2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

KCRL: Krasovskii-Constrained Reinforcement Learning with Guaranteed Stability in Nonlinear Discrete-Time Systems

Sahin Lale, Yuanyuan Shi, Guannan Qu, Kamyar Azizzadenesheli, Adam Wierman, Anima Anandkumar

Abstract— **Learning a dynamical system requires stabilizing the unknown dynamics to avoid state blow-ups. However, the standard reinforcement learning (RL) methods lack formal stabilization guarantees, which limits their applicability for the control of real-world dynamical systems. We propose a novel policy optimization method that adopts Krasovskii's family of Lyapunov functions as a stability constraint. We show that solving this stability-constrained optimization problem using a primal-dual approach recovers a stabilizing policy for the underlying system even under modeling error. Combining this method with model learning, we propose a model-based RL framework with formal stability guarantees, Krasovskii-Constrained Reinforcement Learning (KCRL). We theoretically study KCRL with kernel-based feature representation in model learning and provide a sample complexity guarantee to learn a stabilizing controller for the underlying system. Further, we empirically demonstrate the effectiveness of KCRL in learning stabilizing policies in online voltage control of a distributed power system. We show that KCRL stabilizes the system under various real-world solar and electricity demand profiles, whereas standard RL methods often fail to stabilize.**

I. Introduction

Reinforcement Learning (RL) has been recognized as a promising alternative for traditional decision-making and control tasks in engineering systems, e.g. robotics [1], energy systems [2], and transportation [3]. However, despite the promise, major hurdles remain before deployment in such systems is feasible. One of the key challenges is that many real-world systems are safety-critical and have high standards for stability. Even though RL algorithms outperform classical control methods in complex and uncertain dynamical environments, they often do not provide formal stability guarantees outside of simple systems [4]. In particular, most popular RL algorithms for control of nonlinear systems follow model-free gradient-based policies that focus on minimizing the control cost and do not explicitly consider stability [5]. This lack of stability guarantees currently prevents the deployment of RL algorithms in real-world problems, where the dynamics are usually nonlinear and instabilities are costly, e.g., voltage instability in power systems [6].

In contrast, control-theoretic approaches provide a rich set of tools for analyzing the stability of dynamical systems and synthesizing stable control policies. There is a large body of work that focuses on designing stable and robust controllers for linear systems and beyond [7]. Tools like Lyapunov's direct method, contraction (incremental stability) analysis [8]–[10] and passivity theory [11] provide ways to verify stability and synthesize stabilizing controllers for nonlinear dynamical systems. The key challenge in these methods is to find the valid Lyapunov functions to verify stability, which in practice heavily relies on trial and error.

Contributions. In this work, we integrate control theoretic tools into the policy optimization in RL and provide a new model-based RL framework that is guaranteed to design stabilizing controllers in online control of unknown nonlinear discrete-time dynamical systems. In particular, we propose a policy optimization problem that adapts Krasovskii's construction of quadratic Lyapunov functions [12] as a stability constraint, which guarantees that the Lyapunov stability conditions are met by design for the solution of the policy optimization problem (Theorem 1). Further, we show that this stabilization guarantee holds for the controllers obtained using a learned model of system dynamics in the policy optimization for small enough modeling errors (Theorem 2).

To adapt this stability-constrained policy optimization problem into RL, we propose a primal-dual method. We show that the primal-dual method guarantees the satisfaction of the stability constraint and the design of a stabilizing policy for the underlying system after convergence (Theorem 3). This allows us to design a novel model-based RL framework, Krasovskii-Constrained RL (KCRL), via combining model learning and the proposed policy optimization method. KCRL learns the unknown model dynamics in epochs and solves the proposed stability-constrained policy optimization problem via the primal-dual method using the learned model.

We study the KCRL framework both theoretically and empirically. On the theory side, we consider KCRL with kernel-based feature representations for model learning, i.e. Random Fourier Features (RFF) [13]. We show that KCRL with RFF-learning formally guarantees the design of stabilizing control policies in finite time/samples (Theorem 4). On the empirical side, we study the performance of the KCRL framework in learning a stable policy for voltage control in a distributed power system with different operating conditions obtained via real-world operation data. We show that KCRL guarantees stability under all operating conditions, whereas the standard RL methods fail in stabilizing.

II. Related Work

Our work connects to a broad set of control and RL literature.

Lyapunov theory is a systematic framework to analyze the

Sahin Lale, Adam Wierman and Anima Anandkumar are with the Department of Computing and Mathematical Sciences, Caltech alale, adamw, anima@caltech.edu. Yuanyuan Shi is with the Department of Electrical and Computer Engineering, University of California San Diego yyshi@eng.ucsd.edu. Guannan Qu is with the Department of Electrical and Computer Engineering, Carnegie Mellon University gqu@andrew.cmu.edu. Kamyar Azizzadenesheli is with Nvidia Corporation kamyar@nvidia.com

2023 62nd IEEE Conference on Decision and Control (CDC) | 979-8-3503-0124-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CDC49753.2023.10384011

stability of a control system. To prove stability, Lyapunov's direct method aims to define a positive definite function, that decreases along the system trajectory, *i.e.* a Lyapunov function. There is a large body of tools in control such as Krasovskii's method [12], contraction theory [8], feedback linearization [14] and passivity theory [11], which provide ways to construct Lyapunov function candidates and analyze stability of the systems. In our work, we consider Krasovskii's method in designing stabilizing policies for systems with modeling error. In the context of these control theoretic tools, our contributions bridge one of the classical tools in control with policy optimization in RL.

Control Lyapunov functions (CLFs) are popular tools in designing stabilizing controllers and they are also closely related to our framework [15]. In the construction of CLFs, it is often assumed that the system is control-affine, or more generally input-output linearizable [16]. For such systems, the Lyapunov function design problem simplifies due to linearized system dynamics [17]. However, to achieve such input-output linerization, existing works either assume the knowledge of the model dynamics or assume that the CLF constructed for the learned model is *also* a CLF for the underlying system [18]–[21]. In this work, we do not have these assumptions on the system dynamics or the constructed Lyapunov function, which are violated in many practical systems. Instead, we consider nonlinear systems that admit Krasovskii's family of Lyapunov functions and provide an end-to-end RL method, KCRL, which designs stabilizing controllers for the underlying system using model estimates. In particular, we quantify the amount of modeling error that KCRL can tolerate for stabilization.

Model-based RL in dynamical systems has been studied in many recent works due to its superior sample efficiency and interpretable guarantees. The main focus has been on learning the system dynamics and providing performance guarantees in finite-time for both linear [22] (and references within), and nonlinear systems [23]. While deriving these guarantees, the formal finite-time stability guarantees are also derived for linear systems [24]. However, these guarantees have only been *assumed to hold* with a stabilizing oracle for nonlinear systems [23]. Our work provides formal finite-time (sample) stabilization guarantees for nonlinear systems without these assumptions.

Stability Guarantees in Learning-based Control. What we present here is one among many directions on incorporating stability guarantees in learning-based control, with a focus on incorporate stability guarantees for policy optimization (PO) based RL algorithms. For the benefit of readers from both learning and control community, we highlight a few results from this vast and growing literature. Stability of learning-based MPC was established in [25], [26] and followed, for nonlinear systems, by efforts on joint learning of the controller and(or) Lyapunov functions [27]–[31]. [32], [33] studied learning of stability certificates and stable controllers from data, and [34] developed a provably stable data-driven algorithm based on system measurements and prior system knowledge. Another line of work consider incremental stability for nonlinear systems using contraction theory and convex optimization with modeling errors [35], [36]. Different from existing works, we construct the Lyapunov function based on Krasovskii' method (rather than learning the Lyapunov function from scratch or data), and train the policy network to satisfy the stability conditions derived from Krasovskii' method. In addition to incorporate stability guarantees to policy optimization methods in RL, there have been works [37]–[40] that proved stability and convergence for actor-critic based RL methods [37], [38] and Q-learning [40].

III. Preliminaries

A. Control Problem

Consider a discrete-time nonlinear system given as

$$x_{t+1} = f(x_t, u_t), \tag{1}$$

where $x_t \in \mathbb{R}^n$ is the state of the system, $u_t \in \mathbb{R}^p$ is the control input at time-step t. We study the discrete optimal control setting for the system given in (1). Suppose there is a class of controllers $g_\theta(\cdot)$, parameterized by $\theta \in \Theta$. The goal is to design a controller $g_\theta(\cdot)$ that minimizes a control cost,

$$\min_\theta J(\theta) = \sum\nolimits_{t=0}^{\infty} \gamma^t c(x_t, u_t), \tag{2a}$$

$$\text{s.t. } x_{t+1} = f(x_t, u_t), u_t = g_\theta(x_t), \tag{2b}$$

where $c(x,u)$ is the cost and γ is the discounting factor. Note that there are many ways to solve or approximate the policy optimization problem (2). Generally speaking, the procedure is to run gradient methods on the policy parameter θ with step size η, $\theta \leftarrow \theta - \eta \nabla J(\theta)$. To approximate the gradient $\nabla J(\theta)$, one can use sampled trajectories such as REINFORCE or value function approximation such as actor-critic methods. As we are dealing with deterministic policies, one of the most popular choices is the Deep Deterministic Policy Gradient (DDPG) [5], where the policy gradient is approximated by $\nabla J(\theta) \approx \frac{1}{N} \sum_{i \in B} \nabla_u \hat{Q}(x,u)|_{x=x_i, u=g_\theta(x_i)} \nabla_\theta g_\theta(x)|_{x_i}$. Here $\hat{Q}(x,u)$ is the value (critic) network that can be learned via temporal difference learning, $g_\theta(x)$ is the actor network, and $\{x_i\}_{i \in B_J}$ are a batch of samples with batch size $|B_J| = N$ sampled from the replay buffer which stores historical state-action pairs. For further details on DDPG, please refer to [5].

B. Stability

In control systems, stability studies whether the state trajectory of the closed-loop system $x_{t+1} = f(x_t, g_\theta(x_t))$ asymptotically converges to the desired stationary point or a set of stationary points. The following formally defines stability in our context, using the notation $\text{dist}(x,S) := \inf_{y \in S} \|y - x\|$ to denote distance between point x and set S.

Definition 1. *(Asymptotically stable equilibrium) A dynamical system $x_{t+1} = f(x_t, g_\theta(x_t))$ is asymptotically stable around $x^{(e)}$ if $f(x^{(e)}, g_\theta(x^{(e)})) = x^{(e)}$, and further, there exists a region around $x^{(e)}$, $B_\delta(x^{(e)}) = \{x : \|x - x^{(e)}\| \le \delta\}$ such that $\forall x_0 \in B_\delta(x^{(e)})$, we have $\lim_{t \to \infty} \|x_t - x^{(e)}\| = 0$.*

More generally, the following definition considers a set of equilibrium points, where we have used notation $\text{dist}(x,S) := \inf_{y\in S}\|y-x\|$ to denote distance between point x and set S.

Definition 2. *(Asymptotically stable set) A dynamical system $x_{t+1} = f(x_t, g_\theta(x_t))$ is asymptotically stable around set S_e if $f(x^{(e)}, g_\theta(x^{(e)})) = x^{(e)}, \forall x^{(e)} \in S_e$, and further, there exists $B_\delta(S_e) := \{x : \text{dist}(x, S_e) \le \delta\}$ such that $\forall x_0 \in B_\delta(S_e)$, we have $\lim_{t\to\infty} \text{dist}(x_t, S_e) = 0$.*

A common approach to prove stability of a dynamical system with respect to an equilibrium is via Lyapunov's direct method, and a generalization of Lyapunov's method, known as LaSalle's Invariance Principle for proving stability to a set. Both involve defining a positive definite function that decreases along the system trajectory, *i.e.* a Lyapunov function V. Please refer to [7] for a more complete overview.

In this work, we study problem (2) under unknown system dynamics. Note that for $\phi_t = [x_t^\top, u_t^\top]^\top$, one can write the system dynamics given in (1) as

$$x_{t+1} = F(\phi_t), \tag{3}$$

for some nonlinear function F. Further, we denote the closed-loop system dynamics obtained via the policy $u_t = g_\theta(x_t)$ as

$$x_{t+1} = F_\theta(x_t), \tag{4}$$

where $F_\theta(x_t) = F(\phi_t)$ for $\phi_t = [x_t^\top, g_\theta(x_t)^\top]^\top$. To ease the presentation, we use both notations interchangeably throughout this work. Suppose that F and g_θ are both continuously differentiable. Let $G(x,\theta)$ denote the true Jacobian of the closed-loop system with respect to state x, i.e., $G(x,\theta) = \frac{\partial F(\phi)}{\partial x} + \frac{\partial F(\phi)}{\partial u}\frac{\partial u}{\partial x}$. For discrete-time dynamical systems as in (4), Krasovskii's Lyapunov function candidate follows,

$$V(x) = (x - F_\theta(x))^\top M (x - F_\theta(x)), \tag{5}$$

such that there exists a pair (M,θ), where $M \succ 0$ and $G(x,\theta)^\top M G(x,\theta) - M \preceq 0$. In this work, we assume that the underlying system in (3) satisfies the Krasovskii's Lyapunov function construction for an (M,θ) pair with a stability margin, *i.e.*, for some $\bar\epsilon > 0$,

$$G(x,\theta)^\top M G(x,\theta) - M \preceq -\bar\epsilon I. \tag{6}$$

Remark 1. *The stability margin is required to accommodate modeling errors in the dynamics. If one has access to the true model, $F(\cdot)$, $\bar\epsilon = 0$ would suffice, i.e., asymptotic stability.*

In this manuscript, some of the proofs are omitted due to space constraints. The detailed discussions/proofs are in [41].

IV. Krasovskii-Constrained Policy Optimization

In this section, we introduce our novel stability-constrained policy optimization problem and prove that its solution is a stabilizing policy under perfect model dynamics and also under modeling errors. We then provide a primal-dual policy gradient approach to solve this problem using a learned model and show that it finds a stabilizing policy.

A. Stabilizing Policy Design

Using Krasovskii's method of constructing Lyapunov functions for the underlying system described in Section III-B, we add a stability constraint into the standard policy optimization problem in (2). In particular, for a given (estimated) model $\hat F(\cdot)$ on the true system dynamics $F(\cdot)$, we propose to solve the following constrained optimization problem

$$\min_\theta J(\theta) = \sum\nolimits_{t=0}^{T} \gamma^t c(x_t, u_t), \tag{7a}$$

$$\text{s.t. } x_{t+1} = F(\phi_t), u_t = g_\theta(x_t), \tag{7b}$$

$$\hat G(x,\theta)^\top M \hat G(x,\theta) - M \prec -\epsilon_i I, \quad \forall x \in \mathcal{X}, \tag{7c}$$

where $M \succ 0$, $\hat G(x,\theta) = \widehat{\frac{\partial F(\phi)}{\partial x}} + \widehat{\frac{\partial F(\phi)}{\partial u}}\frac{\partial u}{\partial x}$ for the Jacobian estimates $\widehat{\frac{\partial F(\phi)}{\partial x}}$ and $\widehat{\frac{\partial F(\phi)}{\partial u}}$ which can be computed via finite difference method using $\hat F(\cdot)$, and $\bar\epsilon \ge \epsilon_i > 0$, which is chosen based on the modeling error in $\hat F(\cdot)$ as discussed shortly.

Compared to (2), the formulation (7) incorporates an additional constraint (7c). This constraint adapts Krasovskii's method for Lyapunov function construction and enforces the stability of the learned policy. In following, we show that the solution of the novel stability-constrained policy optimization problem (7) using the true system $F(\cdot)$, particularly the true Jacobians in (7c), is a stabilizing policy by design.

Theorem 1 (Stability of the True Discrete-time System)**.** *Consider solving* (7) *with the knowledge of true model $F(\cdot)$, i.e.,* (7c) *is evaluated using the true Jacobian. Let $\theta_\star$ denote the solution of* (7)*, such that* (7c) *holds for some ϵ_i, where $\bar\epsilon \ge \epsilon_i \ge 0$. Then, we have the trajectory of $x_{t+1} = F_{\theta_\star}(x_t)$ is asymptotically stable around the origin, $F_{\theta_\star}(0) = 0$.*

The proof is given in Section A.1 of [41]. The key idea that underpins this result is to use Kowalewki's mean value theorem to show that the difference equation along the closed-loop system trajectory is negative definite for the Lyapunov function candidate (5), due to our stability constraint (7c). Note that Theorem 1 uses the exact Jacobians rather than estimates obtained via the finite difference method.

B. Effect of Modeling Errors

We extend this result to tolerate modeling errors, in particular errors in the Jacobian estimates. First, we quantify some regularity conditions of the system and the policy class.

Assumption 1 (Regularity Conditions)**.** *(i) F is L_F-Lipschitz, i.e., we have Jacobian of F, $\|J_F\| \le L_F$. (ii) $\|\nabla^2 F_i\| \le \text{F}_\text{H}$, $\forall i$, where F_i denotes the mapping from ϕ_t to ith element of state vector x_{t+1}, i.e., $(x_{t+1})_i = F_i(\phi_t)$ for $i = 1, \ldots, n$. (iii) Policies in the policy class are L_u-Lipschitz, that is, $\|\frac{\partial g_\theta(x)}{\partial x}\| \le L_u$, $\forall\theta$.*

Note that in practice, one can use loose upper bounds for these system-related quantities and update them over time. The following shows that solving (7) using a well-refined model estimate $\hat F$ and an appropriate choice of ϵ_i guarantees the recovery of stabilizing policy for the underlying system.

Theorem 2 (Stability under Modeling Error)**.** *Suppose Assumption 1 holds and the Jacobian estimates obtained*

using a model estimate $\hat{F}(\cdot)$ *satisfy* $\sup_x \max(\|\widehat{\frac{\partial F_i(\phi)}{\partial x}} - \frac{\partial F_i(\phi)}{\partial x}\|, \|\widehat{\frac{\partial F_i(\phi)}{\partial u}} - \frac{\partial F_i(\phi)}{\partial u}\|) \le \varepsilon_J < 1$, *for all* $i = 1, \ldots, n$. *Let* $\theta_\star$ *be the solution of* (7) *using these Jacobian estimates in* (7c) *for* ϵ_i, *such that* $\bar{\epsilon} \ge \epsilon_i \ge 2\bar{G}\|M\|(1+L_u)\varepsilon_J + \|M\|(1+L_u)^2\varepsilon_J^2$, *where* $\bar{G} = (1+L_u)(L_F+\varepsilon_J)$. *Then, we have the trajectory of* $x_{t+1} = F_{\theta_\star}(x_t)$ *is asymptotically stable around the origin.*

Proof. By Theorem 1, we only need to show the following: (here we drop θ dependence as it is fixed in the proof)

$$G(x)^\top M G(x) - M \prec 0, \forall x \in \mathcal{X}. \tag{8}$$

Let $\Delta G_i := \hat{G}_i(x) - G_i(x)$. Using Assumption 1 and the construction of $G_i(x)$, we first bound $\|\Delta G_i\|$:

$$\|\Delta G_i\| \le (1+L_u)\varepsilon_J. \tag{9}$$

Next, note that the stability constraint (7c) indicates $\hat{G}_i(x)^\top M \hat{G}_i(x) - M \prec -\epsilon_i I$. Using this, we get

$$\begin{aligned} G_i(x)^\top M G_i(x) &= \hat{G}_i(x)^\top M \hat{G}_i(x) + \Delta G_i^\top M \hat{G}(x) \\ &\quad + \hat{G}_i(x)^\top M \Delta G_i + \Delta G_i^\top M \Delta G_i \\ &\le M - \epsilon_i I + 2\bar{G}\|M\|\|\Delta G_i\| I + \|M\|\|\Delta G_i\|^2 I \prec M \end{aligned}$$

where in the final step, we use (9) and the choice of ϵ_i. This verifies (8) and gives the advertised result. □

Theorems 1 and 2 show that the solution of (7) stabilizes the underlying system even under modeling errors. To use this framework in online policy optimization, one requires to solve this constrained optimization effectively. To that end, we propose a primal-dual policy gradient technique.

C. Primal-Dual Approach

In the following, we describe the primal-dual technique to solve (7) and show that the convergence of this method guarantees the satisfaction of stability condition in (7c) with appropriate algorithmic choices. We use the following shorthand notation, $K(x,\theta) = \hat{G}(x,\theta)^\top M \hat{G}(x,\theta) - M + \epsilon_i I$. With this, (7) can be reformulated as

$$\min_\theta J(\theta) \text{ s.t. } \sup_x \lambda_{\max}(K(x,\theta)) < 0,$$

where $\lambda_{\max}(\cdot)$ is the largest eigenvalue. The Lagrangian for the problem is given as $L(\theta,\mu) = J(\theta) + \mu \sup_x \lambda_{\max}(K(x,\theta))$. The primal-dual algorithm then proceeds as follows [42],

$$\begin{aligned} \theta &\leftarrow \theta - \eta_1 \Big[\nabla J(\theta) + \mu \nabla_\theta \sup_x \lambda_{\max}(K(x,\theta))\Big], \\ \mu &\leftarrow \max(0, \mu + \eta_2 \sup_x \lambda_{\max}(K(x,\theta))). \end{aligned}$$

Since it is not possible to evaluate $\sup_x$, we replace it with a supremum over a batch of representative points in the state space $\{x_i\}_{i\in\mathcal{B}}$. For the term $\nabla J(\theta)$, we use standard policy gradient estimators, e.g. DDPG [5], to evaluate the policy gradient and denote the estimated gradient as $\widehat{\nabla J(\theta)}$. Thus, the primal-dual algorithm is given as,

$$\theta \leftarrow \theta - \eta_1 \Big[\widehat{\nabla J(\theta)} + \mu \nabla_\theta \sup_{i\in\mathcal{B}} \lambda_{\max}(K(x_i,\theta))\Big],$$

$$\mu \leftarrow \max(0, \mu + \eta_2 \sup_{i\in\mathcal{B}} \lambda_{\max}(K(x_i,\theta) + \epsilon_{\text{pd}} I)), \tag{10}$$

where $\eta_1, \eta_2 > 0$ are the step sizes and $\mathcal{B}$ is a batch of representative points in the state space, and $\epsilon_{\text{pd}} > 0$ is a constant that is chosen to tolerate the possible representation incapability of $\mathcal{B}$. The following gives the characterization of ϵ_{pd} to verify that the solution obtained via the primal-dual method stabilizes the underlying system.

Theorem 3 (Primal-Dual Convergence Guarantees Stability)**.** *Suppose the primal-dual procedure converges, then the stability condition will be met for all samples in the batch of representative points* $\mathcal{B}$ *given in* (10). *Suppose the batch* $\mathcal{B} = \{x_i\}_{i=1}^N$ *contains a finite set of points in* $\mathcal{X}$ *such that,* $\forall x \in \mathcal{X}, \exists x_i \in \mathcal{B}, \|x - x_i\| < h$, *for some* $h > 0$. *Under the conditions of Theorem 2, for* $\|\frac{\partial \hat{G}(x,\theta)}{\partial x}\| \le M_G$, *if* ϵ_{pd} *in* (10) *is set to* $\epsilon_{\text{pd}} = 2\bar{G}\|M\|M_G h$, *then the stability condition is met on the entire state space* $\mathcal{X}$ *for the choice of* $\bar{\epsilon} - \epsilon_{\text{pd}} \ge \epsilon_i \ge 2\bar{G}\|M\|(1+L_u)\varepsilon_J + \|M\|(1+L_u)^2\varepsilon_J^2$ *in* (7c).

Remark 2. *The batch* $\mathcal{B}$ *constitutes arbitrary points in* $\mathcal{X}$ *to estimate the supremum of the stability constraint and does not correspond to data collected from the system. The primal-dual algorithm only requires the evaluation of* $\lambda_{\max}(K(x_i,\theta))$ *at these particularly chosen representative points in* $\mathcal{X}$ *using the estimated dynamics* $\hat{F}(\cdot)$. *Here* h *is the fill distance for the batch* $\mathcal{B}$. *This condition can be met by using* $N = (\Gamma/h+1)^d$ *samples in the batch* $\mathcal{B}$. *Note that this dependency is unavoidable to formally verify stability for the entire* $\mathcal{X}$ *using samples [43]. In practice, one can use falsifiers [44] to find states which violate the stability constraint and add them in* $\mathcal{B}$, *similar to [28]. Furthermore, in the expense of computational burden,* N *can be also picked larger which would reduce* h *and shrink* ϵ_{pd} *arbitrarily.*

Proof. The dual variable update in (10) follows $\mu \leftarrow \max(0, \mu + \eta_2 \sup_{i\in\mathcal{B}} \lambda_{\max}(K(x_i,\theta) + \epsilon_{\text{pd}} I))$, and convergence of (μ,θ) to (μ^*,θ^*) implies $\forall i \in \mathcal{B}, \lambda_{\max}(K(x_i,\theta^*))) \le 0$, that is the Krasovskii's stability condition holds for all the samples in the batch. Since each batch $\mathcal{B}$ is drawn from the state space $\mathcal{X}$, as training time goes to infinity, the stability condition $\lambda_{\max}(K(x_i,\theta^*))) \le 0$ also holds for all $x_i \in \mathcal{B}$. By the fill distance condition, *i.e.*, $\forall x \in \mathcal{X}, \exists x_i \in \mathcal{B}, \|x - x_i\| < h$, (here we drop the dependence on θ^* as it is fixed in the proof)

$$\begin{aligned} &\min_{x_i\in\mathcal{B}} \|K(x) - K(x_i)\| = \min_{x_i\in\mathcal{B}} \|\hat{G}(x)^\top M \hat{G}(x) - \hat{G}(x_i)^\top M \hat{G}(x_i)\| \\ &\le \min_{x_i\in\mathcal{B}} \|(\hat{G}(x) - \hat{G}(x_i))^\top M \hat{G}(x)\| + \|\hat{G}(x_i)^\top M (\hat{G}(x) - \hat{G}(x_i))\| \\ &\le 2\bar{G}\|M\|M_G h \end{aligned} \tag{11}$$

Let $\epsilon_{pd} = 2\bar{G}\|M\|M_G h$, if $\forall x_i \in \mathcal{B}, K(x_i,\theta^*) + \epsilon_{pd} I \prec -\epsilon_i I$, then $K(x,\theta^*) \prec -\epsilon_i I$ holds for all x in the entire state space $\mathcal{X}$. By Theorem 2, stability holds for the true system, i.e., $G(x)^\top M G(x) - M \prec 0$ for all $x \in \mathcal{X}$. □

Remark 3. *The convergence of primal-dual algorithm has been shown for linear systems in risk-constrained control [45], [46] and Q-learning [47]. The similar convergence*

Algorithm 1 KCRL

Input: τ, g_{θ_0}, D, λ, ϵ_i, M, μ, η_1, η_2, ϵ_{pd}
for $i = 0, \dots$ **do**
  for $t = i\tau, \dots, (i+1)\tau$ **do**
    Execute $u_t = g_{\theta_i}(x_t)$
    Store $\phi_t = [x_t^\top, u_t^\top]^\top$ and x_{t+1}
  Estimate the model dynamics $\hat{F}_i(\cdot)$ ▷ **Learning**
  Solve (7) for θ_{i+1} using $\hat{F}_i(\cdot)$ via (10)
  Construct $g_{\theta_{i+1}}$ ▷ **Stable Policy Design**

proofs translate to our setting for linear systems. Showing convergence for general nonlinear dynamics requires a new machinery and is beyond the scope of this work.

Theorem 3 shows that solving (7) via primal-dual approach will recover a stabilizing solution within the given parameter space Θ, provided that the estimation error is sufficiently small. The parameter space Θ and its coupling with the underlying system in closed-loop form determine the level of estimation error required in the system dynamics through L_u, G, and M_G. For instance, in the LQR problem restricted to the linear state-feedback policy class Θ, the estimation error of the model parameters necessary for stable policy design depends on the maximum operator norms of the feedback controllers and their corresponding closed-loop matrices in the given parameter set Θ, while $M_G = 0$. In other words, the feasible set of $\theta \in \Theta$ for a fixed estimation error of the linear model parameters is determined by L_u and G, which can be upper-bounded using the continuous differentiability of F and g_θ. In the following, we design a model-based RL framework using the discussed primal-dual approach and show that for smooth dynamical systems it can be used for learning stabilizing controllers from scratch.

V. Krasovskii-Constrained RL Framework

In this section, we present the novel model-based RL framework: KCRL. The algorithm is outlined in Algorithm 1. KCRL works in epochs of length τ, where the controller is during the epoch is fixed. Each epoch consists of two parts: (i) *Model Learning*, where KCRL deploys the current controller in the underlying system to generate trajectories and update the model estimates, (ii) *Stable Policy Design*, where KCRL uses Krasovskii-constrained policy optimization approach to design the new controller for the next epoch.

Each epoch i of KCRL starts with a data collection from the underlying system for τ time-steps with the current controller, $g_{\theta_i}(\cdot)$. In each time step, KCRL takes the action $u_t = g_{\theta_i}(x_t)$, and stores the current state-action pair $\phi_t = [x_t^\top, u_t^\top]^\top$ and the observed next state x_{t+1}. Note that τ is a user-defined parameter and $g_{\theta_0}(\cdot)$ is the initial policy.

At the end of each epoch, KCRL uses all the data gathered to estimate a model of underlying system dynamics $\hat{F}_i(\cdot)$. This estimate can be obtained in various ways within a general supervised learning framework, e.g., through neural networks or system-dependent feature representations. Using neural networks, one can run a variant of gradient descent to update the model estimates. On the other hand, for system-dependent feature representations, one can consider the best linear approximation of the system dynamics on a nonlinear basis such as Random Fourier Features [13], wavelets, or more generally using an atomic norm minimization framework [48]. Once KCRL has a model estimate after the data collection, it aims to recover a stabilizing policy via solving (7) using (10) to obtain the controller for the next epoch.

VI. KCRL with Random Fourier Features (RFF)

In this section, we theoretically analyze a variant of KCRL that uses RFF to learn the system dynamics. In particular, we give a sample complexity result to learn a stabilizing controller for the underlying system. To obtain such a result, we assume that the unknown nonlinear system F lives within Reproducing Kernel Hilbert Spaces $\mathcal{H}$ (RKHS) of infinitely smooth functions defined by a known positive definite continuous kernel $\kappa(\cdot,\cdot)$, e.g. Gaussian kernel. In particular, we assume that each mapping of ϕ_t to the elements of state vector x_{t+1}, *i.e.* $(x_{t+1})_i = F_i(\phi_t)$ for $i = 1, \dots, n$, lives in this RKHS.

Remark 4. *Note that Gaussian kernels are universal kernels such that they can approximate an arbitrary continuous target function uniformly on any compact subset of the input space using possibly infinite kernel evaluations [49]. Therefore, the class of nonlinear dynamics considered in this work constitutes various nonlinear systems.*

Assume that we have an underlying functional h such that $y_i = h(x_i)$ and some collected data pairs $\mathcal{D} = (x_i, y_i)_{i=0}^K$ for $x \in \mathbb{R}^{d_x}$ and $y \in \mathbb{R}$. The kernel methods construct nonlinear models as $\hat{h}(\cdot) = \sum_{i=1}^K \alpha_i \kappa(x_i, \cdot)$, such that α_i are chosen to best represent $\mathcal{D}$ for some kernel $\kappa(\cdot,\cdot)$. However, for the large number of data points, solving for α_i is computationally expensive. Building upon the kernel trick, *i.e.*, kernel evaluations can be written as inner products between infinite dimensional feature representations $\psi(\cdot)$ in $\mathcal{H}$, [13] proposes finite D-dimensional features $z(\cdot)$ to approximate $\psi(\cdot)$: $\hat{h}(\cdot) = \sum_{i=1}^K \alpha_i \langle \psi(x_i), \psi(\cdot) \rangle_{\mathcal{H}} \approx \sum_{i=1}^K \alpha_i z(x_i)^\top z(\cdot)$. These features are termed as Random Fourier Features (RFF) and generated as

$$z(x) \coloneqq \sqrt{\frac{2}{D}} \left[\cos\left(\omega_1^\top x + b_1\right), \dots, \cos\left(\omega_D^\top x + b_D\right) \right]^\top \quad (12)$$

where ω_i are drawn iid from the normalized Fourier transform of the kernel κ, which corresponds to a proper distribution $p(\omega)$, and b_i are drawn iid from the uniform distribution on $[0, 2\pi]$. Furthermore, [13] shows that this method provides an unbiased estimate of κ and the approximation error decays exponentially in D (Claim 1 of [13]), which motivates the use of RFFs in function approximation in practice [23].

Recently, [23] derived theoretical guarantees for using RFF to approximate vector-valued nonlinear functions that belong to $\mathcal{H}$ within a bounded region, e.g. $F(\cdot)$ in (3). They showed that for large enough RFF (D), for the best D-dimensional RFF approximation of F, $\bar{F}(\cdot) = W_*^\top z(\cdot)$, we have

$$\sup_{\|\phi\| \le \Gamma_\phi} \|\bar{F}(\phi) - F(\phi)\| \le \tilde{\mathcal{O}}(1/\sqrt{D}), \quad (13)$$

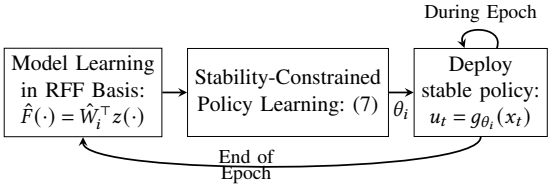

Fig. 1: KCRL Framework with RFF Learning

where Γ_ϕ describes the bounded region. Here $\tilde{O}(\cdot)$ denotes the order up to logarithmic factors and hides the dependencies on n, Γ_ϕ, and the fill distance. Here W_* is the unique min-max optimal model and unique for the particular selection of RFF basis, *i.e.*, a realization obtained via (12). This result is key to our analysis as we use it to derive the finite-time learning and stabilization guarantees of KCRL.

Using the best D-dimensional RFF approximation of F defined in (13), we can approximate (3) as $x_{t+1} \approx W_*^\top z(\phi_t)$, for some unknown $W_* \in \mathbb{R}^{D\times n}$. For model learning, KCRL considers this approximate model and tries to recover the best estimate for W_* using all the data gathered. In particular, after the data collection of epoch i, KCRL solves the following:

$$\min_W \lambda\|W\|_F^2 + \sum\nolimits_{s=0}^{t=(i+1)\tau} \|x_{s+1} - W^\top z(\phi_s)\|_2^2, \qquad (14)$$

for some $\lambda > 0$ to obtain an estimate of W_*. Note that $\hat{W}_i = (Z_t Z_t^\top + \lambda I)^{-1} Z_t X_t^\top$ gives the closed-form solution of (14) for $X_t = [x_{t+1}, \ldots, x_1] \in \mathbb{R}^{n\times(t+1)}$, $Z_t = [z(\phi_t), \ldots, z(\phi_0)] \in \mathbb{R}^{D\times(t+1)}$. Thus, at epoch i, the learned model by KCRL is given by $\hat{F}_i(\cdot) = \hat{W}_i^\top z(\cdot)$. Before we proceed, we have the following assumption on the initial policy of KCRL.

Assumption 2 (Exploratory and Bounded Initial Policy)**.** *The initial controller g_{θ_0} provides persistently exciting (PE) and bounded inputs that can be used for exploration and excite the system uniformly. In other words, the smallest eigenvalue of the design matrix $Z_t Z_t^\top$ scales linearly over time, and for $x_{t+1} = F_{\theta_0}(x_t)$, we have $\|\phi_t\| \le \Gamma_\phi$, for some finite Γ_ϕ.*

These assumptions are standard for consistent estimation of the model dynamics in statistical learning [23]. To achieve such initial controllers, recent tools in control could be deployed [50], [51]. In practice, Dataset Aggregation methods could be used with policy $g_{\theta_0}(\cdot)$ for safe excitement of the systems coupled with randomized feedback policies.

Next, we focus on the learning guarantees of KCRL. We need to guarantee that the model estimation errors are small enough at the end of the first epoch such that the controller obtained via solving (7) would stabilize the system. Using standard least-squares estimation error results for the solution of (14), in particular Theorem 1 and 2 of [23], and under Assumption 2, for large enough D, we get

$$\sup_{\|\phi\|\le\Gamma_\phi} \|F(\phi) - \hat{F}_1(\phi)\| = \tilde{O}(1/\sqrt{D} + \sqrt{D/\tau}), \qquad (15)$$

after τ time-steps, *i.e.*, at the end of first epoch of KCRL. From (15), we derive the following novel finite sample approximation error guarantee on the Jacobian of the underlying function $F(\cdot)$ via the finite difference method. This result could be of independent interest in RFF learning and linearization of RFF-learned model dynamics for the study of different stability notions such as contraction theory or CLFs.

Proposition 1 (Approximation Error of Jacobian using RFF)**.** *Let J_F denote the Jacobian of the underlying system F given in* (3)*. Consider the finite difference approximation of J_F using $\hat{F}_1(\cdot) = \hat{W}_1^\top z(\cdot)$, such that $\hat{J}_F^{(i,j)}(\phi) = \frac{\hat{F}_{1,i}(\phi+\varepsilon\ \mathrm{e_j}) - \hat{F}_{1,i}(\phi-\varepsilon\ \mathrm{e_j})}{2\varepsilon}$, where $\varepsilon > 0$, $\hat{F}_{1,i}(\cdot)$ is the mapping from input to the ith index of the output of $\hat{F}_1$ and $\mathrm{e_j}$ is the jth standard basis. Under Assumptions 1 & 2, for the choice of $\varepsilon = \tilde{O}\big((D^{-1/2} + \sqrt{D/\tau})^{1/3}\big)$, we have that $\sup_{\|\phi\|\le B} \|\hat{J}_F(\phi) - J_F(\phi)\|_F = \tilde{O}(\varepsilon^2)$.*

The proof is given in [41]. It builds upon (15) and uses the Taylor expansions of $F_i(\phi + \varepsilon\ \mathrm{e_j})$ and $F_i(\phi - \varepsilon\ \mathrm{e_j})$ at ϕ. This result shows that the Jacobian of a vector-valued function in a known RKHS is well-approximated using the RFF representation of the function with finite samples. We finally provide the finite-sample stabilization guarantee of KCRL.

Theorem 4 (Finite Sample Stabilization via KCRL)**.** *Suppose Assumptions 1-2 hold and the batch $\mathcal{B}$ is informative enough that its fill distance h satisfies $\bar{\epsilon} - \epsilon_{pd} > 0$, for $\epsilon_{pd} = 2\bar{G}||M||M_G h$. Set $\epsilon_i = \bar{\epsilon} - \epsilon_{pd}$ in the constraint* (7c)*. If* KCRL *uses $D = \tilde{O}\Big(\Big(\frac{2\bar{G}\|M\|(1+L_u)+\|M\|(1+L_u)^2}{\bar{\epsilon}-\epsilon_{pd}}\Big)^3\Big)$ number of RFF in learning the system, after D^2 samples (time-steps), we have the trajectory of $x_{t+1} = f(x_t, g_\theta(x_t))$ is asymptotically stable around the origin, i.e., the solution of* (7) *after $\tau = D^2$ samples from the system gives a stabilizing controller g_{θ_1} for the unknown nonlinear dynamical system.*

Proof. Recall that the stability condition holds for the underlying system with $\bar{\epsilon}$ margin. Thus, combining Theorem 2 and Theorem 3, to guarantee the stabilization of the underlying system for the entire state-space, we require $\epsilon_i \le \bar{\epsilon} - \epsilon_{pd}$, *i.e.*,

$$\frac{\bar{\epsilon} - \epsilon_{pd}}{2\bar{G}\|M\|(1+L_u) + \|M\|(1+L_u)^2} \ge \varepsilon_J, \qquad (16)$$

since $\varepsilon_J < 1$. This gives an upper bound on the error of Jacobian estimates to guarantee stabilization. From Proposition 1, we also have that $\varepsilon_J = \tilde{O}\big((1/\sqrt{D} + \sqrt{D/\tau})^{2/3}\big)$, since $\widehat{\frac{\partial F(\phi)}{\partial x}} - \frac{\partial F(\phi)}{\partial x}$ and $\widehat{\frac{\partial F(\phi)}{\partial u}} - \frac{\partial F(\phi)}{\partial u}$ are submatrices of $\hat{J}_F(\phi) - J_F(\phi)$. The optimal choice of τ and D that minimizes this upper bound is $\tau = D^2$, which results that $\varepsilon_J = \tilde{O}\left(D^{-1/3}\right)$ after τ samples. Thus, for the stated choice of D, after $\tau = D^2$ time-steps, KCRL is guaranteed to stabilize the underlying system. □

This result shows that by setting the epoch length $\tau = D^2$, KCRL guarantees the recovery of a stabilizing controller at the end of first epoch, i.e. g_{θ_1}. The choice of ϵ_i also guarantees the recovery of stabilizing controllers for the subsequent epochs with the non-increasing estimation errors.

VII. Case Study

We numerically study KCRL in learning stable policies for voltage control in a power distribution system [6]. Our case study focuses on the South California Edison

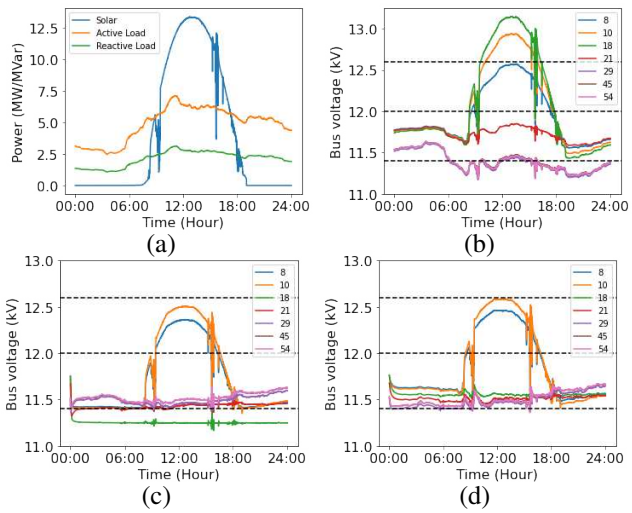

Fig. 2: a) Real-world solar and load data across 24 hours with 6 seconds resolution; b) Serious voltage violations in the system without control; c) Standard DDPG [5] causes voltage violations in some nodes (e.g., node 18); d) KCRL can stabilize the system voltage within the nominal operation region (between the two dashed lines) under all conditions.

56-bus test feeder with high penetration of photovoltaic (PV) generations. The detailed system parameters follow the configuration in [6]. The system model for the voltage control of this system is given by,

$$-p_j(t) = P_{ij}(t) - r_{ij} l_{ij}(t) - \sum_{k:(j,k)\in E} P_{jk}(t), \forall j \tag{17a}$$

$$-q_j(t) = Q_{ij}(t) - x_{ij} l_{ij}(t) - \sum_{k:(j,k)\in E} Q_{jk}(t), \forall j \tag{17b}$$

$$v_j(t) = v_i(t) - 2(r_{ij}P_{ij}(t) + x_{ij}Q_{ij}(t)) + (r_{ij}^2 + x_{ij}^2) l_{ij}(t), \forall (i,j) \in E \tag{17c}$$

Here (17a) and (17b) represent the power conservation at node j, p_j denotes the real power injection at node j and q_j denotes the reactive power injection. (17c) represents the voltage drop from node i to node j. $l_{ij}(t) := |I_{ij}|^2 = (P_{ij}^2 + Q_{ij}^2)/v_i$ is the squared current, $v_i := |V_i|^2$ is the squared voltage, $P_{ij}(t)$ and $Q_{ij}(t)$ represent active and reactive power flow on line (i,j), respectively.

Consider the controller form $q(t+1) = q(t) + g_\theta(v(t))$, where $g_\theta(v(t))$ is represented as a neural network, that can be trained either by the proposed KCRL framework, or standard RL framework. We adopt DDPG [5], a commonly used RL algorithm for continuous control as baseline. [6] shows that a Krasovskii's Lyapunov function exists for the voltage control system (17), where $M = X^{-1}$ with X representing the network reactance matrix. The desired stable set for the system is denote as $\mathcal{S}_v = \{\boldsymbol{v} \in \mathbb{R}^n : \underline{v}_i \le v_i \le \overline{v}_i\}$, where $\underline{v}_i, \overline{v}_i$ are the lower and upper bound for the nominal voltage range. For the considered system, $\overline{v}_i = 12.6$kV $\underline{v}_i = 11.4$kV $\forall i$, that are plotted as the two dashed lines in Figure 2. We simulate the performance of KCRL and DDPG using a real-world voltage control dataset and the results are presented in Figure 2. We also plot the training curve of the KCRL algorithm and the model learning error for the first 100 iterations in Fig. 3. We observe that the model error keeps reducing and the policy performance keeps improving (measured by a lower cost $\sum_{t=0}^{T} \gamma^t c(x_t, u_t)$) per training iteration.

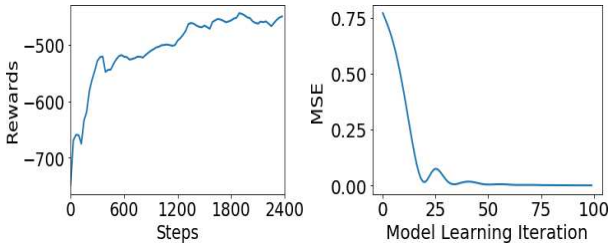

Fig. 3: Model Performance vs Iterations

VIII. Conclusions and Future Work

In this paper, we adapt the classical Krasovskii's Lyapunov function into policy optimization in RL. Using this method, we design stabilizing policies under modeling error with precise robustness guarantees. Furthermore, we propose a model-based RL framework, KCRL, that is guaranteed to design stabilizing controllers in online control of unknown nonlinear dynamical systems using finite samples. In future work, we aim to broaden the KCRL framework to incorporate other Lyapunov function constructions or learn and update the Lyapunov functions on the fly. For example, in closely related contraction analysis, the matrix M can be state-dependent. Extending our current results to such construction would allow KCRL to be deployed for different tasks.

References

[1] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[2] 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*, 2021.

[3] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," *arXiv preprint*, 2017.

[4] G. Qu, Y. Shi, S. Lale, A. Anandkumar, and A. Wierman, "Stable online control of linear time-varying systems," in *Learning for Dynamics and Control*. PMLR, 2021.

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

[6] Y. Shi, G. Qu, S. Low, A. Anandkumar, and A. Wierman, "Stability constrained reinforcement learning for real-time voltage control," *American Control Conference*, 2022.

[7] H. Khalil, *Nonlinear Systems*. Prentice Hall, 2002, vol. 3.

[8] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for nonlinear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.

[9] D. N. Tran, B. S. Rüffer, and C. M. Kellett, "Convergence properties for discrete-time nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3415–3422, 2018.

[10] Y. Kawano and Y. Hosoe, "Contraction analysis of discrete-time stochastic systems," *arXiv preprint arXiv:2106.05635*, 2021.

[11] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[12] N. Krasovskii, "Problems of the theory of stability of motion," 1963.

[13] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, 2007.

[14] M. Krstic, "Feedback linearizability and explicit integrator forwarding controllers for classes of feedforward systems," *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1668–1682, 2004.

[15] E. D. Sontag, "A 'universal' construction of artstein's theorem on nonlinear stabilization," *Systems & control letters*, 1989.

[16] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[17] A. D. Ames, K. Galloway, and J. W. Grizzle, "Control lyapunov functions and hybrid zero dynamics," in *IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 6837–6842.

[18] A. J. Taylor, V. D. Dorobantu, H. M. Le, Y. Yue, and A. D. Ames, "Episodic learning with control lyapunov functions for uncertain robotic systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6878–6884.

[19] J. Umlauft, L. Pöhler, and S. Hirche, "An uncertainty-based control lyapunov approach for control-affine systems modeled by gaussian process," *IEEE Control Systems Letters*, vol. 2, pp. 483–488, 2018.

[20] T. Westenbroek, F. Castañeda, A. Agrawal, S. S. Sastry, and K. Sreenath, "Learning min-norm stabilizing control laws for systems with unknown dynamics," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 737–744.

[21] J. Choi, F. Castaneda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions," 2020.

[22] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, "Finite-time system identification and adaptive control in autoregressive exogenous systems," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 967–979.

[23] ——, "Model learning predictive control in nonlinear dynamical systems," in *60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 757–762.

[24] M. K. S. Faradonbeh, A. Tewari, and G. Michailidis, "Finite-time adaptive stabilization of linear systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3498–3505, 2018.

[25] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109813000678

[26] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. a data-driven control framework," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1883–1896, 2017.

[27] H. Dai, B. Landry, L. Yang, M. Pavone, and R. Tedrake, "Lyapunov-stable neural-network control," 2021. [Online]. Available: https://arxiv.org/abs/2109.14152

[28] Y.-C. Chang, N. Roohi, and S. Gao, "Neural lyapunov control," *Advances in Neural Information Processing Systems*, 2019.

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

[30] ——, "Learning lyapunov functions for hybrid systems," in *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, 2021, pp. 1–11.

[31] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods," *arXiv preprint arXiv:2202.11762*, 2022.

[32] N. Boffi, S. Tu, N. Matni, J.-J. Slotine, and V. Sindhwani, "Learning stability certificates from data," in *Conference on Robot Learning*. PMLR, 2021, pp. 1341–1350.

[33] C. De Persis, M. Rotulo, and P. Tesi, "Learning controllers from data via approximate nonlinearity cancellation," *IEEE Transactions on Automatic Control*, pp. 1–16, 2023.

[34] J. Berberich, C. W. Scherer, and F. Allgöwer, "Combining prior knowledge and data for robust controller design," *IEEE Transactions on Automatic Control*, pp. 1–16, 2022.

[35] H. Tsukamoto and S.-J. Chung, "Neural contraction metrics for robust estimation and control: A convex optimization approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 211–216, 2020.

[36] H. Tsukamoto, S.-J. Chung, and J.-J. Slotine, "Learning-based adaptive control using contraction theory," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 2533–2538.

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

[38] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Mag.*, vol. 32, no. 6, pp. 76–105, 2012.

[39] S. Bhasin, R. Kamalapurkar, M. Johnson, K. Vamvoudakis, F. Lewis, and W. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica (Journal of IFAC)*, vol. 49, no. 1, pp. 82–92, 2013.

[40] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Systems & Control Letters*, vol. 100, pp. 14–20, 2017.

[41] S. Lale, Y. Shi, G. Qu, K. Azizzadenesheli, A. Wierman, and A. Anandkumar, "Kcrl: Krasovskii-constrained reinforcement learning with guaranteed stability in nonlinear dynamical systems," *arXiv preprint arXiv:2206.01704*, 2022.

[42] A. Nedić and A. Ozdaglar, "Subgradient methods for saddle-point problems," *Journal of Optimization Theory and Applications*, vol. 142, no. 1, pp. 205–228, 2009.

[43] P. Giesl, "Construction of a local and global lyapunov function for discrete dynamical systems using radial basis functions," *Journal of Approximation Theory*, vol. 153, no. 2, pp. 184–211, 2008.

[44] S. Gao, J. Avigad, and E. M. Clarke, "δ-complete decision procedures for satisfiability over the reals," in *International Joint Conference on Automated Reasoning*. Springer, 2012, pp. 286–300.

[45] F. Zhao and K. You, "Primal-dual learning for the model-free risk-constrained linear quadratic regulator," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 702–714.

[46] F. Zhao, K. You, and T. Başar, "Global convergence of policy gradient primal-dual methods for risk-constrained lqrs," *IEEE Transactions on Automatic Control*, 2023.

[47] D. Lee and J. Hu, "Primal-dual q-learning framework for lqr design," *IEEE Transactions on Automatic Control*, vol. 64, no. 9, pp. 3756–3763, 2018.

[48] Y. Chi and M. F. Da Costa, "Harnessing sparsity over the continuum: Atomic norm minimization for superresolution," *IEEE Signal Processing Magazine*, vol. 37, no. 2, pp. 39–57, 2020.

[49] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels." *Journal of Machine Learning Research*, vol. 7, no. 12, 2006.

[50] T. Lekang and A. Lamperski, "Sufficient conditions for persistency of excitation with step and relu activation functions," *arXiv preprint arXiv:2209.06286*, 2022.

[51] A. Padoan, G. Scarciotti, and A. Astolfi, "A geometric characterisation of persistently exciting signals generated by continuous-time autonomous systems," *IFAC-PapersOnLine*, pp. 826–831, 2016.