency management of single/multi-area traditional and renewable energy based power systems," *Applied Energy*, vol. 266, p. 114858, 2020.
[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.
[30] C. Liu and I. M. Jaimoukha, "The computation of full-complexity polytopic robust control invariant sets," *Proceedings of the IEEE Conference on Decision and Control*, vol. 54rd IEEE, no. Cdc, pp. 6233–6238, 2015.
[31] F. Tahir, "Efficient computation of Robust Positively Invariant sets with linear state-feedback gain as a variable of optimization," *7th International Conference on Electrical Engineering, Computing Science and Automatic Control, 2010*, pp. 199–204, 2010.
[32] T. B. Blanco, M. Cannon, and B. De Moor, "On efficient computation of low-complexity controlled invariant sets for uncertain linear systems," *International Journal of Control*, vol. 83, no. 7, pp. 1339–1346, 2010.
[33] J. P. Hespanha, *Linear systems theory*. Princeton university press, 2018.
[34] F. Dörfler, M. R. Jovanović, M. Chertkov, and F. Bullo, "Sparsity-promoting optimal wide-area control of power networks," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2281–2291, 2014.
[35] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
[36] D. P. Bertsekas, *Reinforcement Learning and Optimal Control*. Athena Scientific, 2019.
[37] 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," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016.
[38] B. Kroposki, B. Johnson, Y. Zhang, V. Gevorgian, P. Denholm, B. M. Hodge, and B. Hannegan, "Achieving a 100% Renewable Grid: Operating Electric Power Systems with Extremely High Levels of Variable Renewable Energy," *IEEE Power and Energy Magazine*, vol. 15, no. 2, pp. 61–73, 2017.
[39] B. Xu, Y. Shi, D. S. Kirschen, and B. Zhang, "Optimal battery participation in frequency regulation markets," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6715–6725, 2018.
[40] J. Machowski, J. W. Bialek, and J. R. Bumby, *Power System Dynamics: Stability and Control*. Wiley, 2 ed., 2008.
[41] Y. C. Chen, X. Jiang, and A. D. Domínguez-García, "Impact of power generation uncertainty on power system static performance," in *NAPS 2011 - 43rd North American Power Symposium*, 2011.
[42] H. Yoo, V. M. Zavala, and J. H. Lee, "A Dynamic Penalty Function Approach for Constraint-Handling in Reinforcement Learning," *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 487–491, 2021.
[43] A. Baydin, A. A. Radul, B. A. Pearlmutter, and J. M. Siskind, "Automatic Differentiation in Machine Learning: a Survey," *Journal of Machine Learning Research*, vol. 18, pp. 1–43, 2018.

APPENDIX

A. Derivation of (11)

Let $\boldsymbol{Q} = \{x \in \mathbb{R}^n \mid Fx \leq g\}$ be a C-set, where $F \in \mathbb{R}^{r\times n}$, $g \in \mathbb{R}^r$, F_i^T denotes the ith row of F, and g_i denotes the ith element of g. The gauge function $\gamma_{\boldsymbol{Q}}(v)$ is computed as follows.

$$\gamma_{\boldsymbol{Q}}(v) = \inf\{\lambda \geq 0 \mid v \in \lambda \boldsymbol{Q}\} \tag{18}$$

$$= \inf\{\lambda \geq 0 \mid \frac{1}{\lambda} F_i^T v \leq g_i, i = 1, \ldots, r\} \tag{19}$$

$$= \inf\{\lambda \geq 0 \mid \lambda \geq \frac{F_i^T v}{g_i}, i = 1, \ldots, r\} \tag{20}$$

$$= \max\{0, \max_i \{\frac{F_i^T v}{g_i}\}\}. \tag{21}$$

We now argue that $\max_i\{\frac{F_i^T v}{g_i}\} \geq 0$. If $F_i^T v < 0$ for all i, then $\boldsymbol{Q}$ is unbounded in the direction of v and $\boldsymbol{Q}$ cannot be a C-set, a contradiction. Further, since $0 \in \mathbf{int}(\boldsymbol{Q})$, it must

hold that $g_i > 0$ for all i. Therefore, there exists i such that $\frac{F_i^T v}{g_i} \geq 0$. □

B. Additional lemmas

The following lemma will be used in the proofs of Lemma 2 and Theorem 1.

Lemma 3. *Under the assumptions of Theorem 1, the safe action set $\Omega(x_t)$ is a polytope for all $x_t \in \boldsymbol{S}$.*

Proof. Starting from (1), (5), and (7), the safe action set is

$$\begin{aligned}\Omega(x_t) &= \{u_t \in \boldsymbol{U} \mid -\bar{s} \leq V_s x_{t+1} \leq \bar{s},\ \forall\ d_t \in \boldsymbol{D}\} \quad (22)\\ &= \{u_t \in \boldsymbol{U} \mid -\bar{s}_i - \min_{d\in\boldsymbol{D}} V_s^{(i)T} E d \\ &\qquad \leq V_s^{(i)T}(Ax_t + Bu_t) \\ &\qquad \leq \bar{s}_i - \max_{d\in\boldsymbol{D}} V_s^{(i)T} E d, \\ &\qquad \forall\ i = 1,\dots,r\} \quad (23)\end{aligned}$$

where $\bar{s}_i$ is the ith element of $\bar{s}$ and $V_s^{(i)T}$ is the ith row of V_s. Since the $\min$ and $\max$ terms evaluate to constant scalars for each i, and since x_t is fixed, (23) is a set of linear inequalities in u_t, making $\Omega(x_t)$ a polytope [4]. □

C. Proof of Lemma 1

We will prove the more general case in which $\mathbb{B}_\infty$ is replaced by any polytopic C-set. Let $\boldsymbol{P}$ and $\boldsymbol{Q}$ be two polytopic C-sets, and define the gauge map from $\boldsymbol{P}$ to $\boldsymbol{Q}$ as $G(v|\boldsymbol{P},\boldsymbol{Q}) = \frac{\gamma_{\boldsymbol{P}}(v)}{\gamma_{\boldsymbol{Q}}(v)} \cdot v$. We will prove that G is a bijection from $\boldsymbol{P}$ to $\boldsymbol{Q}$. The proof is then completed by noting that $\gamma_{\mathbb{B}_\infty}$ is the same as the ∞-norm.

To prove injectivity, we fix $v_1, v_2 \in \boldsymbol{P}$ and show that if $G(v_1|\boldsymbol{P},\boldsymbol{Q}) = G(v_2|\boldsymbol{P},\boldsymbol{Q})$ then $v_1 = v_2$. Assume $G(v_1|\boldsymbol{P},\boldsymbol{Q}) = G(v_2|\boldsymbol{P},\boldsymbol{Q})$. Then v_1 and v_2 must be non-negative scalar multiples of each other, i.e. $v_2 = \beta v_1$ for some $\beta \geq 0$. Making this substitution and applying positive homogeneity of the gauge function [4] yields

$$G(v_2|\boldsymbol{P},\boldsymbol{Q}) = \frac{\gamma_{\boldsymbol{P}}(v_2)}{\gamma_{\boldsymbol{Q}}(v_2)} v_2 = \frac{\gamma_{\boldsymbol{P}}(v_1)}{\gamma_{\boldsymbol{Q}}(v_1)} v_2. \quad (24)$$

Noting that $G(v_1|\boldsymbol{P},\boldsymbol{Q}) = \frac{\gamma_{\boldsymbol{P}}(v_1)}{\gamma_{\boldsymbol{Q}}(v_1)} v_1$, we conclude that $\beta = 1$, thus $v_1 = v_2$.

To prove surjectivity, fix $w \in \boldsymbol{Q}$. We must find $v \in \boldsymbol{P}$ such that $G(v|\boldsymbol{P},\boldsymbol{Q}) = w$. Since $\boldsymbol{P}$ and $\boldsymbol{Q}$ are C-sets, each set contains an open ball around the origin, thus $\boldsymbol{P}$ and $\boldsymbol{Q}$ each contain all directions at sufficiently small magnitude. Choose v in the same direction as w such that $\gamma_{\boldsymbol{P}}(v) = \gamma_{\boldsymbol{Q}}(w)$. Since $w \in \boldsymbol{Q}$, $v \in \boldsymbol{P}$. Then, we have

$$\begin{aligned}G(v|\boldsymbol{P},\boldsymbol{Q}) &= \frac{\gamma_{\boldsymbol{P}}(v)}{\gamma_{\boldsymbol{Q}}(v)} v \quad (25)\\ &= \frac{\gamma_{\boldsymbol{Q}}(w)}{\gamma_{\boldsymbol{Q}}(v)} v. \quad (26)\end{aligned}$$

Since v and w are in the same direction, $\frac{v}{\gamma_{\boldsymbol{Q}}(v)} = \frac{w}{\gamma_{\boldsymbol{Q}}(w)}$. Making this substitution completes the proof. □

D. Proof of Lemma 2

Let **int** and **bd** denote the interior and boundary of a set, and rewrite (23) as $\Omega(x) = \{u \in \mathbb{R}^m \mid Hu \leq h, F(Ax + Bu) \leq g\}$. Fix $x \in \textbf{int}(\boldsymbol{S})$ and let $u^* = Kx$. By Lemma 3, $\Omega(x)$ is convex and compact. To fulfill the properties of a C-set, it remains to show that $u^* \in \textbf{int}(\Omega(x))$. Since $\pi_0 \in \Pi$, $u^* \in \Omega(x)$. Assume for the sake of contradiction that $u^* \in \textbf{bd}(\Omega(x))$. Then either $F_i^T(A+BK)x = g_i$ or $H_j^T K x = h_j$ for some i or j, where the subscript denotes a row index. Suppose without loss of generality that the former holds, i.e. $F_i^T(A+BK)x = g_i$ for some i. Since $x \in \textbf{int}(\boldsymbol{S})$, there exists $\varepsilon \in (0,1)$ and $\alpha = [1+\varepsilon\cdot\textbf{sign}(g_i)]$ such that $y = \alpha x$ is also in $\textbf{int}(\boldsymbol{S})$. The set $\Omega(y)$ is contained in the halfspace $\{u \mid F_i^T(Ay + Bu) \leq g_i\}$. Evaluating this inequality with $u = Ky$, we have $F_i^T(A+BK)y = \alpha F_i^T(A+BK)x = \alpha g_i > g_i$, thus $Ky \notin \Omega(y)$ even though $y \in \boldsymbol{S}$, contradicting the assumption that $\pi_0 \in \Pi$. We conclude that $u^* \notin \textbf{bd}(\Omega(x))$. Since $u^* \in \Omega(x)$, u^* must be an element of $\textbf{int}(\Omega(x))$. □

E. Proof of Theorem 1

1) It suffices to show that the gauge map from $\mathbb{B}_\infty$ to $\hat{\Omega}_t$ is well-defined on $\textbf{int}(\boldsymbol{S})$. This is a direct result of Lemma 2.
2) By Lemmas 2 and 3, $\hat{\Omega}_t$ is a polytopic C-set. By (11), $\gamma_{\hat{\Omega}_t}$ (and π_θ) can be computed in closed form.
3) A standard result from convex analysis shows that the subdifferential of (11) is defined for all $v \in \mathbb{R}^m$, for any polytopic C-set $\boldsymbol{Q}$ [4]. If ψ_θ is a neural network, automatic differentiation techniques can be used to compute a subgradient of π_θ with respect to θ [43].
4) This is due to the fact that ψ_θ is a universal function approximator for functions from $\boldsymbol{S}$ to $\mathbb{B}_\infty$ [35]. By (13) and Lemma 1, π_θ approximates any function in Π.