KCRL: Krasovskii-Constrained Reinforcement Learning with Guaranteed Stability in Nonlinear Dynamical 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, current reinforcement learning (RL) methods lack stabilization guarantees, which limits their applicability for the control of safety-critical systems. We propose a model-based RL framework with formal stability guarantees, Krasovskii Constrained RL (KCRL), that adopts Krasovskii's family of Lyapunov functions as a stability constraint. The proposed method learns the system dynamics up to a confidence interval using feature representation, e.g. Random Fourier Features. It then solves a constrained policy optimization problem with a stability constraint based on Krasovskii's method using a primal-dual approach to recover a stabilizing policy. We show that KCRL is guaranteed to learn a stabilizing policy in a finite number of interactions with the underlying unknown system. We also derive the sample complexity upper bound for stabilization of unknown nonlinear dynamical systems via the KCRL framework.**

I. Introduction

Reinforcement Learning (RL) has achieved impressive success in the past decade, highlighted by success stories in game play [1]. As a result, RL has been recognized as a promising alternative for traditional decision-making and control tasks in engineering systems, e.g. robotics [2], energy systems [3], and transportation [4]. 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 generally do not provide formal stability guarantees outside of linear dynamical systems [5], [6]. 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 [7], [8]. 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., frequency instability in power systems [9].

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 [10], [11]. Tools like Lyapunov's direct method, contraction analysis [12] and passivity theory [13] provide ways to verify stability and synthesize stabilizing controllers for nonlinear dynamical systems. The key challenge in these methods is to find the right Lyapunov functions to verify stability, which in practice heavily relies on trial and error. To this end, Krasovskii's approach [14] gives a principled way to explicitly construct Lyapunov functions by considering a quadratic form of the derivative of the state variables.

The common denominator of these control-theoretic methods is that they require a highly accurate model of the system and assume that such a model is known a priori. Recently, there has been a growing interest in incorporating modeling errors in deriving various stability guarantees [15], [16], [17]. However, these works neither provide guarantees in the *online* learning/control setting, nor achieve the desired modeling errors and stability performance in *finite time*, which are the key requirements for deploying learning-based methods in real-world systems.

Contributions. Motivated by the contrast above, we integrate control theoretic tools into RL and provide a finite-time learning to stabilize guarantee in online control of unknown nonlinear dynamical systems. In particular, we propose a model-based RL framework, Krasovskii Constrained RL (KCRL), that adapts Krasovskii's construction of quadratic Lyapunov functions as a stability constraint in the policy optimization problem. This novel stability constraint is a linear matrix inequality type constraint on the Jacobian of the closed-loop system, which guarantees that the Lyapunov stability conditions are met by design (Theorem 1). KCRL learns the unknown model dynamics in epochs via kernel-based feature representations, i.e. Random Fourier Features (RFF) [18], and solves the stability-constrained policy optimization problem via a primal-dual approach using the learned model in the constraint (Figure 1).

To formally guarantee the learning of stabilizing control policies in finite-time, we show that (i) the Jacobian of the underlying closed-loop system can be learned up to a confidence interval using RFF (Proposition 1), (ii) the solution of the policy optimization problem with the stability constraint that uses a learned model still stabilizes the underlying system (Theorem 2), and (iii) the primal-dual approach guarantees the satisfaction of this stability constraint (Theorem 3) after convergence. Combining these results, we provide the number of samples required to learn

[1] Sahin Lale, Adam Wierman and Anima Anandkumar are with the Department of Computing and Mathematical Sciences, Caltech alale, adamw, anima@caltech.edu

[2] Yuanyuan Shi is with the Department of Electrical and Computer Engineering, University of California San Diego yyshi@eng.ucsd.edu

[3] Guannan Qu is with the Department of Electrical and Computer Engineering, Carnegie Mellon University gqu@andrew.cmu.edu

[4] Kamyar Azizzadenesheli is with the Department of Computer Science Purdue University

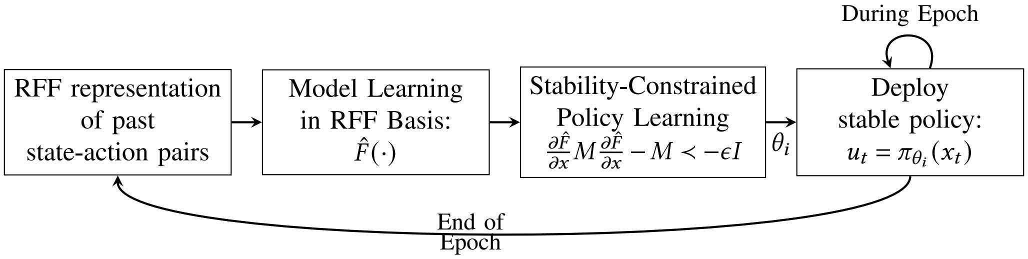
During Epoch
RFF representation of past state-action pairs
Model Learning in RFF Basis: $\hat{F}(\cdot)$
Stability-Constrained Policy Learning $\frac{\partial \hat{F}}{\partial x} M \frac{\partial \hat{F}}{\partial x} - M \prec -\epsilon I$
θ_i
Deploy stable policy: $u_t = \pi_{\theta_i}(x_t)$
End of Epoch

Fig. 1: KCRL Framework

a stabilizing policy for the underlying system in terms of system properties and the number of RFF (Theorem 4).

Due to space constraints, some of the proofs are given as sketches in this manuscript. The details and full proofs can be found in the extended version of this work online.

II. Related Work

This paper connects to a broad set of literature in RL and control.

Model-based RL in dynamical systems. There has been a flurry of studies that consider model-based RL in dynamical systems due to their superior sample efficiency, task generalization and interpretable guarantees [19]. The main focus has been on learning the system dynamics and providing performance guarantees in finite-time for both linear [20] (and references within), and nonlinear systems [21], [22]. While deriving these performance guarantees, the formal finite-time stability guarantees are also derived for linear systems [23]. However, these guarantees have only been *assumed to hold* with a stabilizing oracle for nonlinear systems [21], [22]. Therefore, our work completes an important part of the picture in finite-time stabilization of dynamical systems.

Control Theory. Our work is related to the Krasovskii's method [14] and contraction analysis [12], which can be viewed as a specific way of constructing a Lyapunov function. More broadly, there are a large body of tools in control theory that synthesize controllers that are stable, e.g., passivity theory and feedback linearization [13]. In the context of these works, our contribution can be viewed as bridging one of these classical tools with policy learning in RL, where the system model is unknown.

Lyapunov-based Policy Learning. Lyapunov theory is a systematic framework to analyze the stability of a control system. The core idea is to identify a Lyapunov function of the state, with negative derivative along system trajectories. Using Lyapunov functions in RL was first introduced by [24], but the work did not discuss how to find a candidate Lyapunov function in general. A set of recent works including [25], [26], [27], [28] have attempted to address this challenge by jointly learning the policy and the Lyapunov function. Our goals and approaches are different. We use a principled framework to *construct* (rather than learn) the Lyapunov function, and design the policy learning such that the stability conditions are met. In this sense, [29] is most related to our work. It considers a particular power system model and proposes model-free monotonic policies to stabilize the system. However, their model assumption is specific to power systems, whereas our work considers general nonlinear systems.

Safe RL. The goal of safe RL is to learn a policy that avoids entering unsafe regions, rather than requiring a policy to have attractive or bounded behavior. Various methods have been proposed to ensure safety in RL, such as using control barrier functions [30], Hamilton-Jacobi safety analysis [31], and MPC as a safety filter [32]. We note that safety does not imply stability and vice visa, and our main focus in this paper is to learn an RL policy with the stability guarantee.

III. Preliminaries

In this section, we first describe the control problem and formally define the stability criteria. Then, we discuss the model dynamics and give a brief overview of RFF with their approximation guarantees.

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,

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

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

$$u_t = g_\theta(x_t), \tag{2c}$$

where $c(x,u)$ is the per stage cost and γ is the discounting factor.

Note that there are many ways to solve or approximate the policy minimization 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) [7], 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 network (the critic) that can be learned via temporal difference learning, $g_\theta(x)$ is the actor network, and $\{x_i\}_{i\in B}$ are a batch of samples with batch size $|B| = N$ sampled from the replay buffer which stores historical state-action pairs.

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. However, in general RL formulations such as (2), stability is not an explicit requirement. It is mostly treated as an implicit regularization, since instability usually leads to high (or infinite) costs. The lack of an explicit stability requirement can lead to several issues. During the training phase, the policy may become unstable, causing the training process to terminate. Even after a policy is trained, there is no formal guarantee that the closed loop system is stable, which hinders the learned policy's deployment in real-world engineering systems where there is a very strict requirement of stability. In order to explicitly constrain stability in policy learning, we constrain the search space of policy in the set of stabilizing controllers from Lyapunov stability theory.

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 $\mathrm{dist}(x,S) := \inf_{y\in S} \|y - x\|$ 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 $B_\delta(S_e) := \{x : \mathrm{dist}(x, S_e) \le \delta\}$ such that $\forall x_0 \in B_\delta(S_e)$, we have $\lim_{t\to\infty} \mathrm{dist}(x_t, S_e) = 0$.*

A common approach to proving stability of a dynamical system is via Lyapunov's direct method, which involves defining a positive definite function that decreases along the system trajectory. For a more complete overview of Lyapunov stability theory, please refer to [11, Chapter 3].

C. System Dynamics

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. For this given system, we assume that the unknown nonlinear function F lives within Reproducing Kernel Hilbert Spaces (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 $i = 1, \ldots, n$, lives in this known RKHS.

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 [33]. Therefore, the class of nonlinear dynamics considered in this work, which can be infinite sum of kernel evaluations, constitutes a vast variety of nonlinear systems. In particular, they are more general than the kernelized nonlinear systems, where the system dynamics are assumed to be represented by a finite sum of kernel evaluations [21], [34].

Remark 1. *We consider fully observable (state-feedback) nonlinear systems in this paper. However, the results of this work can be extended to partially observable nonlinear dynamical systems. More specifically, the results can be rewritten for an order-h nonlinear autoregressive exogenous system as depicted in [22], i.e., considering the last h input-output pairs as the state x_t.*

D. RFF and Approximation Guarantees

Kernel methods are foundational tools used in modeling complicated functional relationships in many machine learning problems. The so-called "kernel trick" is the main component of these methods. The kernel trick states that for some positive definite kernel $\kappa(\cdot,\cdot)$, the kernel evaluation of data points x_1 and x_2 are equivalent to inner product between possibly infinite dimensional feature representations $\psi(\cdot)$ of x_1 and x_2 in a Hilbert space $\mathcal{H}$: $\kappa(x_1, x_2) = \langle \psi(x_1), \psi(x_2)\rangle_{\mathcal{H}}$.

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 large number of data points, solving for α is computationally expensive. For this very reason, [18] proposes to approximate $\psi(\cdot)$ with finite D-dimensional features $z(\cdot)$:

$$\hat{h}(\cdot) = \sum_{i=1}^K \alpha_i \langle \psi(x_i), \psi(\cdot)\rangle_{\mathcal{H}} \approx \sum_{i=1}^K \alpha_i z(x_i)^\top z(\cdot). \tag{4}$$

This feature representation $z(x)$ of x is termed as Random Fourier Features (RFF) and generated as

$$z(x) := \sqrt{\frac{2}{D}}\left[\cos\left(\omega_1^\top x + b_1\right), \ldots, \cos\left(\omega_D^\top x + b_D\right)\right]^\top \tag{5}$$

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 uniform distribution on $[0, 2\pi]$. Furthermore, [18] shows that this method provides an unbiased estimate of κ and the approximation error decays

Algorithm 1 KCRL

```
Input: $\tau$, $g_{\theta_0}$, $D$, $\lambda$, $\epsilon_i$, $M$, $\mu$, $\eta_1$, $\eta_2$, $\epsilon_{pd}$
for $i = 0, \ldots$ do
    for $t = i\tau, \ldots, (i+1)\tau$ do
        Execute $u_t = g_{\theta_i}(x_t)$
        Form $\phi_t = [x_t^\top, u_t^\top]^\top$ and store $z(\phi_t)$
    Solve (7) for $\hat{W}_i$ & Form $\hat{F}_i(\cdot) = \hat{W}_i^\top z(\cdot)$   ▷ Learning
    Solve (8) for $\theta_{i+1}$ using $\hat{F}_i(\cdot)$ via (9)
    Construct $g_{\theta_{i+1}}$   ▷ Stable Policy Design
```

exponentially in D (Claim 1 of [18]), which motivates the use of RFFs in function approximation in practice [21].

Notice that so far $h(\cdot)$ considered in this section is a scalar-valued function. Recently, [22] derived approximation theoretic guarantees for using RFF to approximate vector-valued nonlinear functions that live in a known RKHS within a bounded region, such as $F(\cdot)$ in (3). In particular, [22] shows that for large enough number of RFF (D), the best D-dimensional RFF approximation of F, $\bar{F}(\cdot) = W_*^\top z(\cdot)$, attains the approximation guarantee of

$$\sup_{\|\phi\| \leq \Gamma_\phi} \|\bar{F}(\phi) - F(\phi)\| \leq \tilde{O}(1/\sqrt{D}) \tag{6}$$

with high probability, where Γ describes the bounded region. Here $\tilde{O}(\cdot)$ denotes the order up to logarithmic factors of D and hides the dependencies on n and Γ_ϕ.

This result is key to our analysis as we use it to derive the finite-time learning and stabilization guarantees of KCRL in Section V. In particular, we use this result to derive a novel finite-time approximation guarantee for the Jacobian of the underlying system (3) using RFF, which is then used to guarantee the design of a finite-sample stabilizing controller via KCRL.

IV. KCRL Framework

In this section, we present KCRL. The outline of the algorithm is given 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:

1) *Model Learning:* We generate trajectories from the underlying system using the current controller, and use these trajectories to update the model estimate.
2) *Stable Policy Design:* We use a primal dual policy gradient approach to solve a stability constrained policy optimization problem and design the new controller for the next epoch.

A. Model Learning

Each epoch i of KCRL starts with a data collection from the underlying system for τ time-steps using the current controller, $g_{\theta_i}(\cdot)$. In each time step, KCRL takes the action $u_t = g_{\theta_i}(x_t)$, it computes and stores the D-dimensional RFF representation of the current state-action pair $\phi_t = [x_t^\top, u_t^\top]^\top$ as $z(\phi_t)$ using (5). Note that D and τ are user-defined parameters and the policy in the initial epoch is $g_{\theta_0}(\cdot)$.

Using the best D-dimensional RFF approximation of F defined in (6), 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 regularized least squares problem:

$$\min_W \lambda \|W\|_F^2 + \sum_{s=0}^{t=(i+1)\tau} \|x_{s+1} - W^\top z(\phi_s)\|_2^2, \tag{7}$$

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 (7) 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 $\hat{F}_i(\cdot) = \hat{W}_i^\top z(\cdot)$. Note that instead of solving (7) from scratch in every epoch, KCRL can also use online updates for constructing $\hat{W}_i$.

B. Stable Policy Design

Once KCRL has a model estimate after the data collection, it aims to recover a stabilizing policy for the underlying system. To this end, it solves the following stability-constrained policy optimization problem at epoch i using the estimated model $\hat{F}_i(\cdot)$:

$$\min_\theta J(\theta) = \sum_{t=0}^{\infty} \gamma^t c(x_t, u_t), \tag{8a}$$

$$\text{s.t. } x_{t+1} = F(\phi_t), \tag{8b}$$

$$u_t = g_\theta(x_t), \tag{8c}$$

$$\hat{G}_i(x,\theta)^\top M \hat{G}_i(x,\theta) - M \prec -\epsilon_i I, \forall x \in \mathbb{R}^n, \tag{8d}$$

where $\hat{G}_i(x,\theta) = \widehat{\frac{\partial F(\phi)}{\partial x}} + \widehat{\frac{\partial F(\phi)}{\partial u}} \frac{\partial u}{\partial x}$ for the Jacobian estimates $\widehat{\frac{\partial F(\phi)}{\partial x}}$ and $\widehat{\frac{\partial F(\phi)}{\partial u}}$ that are computed via finite difference method on the estimated model $\hat{F}_i(\cdot)$, $M \succ 0$ is an appropriately chosen positive definite matrix and $\epsilon_i > 0$ is chosen with respect to the model learning error at epoch i.

Compared to (2), the formulation (8) incorporates an additional constraint (8d). This constraint adapts Krasovskii's method for Lyapunov function construction and helps KCRL to enforce stability of the learned policy. In Section V, we provide a formal treatment of this constraint. In particular, we will first show that solving (8) using the true system $F(\cdot)$ in (8d) gives a stabilizing policy. We further show that solving (8) using a well-refined estimate $\hat{F}_i$ and an appropriate choice of ϵ_i guarantees the recovery of stabilizing policy for the underlying system $F(\cdot)$. To solve the constrained optimization problem (8), KCRL uses a primal-dual technique.

Primal-Dual Approach. To describe the approach, we use the following short-hand notation, $K(x,\theta) = \hat{G}(x,\theta)^\top M \hat{G}(x,\theta) - M + \epsilon_i I$. With this notation, (8) 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 [35],

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

Since it is not possible to evaluate the $\sup_x$, we replace the sup over the state space with a sup over a batch of representative points in the state space $\{x_i\}_{i\in\mathcal{B}}$. For the term $\nabla J(\theta)$, KCRL uses standard policy gradient estimators, e.g. DDPG [7], to evaluate the policy gradient and denote the estimated gradient as $\widehat{\nabla J(\theta)}$. Thus, the primal-dual algorithm of KCRL is given as,

$$\theta \leftarrow \theta - \eta_1 \Big[\widehat{\nabla J(\theta)} + \mu \nabla_\theta \sup_{i\in\mathcal{B}} \lambda_{\max}(K(x_i,\theta)) \Big],$$
$$\mu \leftarrow \max(0, \mu + \eta_2 \sup_{i\in\mathcal{B}} \lambda_{\max}(K(x_i,\theta) + \epsilon_{\text{pd}} I)), \tag{9}$$

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 another constant that is chosen to tolerate the possible representation incapability of $\mathcal{B}$. In Theorem 9, we give the characterization of ϵ_{pd} to provide the theoretical guarantees of KCRL.

V. Main Result: Stability Guarantee of KCRL

After presenting the algorithmic details of KCRL in the previous section, we now provide the theoretical guarantees of KCRL. First, we have the following learning and stabilizability assumptions for the underlying system $F(\cdot)$.

Assumption 1 (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 (sample covariance) matrix $Z_t Z_t^\top$ scales linearly over time. Moreover, for the closed-loop system generated by using the controller $u_t = g_{\theta_0}(x_t)$, i.e. $F(\phi_t)$ with $\phi_t = [x_t^\top, g_{\theta_0}(x)^\top]^\top$, we have $\|\phi_t\| \le \Gamma_\phi$, for some finite Γ_ϕ.*

Assumption 2 (Krasovskii's Lyapunov Function for F)**.** *Let $G(x,\theta)$ denote the true Jacobian of the closed-loop system with respect to state x, i.e., $G(x,\theta) = \frac{\partial F(\phi)}{\partial x} + \frac{\partial F(\phi)}{\partial u}\frac{\partial u}{\partial x}$. We assume that there exists an (M,θ) pair such that $G(x,\theta)^\top M G(x,\theta) - M \prec -\bar{\epsilon} I$, for some $\bar{\epsilon} > 0$.*

Assumption 3 (Regularity Conditions)**.** *(i) Both F and g_θ are continuously differentiable. (ii) F is L_F-Lipschitz, or in other words, we have Jacobian of F, $\|J_F\| \le L_F$. (iii) $\|\nabla^2 F_i\| \le \mathrm{F_H}$ for all i, where F_i denotes the mapping from ϕ_t to ith element of state vector x_{t+1} for all t, i.e., $(x_{t+1})_i = F_i(\phi_t)$ for $i = 1,\ldots,n$. (iv) Any policy in the considered policy class is always L_u-Lipschitz, that is, $\|\frac{\partial g_\theta(x)}{\partial x}\| \le L_u$, $\forall\theta$.*

The first assumption in Assumption 1 is fairly standard and it guarantees the consistent and reliable estimation of the underlying system, whereas the second assumption can be interpreted as bounded-input to bounded-state condition on the initial controller. Note that in many nonlinear control systems this condition is already satisfied due to physical laws, i.e. the state of the system cannot go unboundedly. Assumption 2 guarantees that the underlying system can be made stable with respect to Krasovskii's Lyapunov function with a stability margin, $\bar{\epsilon}$. While this assumption applies in many systems, including network congestion control [36] and power system control [29], it is an important future direction to relax this assumption and consider how to incorporate other Lyapunov function construction techniques. Finally, Assumption 3 provides standard bounds on the system properties in order to provide the analysis.

With these assumptions in hand, we are ready to present our first result that the solution of the novel stability-constrained policy optimization problem (8) with the perfect knowledge of the underlying system, particularly the true Jacobian J_F, is a stabilizing policy by design.

Theorem 1 (Stability of the True Discrete-time System)**.** *Suppose Assumptions 1-3 hold. Consider solving* (8) *with the knowledge of true model $F(\cdot)$, such that* (8d) *is evaluated using the true Jacobian J_F. If* (8d) *holds for $\epsilon_i = 0$, we have the trajectory of $x_{t+1} = f(x_t, g_\theta(x_t))$ is asymptotically stable around set $S_e = \{x : f(x, g(x)) = x\}$.*

The proof is in Appendix of the extended version online. The key idea that underpins this result is to construct Krasovskii's Lyapunov function for the system as $V(x) = (x - f(x, g_\theta(x)))^\top M (x - f(x, g_\theta(x)))$ and using Kowalewki's mean value theorem [37] to show that difference equation along the system trajectory is negative definite, which leads to our stability constraint in (8d). Note that Theorem 1 uses the exact Jacobians rather than estimates obtained via finite difference method. The following extends this result to tolerate modeling errors, in particular errors in the Jacobian estimates.

Theorem 2 (Stability under Modeling Error)**.** *Suppose Assumptions 1-3 hold and the Jacobian estimation errors 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 $i = 1,\ldots,n$. Then, when* (8d) *holds with $\epsilon_i = 2\bar{G}\|M\|(1 + L_u)\varepsilon_J + \|M\|(1 + L_u)^2 \varepsilon_J^2$, where $\bar{G} = (1 + L_u)(L_F + \varepsilon_J)$, we have the trajectory of $x_{t+1} = f(x_t, g_\theta(x_t))$ is asymptotically stable around set $S_e = \{x : f(x, g(x)) = x\}$.*

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

$$G(x)^\top M G(x) - M \prec 0, \forall x. \tag{10}$$

Let $\Delta G_i := \hat{G}_i(x) - G_i(x)$. We first bound $\|\Delta G_i\|$:

$$\|\Delta G_i\| = \left\| \widehat{\frac{\partial F_i(\phi)}{\partial x}} - \frac{\partial F_i(\phi)}{\partial x} + \Big(\widehat{\frac{\partial F_i(\phi)}{\partial u}} - \frac{\partial F_i(\phi)}{\partial u} \Big) \frac{\partial u}{\partial x} \right\|$$
$$\le (1 + L_u)\varepsilon_J. \tag{11}$$

Next, note that the stability constraint (8d) indicates

$$\hat{G}_i(x)^\top M \hat{G}_i(x) - M \prec -\epsilon_i I, \forall x.$$

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 (11) and the choice of ϵ_i. This verifies (10) and gives the advertised result. □

Next, we turn our attention to the learning guarantees of KCRL. In particular, we need to guarantee that the model estimation errors are small enough at the end of first epoch such that the controller obtained via solving (8) would stabilize the system. Recall that RFF representation translates the nonlinear system in (3) into a linear system of the form $W_*^\top z(\phi_t)$, where $z(\phi)$ is the D-dimensional RFF representation of ϕ. Thus, using standard least-squares estimation error results for the solution of (7), in particular Theorem 1 and 2 of [22], and under Assumption 1, for large enough D, we get

$$\sup_{\|\phi\| \le \Gamma_\phi} \|F(\phi) - \hat{F}_1(\phi)\| = \tilde{O}(1/\sqrt{D} + \sqrt{D/\tau}), \tag{12}$$

after τ time-steps, *i.e.*, at the end of first epoch of KCRL.

Using (12), we derive the following novel finite sample approximation error guarantee on the Jacobian of the underlying function $F(\cdot)$ via the finite difference method for the computation of partial derivatives.

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 $\hat{F}_1(\cdot) = \hat{W}_1^\top z(\cdot)$, i.e., the estimated system dynamics at the end of the first epoch, such that*

$$\hat{J}_F^{(i,j)}(\phi) = \frac{\hat{F}_{1,i}(\phi + \varepsilon\ \mathrm{e_j}) - \hat{F}_{1,i}(\phi - \varepsilon\ \mathrm{e_j})}{2\varepsilon}, \tag{13}$$

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 & 3, for the optimal choice of $\varepsilon = \tilde{O}\left((1/\sqrt{D} + \sqrt{D/\tau})^{1/3}\right)$, we have that $\sup_{\|\phi\| \le B} \|\hat{J}_F(\phi) - J_F(\phi)\|_F = \tilde{O}\left(\varepsilon^2\right)$.

Proof. Consider the following

$$F_i(\phi + \varepsilon\ \mathrm{e_j}) = F_i(\phi) + \varepsilon \frac{\partial F_i}{\partial \phi_j} + \varepsilon^2 \frac{\partial^2 F_i}{\partial \phi_j^2} + O(\varepsilon^3)$$

$$F_i(\phi - \varepsilon\ \mathrm{e_j}) = F_i(x) - \varepsilon \frac{\partial F_i}{\partial \phi_j} + \varepsilon^2 \frac{\partial^2 F_i}{\partial \phi_j^2} - O(\varepsilon^3).$$

From Assumption 3, we have

$$\frac{F_i(\phi + \varepsilon\ \mathrm{e_j}) - F_i(\phi - \varepsilon\ \mathrm{e_j})}{2\varepsilon} = \frac{\partial F_i}{\partial \phi_j} + \mathrm{F_H} O(\varepsilon^2). \tag{14}$$

Now consider (13). Let $\delta_\tau = \tilde{O}(1/\sqrt{D} + \sqrt{D/\tau})$. From (12), we have $\hat{F}_{1,i}(\phi + \varepsilon\ \mathrm{e_j}) = F_i(\phi + \varepsilon\ \mathrm{e_j}) + \epsilon_1$ and $\hat{F}_{1,i}(\phi - \varepsilon\ \mathrm{e_j}) = F_i(\phi - \varepsilon\ \mathrm{e_j}) + \epsilon_2$ for $0 \le \epsilon_1, \epsilon_2 \le \delta_\tau$. Combining this with (14), we obtain

$$\begin{aligned} \hat{J}_F^{(i,j)}(\phi) - \frac{\partial F_i}{\partial \phi_j} &= \frac{\hat{F}_{1,i}(\phi + \varepsilon\ \mathrm{e_j}) - \hat{F}_{1,i}(\phi - \varepsilon\ \mathrm{e_j})}{2\varepsilon} - \frac{\partial F_i}{\partial \phi_j} \\ &= \frac{F_i(\phi + \varepsilon\ \mathrm{e_j}) + \epsilon_1 - F_i(\phi - \varepsilon\ \mathrm{e_j}) - \epsilon_2}{2\varepsilon} - \frac{\partial F_i}{\partial \phi_j} \\ &= \mathrm{F_H} O(\varepsilon^2) + \frac{\epsilon_1 - \epsilon_2}{2\varepsilon}. \end{aligned}$$

This gives us that $|\hat{J}_F^{(i,j)}(\phi) - \frac{\partial F_i}{\partial \phi_j}| \le \mathrm{F_H} O(\varepsilon^2) + \delta_\tau/\varepsilon$ for all i, j. Combining these yields

$$\|\hat{J}_F(\phi) - J_F(\phi)\|_F \le n\left(c\mathrm{F_H}\varepsilon^2 + \frac{\delta_\tau}{\varepsilon}\right),$$

for some constant c. Note that the optimal choice of ε is $\varepsilon = \tilde{O}\left((1/\sqrt{D} + \sqrt{D/\tau})^{1/3}\right)$, which gives

$$\|\hat{J}_F(\phi) - J_F(\phi)\|_F = \tilde{O}\left((1/\sqrt{D} + \sqrt{D/\tau})^{2/3}\right).$$

□

This result shows that the Jacobian of a vector-valued function in a known RKHS is well-approximated using RFF representation of the function with finite samples, which can be of independent interest. Before stating our sample complexity result for stabilizing the system dynamics via KCRL, we first show that the convergence of primal-dual method in (9) guarantees the satisfaction of stability condition (8d) for all points with proper choice of ϵ_{pd} and ϵ_i.

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* (9)*. Further, let $\mathcal{X}$ denote the bounded compact state space of the system for $\|\phi\| \le \Gamma_\phi$ and 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* KCRL *sets $\epsilon_{\mathrm{pd}} = 2\bar{G}\|M\|M_G h$ in Eq* (9)*, then the stability condition is met on the entire state space $\mathcal{X}$ by choosing $\epsilon_i = 2\bar{G}\|M\|(1 + L_u)\varepsilon_J + \|M\|(1 + L_u)^2 \varepsilon_J^2$ in the constraint* (8d).

Remark 2. *Recall from Section IV that the batch $\mathcal{B}$ constitutes arbitrary points in state-space to estimate the supremum of the stability constraint and does not correspond to data collected from the system. In particular,* KCRL *only requires the evaluation in* (9) *using the estimated system dynamics in these particularly chosen representative points in the state-space $\mathcal{X}$. Note that h corresponds to the fill distance for the batch $\mathcal{B}$. This condition can be met by using $N = (\Gamma_\phi/h + 1)^d$ samples in the batch $\mathcal{B}$. Furthermore, in the expense of computational burden, N can be even picked larger which would in return reduce h and consequently shrink ϵ_{pd} arbitrarily.*

Proof. Recall the definition of $K(x,\theta)=\hat{G}(x,\theta)^\top M\hat{G}(x,\theta)-M+\epsilon_i I$. The dual variable update in (9) 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 $\forall i\in\mathcal{B},\lambda_{\max}(K(x_i,\theta^*)))\leq 0$, 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 $\lambda_{\max}(K(x_i,\theta^*)))\leq 0$ 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 throughout the proof)

$$\begin{aligned}&\min_{x_i\in\mathcal{B}}||K(x)-K(x_i)||=\min_{x_i\in\mathcal{B}}||\hat{G}(x)^\top M\hat{G}(x)-\hat{G}(x_i)^\top M\hat{G}(x_i)||\\&\leq\min_{x_i\in\mathcal{B}_h}||(\hat{G}(x)-\hat{G}(x_i))^\top M\hat{G}(x)||+||\hat{G}(x_i)^\top M(\hat{G}(x)-\hat{G}(x_i))||\\&\leq 2\bar{G}||M||M_G h \qquad (15)\end{aligned}$$

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 $K(x_i,\theta^*)\prec-\epsilon_i I$ holds for all x in the entire state space $\mathcal{X}$. By Theorem 2, stability holds for the true system, i.e., $G(x)^\top MG(x)-M\prec 0$ for all $x\in\mathcal{X}$. □

After stating these key guarantees, we finally provide the finite-sample stabilization guarantee of KCRL.

Theorem 4 (Finite Sample Stabilization via KCRL)**.** *Suppose Assumptions 1-3 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* (8d). *If* KCRL *uses $D=\tilde{O}\left(\left(\frac{2\bar{G}\|M\|(1+L_u)+\|M\|(1+L_u)^2}{\bar{\epsilon}-\epsilon_{pd}}\right)^3\right)$ 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 set $S_e=\{x:f(x,g(x))=x\}$, i.e., the solution of* (8) *after D^2 samples from the system gives a stabilizing controller g_θ for the unknown nonlinear dynamical system.*

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

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

since $\varepsilon_J<1$. Note that this gives an upper bound on the error of Jacobian estimates to guarantee stabilization. From Proposition 1, we also have that $\varepsilon_J=\tilde{O}\left(\left(1/\sqrt{D}+\sqrt{D/\tau}\right)^{2/3}\right)$, 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 $\hat{J}_F(\phi)-J_F(\phi)$. Furthermore, the optimal choice of τ and D that minimizes this upped bound is $\tau=D^2$, which results that $\varepsilon_J=\tilde{O}\left(D^{-1/3}\right)$ after $\tau=D^2$ samples. Therefore, for the choice of

$$D=\tilde{O}\left(\left(\frac{2\bar{G}\|M\|(1+L_u)+\|M\|(1+L_u)^2}{\bar{\epsilon}-\epsilon_{pd}}\right)^3\right)$$

number of RFF after $\tau=D^2$ time-steps KCRL is guaranteed to stabilize the underlying system dynamics. □

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}. Furthermore, the choice of ϵ_i in Theorem 4 guarantees the recovery of stabilizing controllers for the subsequent epochs with the non-increasing behavior of the estimation errors.

VI. Conclusions and Future Work

In this paper, we study the online control of unknown nonlinear dynamical systems. We propose a model-based RL framework, Krasovskii Constrainted RL (KCRL), that provides the first sample complexity result for stabilization of unknown nonlinear systems. KCRL iteratively learns the unknown model dynamics via RFF representation, and solves the policy optimization using the primal-dual approach, with a stability constraint generated by Krasovskii's construction of Lyapunov function. There are many interesting open questions that remain. For example, in practice the Lyapunov function design is a challenging task which requires heuristics and Krasovskii's method is only one way to construct Lyapunov functions. It is important to broaden the framework and consider how to incorporate other Lyapunov function construction techniques. In particular, Krasovskii's method is related to contraction analysis where the matrix M is allowed to be time-dependent. This can be viewed as a generalization of the Krasovskii's method and has been widely used in robotic manipulation and locomotion tasks.

References

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, p. 484, 2016.

[2] 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.

[3] 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.

[4] 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.

[5] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, “Explore more and improve regret in linear quadratic regulators,” *arXiv preprint*, 2020.

[6] 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.

[7] 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.

[8] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 1861–1870.

[9] W. Cui and B. Zhang, "Reinforcement learning for optimal frequency control: A lyapunov approach," *arXiv preprint arXiv:2009.05654*, 2020.

[10] K. Zhou, J. C. Doyle, K. Glover *et al.*, *Robust and optimal control*, vol. 40.

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

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

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

[14] D. Feijer and F. Paganini, "Krasovskii's method in the stability of network control," in *2009 American Control Conference*. IEEE, 2009, pp. 3292–3297.

[15] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, "Neural stochastic contraction metrics for learning-based control and estimation," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1825–1830, 2020.

[16] S. Singh, S. M. Richards, V. Sindhwani, J.-J. E. Slotine, and M. Pavone, "Learning stabilizable nonlinear dynamics with contraction-based regularization," *The International Journal of Robotics Research*, 2020.

[17] N. M. Boffi, S. Tu, N. Matni, J.-J. E. Slotine, and V. Sindhwani, "Learning stability certificates from data," *arXiv preprint*, 2020.

[18] 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.

[19] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 253–279, 2019.

[20] M. Simchowitz and D. Foster, "Naive exploration is optimal for online lqr," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8937–8948.

[21] S. Kakade, A. Krishnamurthy, K. Lowrey, M. Ohnishi, and W. Sun, "Information theoretic regret bounds for online nonlinear control," *arXiv preprint*, 2020.

[22] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, "Model learning predictive control in nonlinear dynamical systems," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 757–762.

[23] 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.

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

[25] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A lyapunov-based approach to safe reinforcement learning," *Advances in Neural Information Processing Systems*, 2018.

[26] S. M. Richards, F. Berkenkamp, and A. Krause, "The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems," in *Conference on Robot Learning*. PMLR, 2018, pp. 466–476.

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

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

[29] Y. Shi, G. Qu, S. Low, A. Anandkumar, and A. Wierman, "Stability constrained reinforcement learning for real-time voltage control," *arXiv preprint*, 2021.

[30] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Learning for Dynamics and Control*. PMLR, 2020.

[31] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin, "Bridging hamilton-jacobi safety analysis and reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8550–8556.

[32] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

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

[34] H. Mania, M. I. Jordan, and B. Recht, "Active learning for nonlinear system identification with guarantees," *arXiv preprint arXiv:2006.10277*, 2020.

[35] 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.

[36] D. Feijer and F. Paganini, "Stability of primal–dual gradient dynamics and applications to network optimization," *Automatica*, vol. 46, no. 12, pp. 1974–1981, 2010.

[37] G. Kowalewski, "Ein mittelwertsatz für ein system von n integralen," *Z. Math. Phys.(Schlömilch Z.)*, vol. 42, pp. 153–157, 1895.

APPENDIX

In the Appendix, we show that the stability constraint of KCRL, (8d), guarantees the stability of the underlying system using the true system dynamics, Theorem 1.

Proposition 2 (Kowalewski's Mean Value Theorem [37])**.** *Let $x_1, \ldots, x_n$ be continuous functions in a variable $t \in [a,b]$. There exists real numbers $t_1, \ldots, t_n$ in $[a,b]$ and non-negative $\lambda_1, \ldots, \lambda_n$, with $\sum_i^n \lambda_i = b - a$, such that*

$$\int_a^b x_k(t)dt = \sum_{i=1}^n \lambda_i x_k(t_i)$$

for each $k = 1, \ldots, n$.

A. Proof of Theorem 1

Proof. The key technique that underpins the proposed framework is to use Krasovskii's method for Lyapunov function construction to certifty stability. Note that Krasovskii's method considers continuous time dynamical systems, *i.e.*, $\frac{d}{dt}x(t) = h(x(t))$. For this system, Krasovskii's method considers the candidate Lyapunov function of $V_h(x) = h^\top(x)Ph(x)$, for some positive definite matrix $P \succ 0$. We first convert this approach to discrete-time nonlinear dynamical systems. To this end, notice that $h(x) = x - f(x)$ mimics the continuous case. In particular, let x_e be the fixed point for $f(x)$, i.e., $f(x_e) = x_e$, which makes $h(x_e) = 0$. Following the Krasovskii's method, this gives the candidate Lyapunov function of

$$V_f(x) = (x - f(x))^\top M(x - f(x)) \tag{17}$$

for the underlying system $f(x)$. Note that we are considering the non-autonomous systems $F(\phi)$, where $\phi = [x^\top, u^\top]^\top$ given the controller $u = g_\theta(x)$. Let $F_\theta(x)$ denote the closed-loop system dynamics obtained via controller $g_\theta(x)$, i.e. $F_\theta(x) = F(\phi)$ with $\phi = [x^\top, g_\theta(x)^\top]^\top$. Therefore, we consider the following Lyapunov function:

$$V(x) = (x - F_\theta(x))^\top M(x - F_\theta(x)). \tag{18}$$

Firstly, note that $V(x) \geq 0$, and $V(x) = 0$ if and only if $x \in S_e$ as M is positive definite. Then, we consider

$$V(x_{t+1}) - V(x_t) = (x_{t+1} - F_\theta(x_{t+1}))^\top M(x_{t+1} - F_\theta(x_{t+1})) - (x_t - F_\theta(x_t))^\top M(x_t - F_\theta(x_t)). \tag{19}$$

First, consider the following for $h(x) = x - F_\theta(x)$. One can write $h(x_{t+1})$ in terms of $h(x_t)$ as follows,

$$h(x_{t+1}) = h(x_t) + \int_0^1 \frac{\partial h}{\partial x}(x_t + t(x_{t+1} - x_t))(x_{t+1} - x_t)dt.$$

From Proposition 2, we have

$$h(x_{t+1}) = h(x_t) + J_h(x_{t+1} - x_t),$$

where $J_h = \sum_{i=1}^n \lambda_i \frac{\partial h}{\partial x}(x_t + k_i(x_{t+1} - x_t))$ for $k_i \in [0,1]$, $\lambda_i \geq 0$ for all i and $\sum_i^n \lambda_i = 1$. Plugging this in $V(x_{t+1})$, we get

$$V(x_{t+1}) = h(x_t)^\top Mh(x_t) + 2(x_{t+1} - x_t)^\top J_h^\top Mh(x_t) + (x_{t+1} - x_t)^\top J_h^\top MJ_h(x_{t+1} - x_t)$$

Note that $x_{t+1} - x_t = F_\theta(x_t) - x_t = -h(x_t)$. Therefore, we get

$$\begin{aligned} V(x_{t+1}) &= h(x_t)^\top Mh(x_t) - 2h(x_t)^\top J_h^\top Mh(x_t) + h(x_t)^\top J_h^\top MJ_h h(x_t) \\ &= h(x_t)^\top (I - J_h)^\top M(I - J_h)h(x_t) \end{aligned}$$

From the definition of J_h and $h(x) = x - F_\theta(x)$, we have $J_h = I - J_{F_\theta}$, where $J_{F_\theta} = \sum_{i=1}^n \lambda_i \frac{\partial F_\theta}{\partial x}(x_t + k_i(x_{t+1} - x_t))$, where k_i and λ_i follow from the definition of J_h. Thus we get

$$V(x_{t+1}) = (x_t - F_\theta(x_t))^\top J_{F_\theta}^\top MJ_{F_\theta}(x_t - F_\theta(x_t)). \tag{20}$$

Plugging this in (19) gives

$$V(x_{t+1}) - V(x_t) = (x_t - F_\theta(x_t))^\top (J_{F_\theta}^\top MJ_{F_\theta} - M)(x_t - F_\theta(x_t)). \tag{21}$$

For any $x \in \mathbb{R}^n$, we have

$$x^\top J_{F_\theta}^\top MJ_{F_\theta}x = \left\| M^{1/2}J_{F_\theta}x \right\|^2$$

$$
\begin{aligned}
&= \left\| \sum_{i=1}^{n} \lambda_i M^{1/2} \frac{\partial F_\theta}{\partial x} \Big(x_t + k_i (x_{t+1} - x_t) \Big) x \right\|^2 \\
&\leq \sum_{i=1}^{n} \lambda_i \left\| M^{1/2} \frac{\partial F_\theta}{\partial x} \Big(x_t + k_i (x_{t+1} - x_t) \Big) x \right\|^2 \\
&= \sum_{i=1}^{n} \lambda_i x^\top \frac{\partial F_\theta}{\partial x} \Big(x_t + k_i (x_{t+1} - x_t) \Big)^\top M \frac{\partial F_\theta}{\partial x} \Big(x_t + k_i (x_{t+1} - x_t) \Big) x,
\end{aligned}
$$

where the inequality is due to Jensen's inequality. Due to the constraint (8d), we have that $\frac{\partial F_\theta}{\partial x}(x)^\top M \frac{\partial F_\theta}{\partial x}(x) \prec M - \epsilon I$ for all $x \in \mathbb{R}^n$. Thus, we have that

$$
J_{F_\theta}^\top M J_{F_\theta} - M \leq \sum_i \lambda_i \left[\frac{\partial F_\theta}{\partial x} \Big(x_t + k_i (x_{t+1} - x_t) \Big)^\top M \frac{\partial F_\theta}{\partial x} \Big(x_t + k_i (x_{t+1} - x_t) \Big) - M \right] \leq -\epsilon I. \tag{22}
$$

This shows that the Lyapunov function is decreasing along the system trajectory, i.e. $V(x_{t+1}) - V(x_t) = (x_t - F_\theta(x_t))^\top (J_{F_\theta}^\top M J_{F_\theta} - M)(x_t - F_\theta(x_t)) < 0$.

Lastly, if a trajectory x_t is such that $V(x_{t+1}) - V(x_t) = 0, \forall t \geq 0$, then we must have $F_\theta(x_t) = x_t$ for all t, i.e. $x_t \in S_e$ for all t. Therefore, by LaSalle's Invariance Principle, we must have S_e is asymptotically stable.

□