MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Robust Model Predictive Control with Data-Driven Koopman Operators

Mamakoukas, Giorgos; Di Cairano, Stefano; Vinod, Abraham P.

TR2022-054 June 02, 2022

Abstract

This paper presents robust Koopman model predictive control (RK-MPC), a framework that leverages the training errors of data-driven models to improve constraint satisfaction. Koopman-based control has enabled fast nonlinear feedback using linear tools, but existing approaches ignore the modeling error during control, which can lead to constraint violations. Our approach assumes that the unknown dynamics are Lipschitz-continuous and uses the training error of datadriven Koopman models to approximate a Lipschitz constant for the state- and control-dependent model error. We then use the Lipschitz constant to bound the prediction error along the planning horizon and formulate a convex, robust finite-horizon optimal control problem that is real-time implementable. We demonstrate the efficacy of this approach with simulation results using the dynamics of a forced Duffing oscillator and a quadrotor. Our Python implementation can run in real-time at 66Hz for the 17-dimensional duffing oscillator and at 12Hz for the 44-dimensional quadrotor on a standard laptop.

American Control Conference (ACC) 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Robust Model Predictive Control with Data-Driven Koopman Operators

Giorgos Mamakoukas, Stefano Di Cairano, and Abraham P. Vinod

Abstract— **This paper presents robust Koopman model predictive control (RK-MPC), a framework that leverages the training errors of data-driven models to improve constraint satisfaction. Koopman-based control has enabled fast nonlinear feedback using linear tools, but existing approaches ignore the modeling error during control, which can lead to constraint violations. Our approach assumes that the unknown dynamics are Lipschitz-continuous and uses the training error of data-driven Koopman models to approximate a Lipschitz constant for the state- and control-dependent model error. We then use the Lipschitz constant to bound the prediction error along the planning horizon and formulate a convex, robust finite-horizon optimal control problem that is real-time implementable. We demonstrate the efficacy of this approach with simulation results using the dynamics of a forced Duffing oscillator and a quadrotor. Our Python implementation can run in real-time at 66Hz for the 17-dimensional duffing oscillator and at 12Hz for the 44-dimensional quadrotor on a standard laptop.**

I. Introduction

Robotics often involve applications where a model of the system dynamics is not available (e.g., soft robotics [1], [2], human-robot interaction [3]), is tedious to develop, or complicated enough to present challenges for high-fidelity real-time implementation [4]. Further, the dynamics of robots or their environments can change, rendering models inaccurate [5]–[7]. These challenges have motivated researchers to seek data-driven methods to identify, update, and simplify models [8]–[11] or directly learn successful control policies [12], [13]. However, models and policies learned from data involve uncertainty, which creates a challenge in providing safety and performance guarantees.

Safe data-driven control, in particular, has recently attracted a lot of research interest, with several efforts focusing on reinforcement learning [14]–[19] and synthesis of constrained learning-based control techniques [20]–[25]. However, these methods often require a significant amount of data. Another line of research uses robust control methods [26] to provide solutions that guarantee the satisfaction of state constraints against the worst-case effect of uncertainty. These methods typically propagate the model uncertainty into the future and generate a set of possible deviations from the nominal prediction. Then, they use the worst-case deviation to tighten the state constraints accordingly to ensure safe control [27]–[34].

Despite the promise of these efforts, existing algorithms face limitations. Some make restricting assumptions regarding the model uncertainty, e.g., that the error is independent of control, state, or both, or that it is upper bounded by a value that is known. Other methods apply only to linear systems, or are computationally expensive and cannot be implemented in real-time. Moreover, most of these algorithms require their training set to cover a sufficiently large part of the state space. To address these challenges, we propose a data-driven, real-time implementable approach based on Koopman operators and robust control for *a priori* unknown, possibly nonlinear dynamics, while explicitly accounting for state- and control-dependent model uncertainty.

Koopman operators evolve linearly higher-dimensional, lifted functions of the system states without loss of accuracy everywhere in the state-space. The ability to use linear tools for nonlinear prediction and control has attracted recent research efforts in robotics and control [35]. Because Koopman operators are typically infinite-dimensional, researchers focus on either i) finding finite-dimensional Koopman eigenspaces from data [36]–[42] or ii) obtaining approximate Koopman models that offer a balance between high accuracy and low dimensionality [43], [44]. The linearity of Koopman representations is desirable due to the computational efficiency, the convexity of dynamical constraints. and, when a finite-dimensional Koopman model exists without model error, the improved control performance [36]. Koopman models have been successfully used with model predictive control [45] and LQR [44] and have shown great promise in many applications, such as in soft [2], [46] and underwater [44] robotics. For a more comprehensive review of Koopman operator theory, we refer the reader to [47].

We present a novel approach for constrained data-driven control of unknown systems by combining Koopman representations with robust model predictive control. The proposed method, which requires only that the unknown dynamics are Lipschitz continuous, synthesizes controllers that are cognizant of the model error and yields convex optimization problems that can be solved via off-the-shelf solvers. The proposed approach, which we call *robust Koopman model predictive control (RK-MPC)*, consists of two key contributions: 1) characterization of the state- and input-dependent modeling error for data-driven Koopman models using Lipschitz analysis, and 2) tightened state-constraints along the prediction steps in the presence of state- and input-dependent uncertainty for Koopman operators. The proposed approach utilizes library-free Koopman models that are synthesized using only past state and control measurements, without the need to search for basis functions.

II. Preliminaries & problem statements

We use the following notation throughout the paper: The interval $\mathbb{N}_{[a,b]}$ enumerates all natural numbers between and

All authors are with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA.

including $a, b \in \mathbb{N}$; $\mathcal{S}^n$ is the Cartesian product of the set $\mathcal{S}$ with itself n-times for any $n \in \mathbb{N}$; I_n and $0_{n\times m}$ denote the $n \times n$ identity and $n \times m$ zero matrices, respectively; $\|x\|$ is the 2-norm of $x \in \mathbb{R}^n$; $\|A\|_{\mathrm{op}} = \sup_{x\in\mathbb{R}^n}(\|Ax\|/\|x\|)$ is the operator-norm of $A \in \mathbb{R}^{m\times n}$; $x\cdot y$ is the inner product of $x, y \in \mathbb{R}^n$; $x^\top$ is the transpose of $x \in \mathbb{R}^n$; and $\mathrm{Ball}(c, r) = \{x \in \mathbb{R}^n : \|x - c\| \leq r\}$ is a n-dimensional ball of radius $r > 0$ centered at $c \in \mathbb{R}^n$. Last, we refer to a function $f : \mathbb{R}^n \to \mathbb{R}$ as Lipschitz continuous, when $\|f(x) - f(y)\| \leq L_f \|x - y\|$ for any $x, y \in \mathbb{R}^n$ for some constant $L_f > 0$. We denote such a function to be L_f-Lipschitz, and refer to L_f as the Lipschitz constant of f.

A. Hankel Koopman theory for nonlinear systems

In this subsection, we review Hankel Koopman models for nonlinear systems [48]. Throughout this paper, we assume that the true dynamics of a system are unknown and nonlinear and expressed as

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

with state $x_t \in \mathcal{X} \subset \mathbb{R}^n$, input $u_t \in \mathcal{U} \subset \mathbb{R}^m$, and dynamics $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$. Hankel Koopman theory uses delay measurements to form the Koopman basis functions. Let $d_x, d_u \in \mathbb{N}$ denote the number of delay state- and control-dependent measurements used in the Koopman model, respectively. Then, Hankel Koopman theory uses an N-dimensional lifted state Ψ_t,

$$\Psi_t = \begin{bmatrix} x_t^\top & h_t^\top \end{bmatrix}^\top = \begin{bmatrix} x_t^\top & x_{t-1}^\top & \cdots & x_{t-d_x}^\top & u_{t-1}^\top & \cdots & u_{t-d_u}^\top \end{bmatrix}^\top,$$

where $N = n + nd_x + md_u$ and $h_t \in \mathbb{R}^{N-n}$ is the corresponding history of states and inputs at time t. We further define h_t as a $(N - n)$-dimensional zero vector.

We express the dynamics in the lifted state space using a tuple (A, B, C), where

$$\begin{aligned} \Psi_{t+1} &= A\Psi_t + Bu_t + \epsilon(\Psi_t, u_t) \qquad &\text{(2a)}\\ &= \begin{bmatrix} A_{\text{data}} \\ A_{\text{fixed}} \end{bmatrix} \begin{bmatrix} x_t \\ h_t \end{bmatrix} + \begin{bmatrix} B_{\text{data}} \\ B_{\text{fixed}} \end{bmatrix} u_t + \begin{bmatrix} \epsilon_x(\Psi_t, u_t) \\ 0 \end{bmatrix} & \\ x_t &\triangleq C\Psi_t, &\text{(2b)} \end{aligned}$$

where $A_{\text{data}} \in \mathbb{R}^{n\times N}$ and $B_{\text{data}} \in \mathbb{R}^{n\times m}$ are obtained from data, $A_{\text{fixed}} \in \mathbb{R}^{(N-n)\times N}$ and $B_{\text{fixed}} \in \mathbb{R}^{(N-n)\times m}$ are defined by the appropriate time shifts in the history h_t, and $C \in \mathbb{R}^{n\times N}$ is the projection matrix that projects Ψ_t back to x_t. In practice, we compute $(A_{\text{data}}, B_{\text{data}})$ using a closed-form least-squares solution, as in [44], and set $(A_{\text{fixed}}, B_{\text{fixed}})$

$$A_{\text{fixed}} = \begin{bmatrix} I_{nd_x} & 0_{(nd_x)\times(md_u)} & 0_{n\times n} \\ 0_{m\times(nd_u)} & 0_{m\times(md_u)} & 0_{n\times n} \\ 0_{(m(d_u-1))\times(nd_u+1)} & I_{m(d_u-1)} & 0_{(m(d_u-1))\times m} \end{bmatrix},$$

$$B_{\text{fixed}} = \begin{bmatrix} 0_{(nd_x)\times m} \\ I_m \\ 0_{((m-1)d_u)\times m} \end{bmatrix}.$$

Given our choice of $(A_{\text{fixed}}, B_{\text{fixed}})$, the modeling error $\epsilon : \mathbb{R}^N \times \mathcal{U} \to \mathbb{R}^N$ in the Koopman model is explicitly defined by the modeling error in the low-dimensional state ϵ_x, where

$$\epsilon_x(\Psi_t, u_t) \triangleq f(C\Psi_t, u_t) - A_{\text{data}}\Psi_t - B_{\text{data}}u_t. \tag{3}$$

Using (3), the nonlinear model (1) and the Koopman model (2) provide an identical description of the dynamics.

B. Problem setup

Given the current state x_0, we consider the following optimization problem to be solved at every MPC iteration,

$$\begin{aligned} \underset{u_0,\ldots,u_{T-1}}{\min.} \quad & J(x_1, \ldots, x_T, u_0, \ldots, u_{T-1}) & \text{(4a)}\\ \text{s. t.} \quad & \begin{cases} \text{Terminal constraints on } x_T \\ \text{for recursive feasibility,} \end{cases} & \text{(4b)}\\ t \in \mathbb{N}_{[0,T-1]}, \quad & x_{t+1} = f(x_t, u_t), & \text{(4c)}\\ t \in \mathbb{N}_{[0,T-1]}, \quad & u_t \in \mathcal{U}, & \text{(4d)}\\ t \in \mathbb{N}_{[1,T]}, \quad & x_t \in \mathrm{SafeSet}, & \text{(4e)} \end{aligned}$$

where $\mathrm{SafeSet} = \cap_{i\in[N_S]}\{p_i \cdot x \leq q\} \subset \mathcal{X}$ defines a polytope formed from $N_S \in \mathbb{N}$ hyperplanes with $p_i \in \mathbb{R}^n$ and $q_i \in \mathbb{R}$. In this work, we choose a quadratic cost $J : \mathcal{X}^T \times \mathcal{U}^T \to \mathbb{R}$

$$J = \sum_{t=1}^{T} \left(x_t - x_t^{\text{target}}\right) \cdot \left(Q\left(x_t - x_t^{\text{target}}\right)\right) + \sum_{t=0}^{T-1} u_t \cdot (Ru_t),$$

where $x_t^{\text{target}} \in \mathcal{X}$ is a user-specified target state.

Consider N_D training measurements of the lifted state Ψ_i^{tr}, the input u_i^{tr}, and the next lifted state $(\Psi_i^{\text{tr}})^+$. Then, using (3) and (2b), the modeling error at the training points is

$$\epsilon_x(\Psi_i^{\text{tr}}, u_i^{\text{tr}}) = C(\Psi_i^{\text{tr}})^+ - A_{\text{data}}\Psi_i^{\text{tr}} - B_{\text{data}}u_i^{\text{tr}}. \tag{5}$$

Using Hankel Koopman theory, we express (4) with decision variables $u_0, u_1, \ldots, u_{T-1}$ and current lifted state Ψ_0:

$$\begin{aligned} \text{minimize } & J(C\Psi_1, \ldots, C\Psi_T, u_0, \ldots, U_{T-1}) & \text{(6a)}\\ \text{subject to } & \begin{cases} \text{Terminal constraints on } \Psi_T \\ \text{for recursive feasibility,} \end{cases} & \text{(6b)}\\ t \in \mathbb{N}_{[0,T-1]},\ & \Psi_{t+1} = A\Psi_t + Bu_t + \epsilon(\Psi_t, u_t), & \text{(6c)}\\ t \in \mathbb{N}_{[0,T-1]},\ & u_t \in \mathcal{U}, & \text{(6d)}\\ t \in \mathbb{N}_{[1,T]},\ & C\Psi_t \in \mathrm{SafeSet}. & \text{(6e)} \end{aligned}$$

We emphasize that (6) is an exact reformulation of (4) and does not introduce any approximation.

Existing approaches in Koopman MPC [44]–[46] typically ignore the modeling error ϵ and the issue of recursive feasibility and solve the following convex optimization problem:

$$\begin{aligned} \text{minimize} \quad & J(C\Psi_1, \ldots, C\Psi_T, u_0, \ldots, U_{T-1}) & \text{(7a)}\\ \text{subject to} \quad & & \\ t \in \mathbb{N}_{[0,T-1]}, \quad & \Psi_{t+1} = A\Psi_t + Bu_t, \text{ (6d), (6e).} & \text{(7b)} \end{aligned}$$

However, (7) can be infeasible due to the modeling error and the lack of recursive feasibility constraints, especially when the modeling error is high. This motivates Problem 1.

Problem 1. *Given a general nonlinear system* (1) *and a Koopman model* (2) *trained on data* $\{\Psi_i^{\text{tr}}, u_i^{\text{tr}}, \epsilon(\Psi_i^{\text{tr}}, u_i^{\text{tr}})\}_{i=1}^{N_D}$, *characterize the* a priori *unknown modeling error* ϵ *using Lipschitz analysis, model its effect on the state constraints, and propose a convex conservative approximation of* (6).

III. Robust Koopman Model Predictive Control

In this section, we present the proposed convex, robust Koopman model predictive control (RK-MPC) framework. RK-MPC consists of an offline and an online phase. Offline, we learn the linear dynamics in the lifted space $(A_{\text{data}}, B_{\text{data}})$ from training data, and also characterize the Lipschitz constant of the modeling error ϵ_x. Online, we formulate a finite-horizon convex, robust optimal control problem that makes use of the data-driven error Lipschitz constant to approximate state- and control- dependent bounds on the modeling error.

We organize this section as follows. First, we propose a Lipschitz-based characterization of the modeling error. Next, we formulate a robust optimal control problem under Lipschitz continuous state- and input-dependent uncertainty. We conclude with a discussion of the strengths and weaknesses of the proposed approach.

A. Lipschitz-based characterization of the modeling error

To extrapolate the modeling error of a data-driven Koopman representation beyond the training set, we make the following assumption.

Assumption 1 (Lipschitz modeling error). *The modeling error ϵ_x is L_ϵ-Lipschitz for some $L_\epsilon > 0$.*

A consequence of Assumption 1 is that the modeling error ϵ is also Lipschitz continuous with a Lipschitz constant L_ϵ. Further, the evaluation of the *a priori* unknown modeling error ϵ_x at any *query* lifted state-input pair (Ψ, u) satisfies the following characterization based on the training data:

$$\epsilon_x(\Psi,u) \in \bigcap_{i=0}^{N_D} \mathrm{Ball}\left(\epsilon_x(\Psi_i^{\mathrm{tr}}, u_i^{\mathrm{tr}}), L_\epsilon \left\| \begin{bmatrix} \Psi - \Psi_i^{\mathrm{tr}} \\ u - u_i^{\mathrm{tr}} \end{bmatrix} \right\| \right). \tag{8}$$

In other words, the uncertainty in the modeling error ϵ_x grows as the query point (Ψ, u) grows further away from the training points.

We note that Assumption 1 holds for a system with Lipschitz-continuous dynamics, which include any continuously differentiable dynamics defined over a compact set $\mathcal{X}$. Examples of such dynamics include Dubin's car, quadrotors, and the forced Duffing oscillator. In the remainder of this section, we show that Assumption 1 is not restrictive and show how we estimate L_ϵ in practice.

Proposition 1 (Lipschitz dynamics imply Lipschitz modelling error). *Let f be L_f-Lipschitz over a set $\mathcal{X}$. Then, the modeling error ϵ_x is Lipschitz-continuous, with a Lipschitz constant $L_\epsilon = L_f + \|C[A\ B]\|$.*

Proof. See Appendix A. □

Proposition 1 provides a Lipschitz constant for ϵ_x in terms of the Lipschitz constant L_f of the underlying nonlinear dynamics (1) and Koopman model (A, B, C). However, L_f may not be available in practice. Alternatively, we compute L_ϵ using a finite set of data points $\mathcal{Z} = \{(\Psi_i, u_i)\}_{i=1}^{N_Z}$ to obtain a lower bound L_ϵ^- of the true Lipschitz constant L_ϵ,

$$L_\epsilon^-(\mathcal{Z}) \triangleq \max_{\substack{i,j \in \mathbb{N}_{[1,N_Z]} \\ i \neq j}} \frac{\|\epsilon_x(\Psi_i, u_i) - \epsilon_x(\Psi_j, u_j)\|}{\left\| \begin{bmatrix} \Psi_i - \Psi_j \\ u_i - u_j \end{bmatrix} \right\|}. \tag{9}$$

Recently, authors in [25] proposed a more data-intensive algorithm to obtain a probabilistic upper bound of the Lipschitz constant. There, the researchers use a collection of data sets obtained by uniformly sampling the space, and use statistical methods to obtain the upper bound with sufficiently high confidence. While the result from [25] can be used here as well to provide probabilistic guarantees, the estimation of the Lipschitz constant using (9) is less data-intensive and does not require uniform sampling of the state space.

B. Robust control under Lipschitz uncertainty

Since the modeling error along the prediction horizon is unknown, we utilize the modeling error at the measurements as a surrogate for the model error along each step of the prediction horizon in (6). Using this notion of *surrogate trajectories*, we obtain a tractable conservative approximation of (6) based on robust optimal control.

Definition 1 (Surrogate trajectory). *Given an open-loop control sequence $\{u_t\}_{t=0}^{T-1}$ and an error sequence $\{\epsilon_t^{\mathrm{sur}}\}_{t=0}^{T-1}$ with $\epsilon_t^{\mathrm{sur}} \in \mathbb{R}^N$, we define a* surrogate *trajectory $\{\Psi_t^{\mathrm{sur}}\}_{t=0}^{T}$ with $\Psi_0^{\mathrm{sur}} = \Psi_0$ as follows,*

$$\Psi_{t+1}^{\mathrm{sur}} = A\Psi_t^{\mathrm{sur}} + Bu_t + \epsilon_t^{\mathrm{sur}}. \tag{10}$$

We refer to $\epsilon_t^{\mathrm{sur}}$ as the surrogate uncertainty at time t.

For any open-loop control sequence, we use the surrogate trajectory to approximate the true lifted trajectory without relying on the unknown modeling error. For every halfspace $p_i \cdot (C\Psi_t) \leq q_i$ in SafeSet, we upper bound $p_i \cdot (C\Psi_t)$ as follows

$$p_i \cdot (C\Psi_t) \leq p_i \cdot (C\Psi_t^{\mathrm{sur}}) + \|p_i \cdot (C(\Psi_t - \Psi_t^{\mathrm{sur}}))\|. \tag{11}$$

by adding and subtracting $(C\Psi_t^{\mathrm{sur}})$. Therefore, for any choice of *back-off variables* $\Gamma_{i,t}$ that satisfies $\|p_i \cdot C(\Psi_t - \Psi_t^{\mathrm{sur}})\| \leq \Gamma_{i,t}$, it is true that

$$p_i \cdot (C\Psi_t^{\mathrm{sur}}) + \Gamma_{i,t} \leq q_i,\ \forall i,t \Longrightarrow C\Psi_t \in \mathrm{SafeSet}. \tag{12}$$

We then use (12) to arrive at the following problem:

$$\begin{aligned}
&\min_{\substack{u_0,\ldots,u_{T-1} \\ \Gamma_{1,1},\ldots,\Gamma_{N_S,T}}} && J(C\Psi_1, \ldots, C\Psi_T, u_0, \ldots, u_{T-1}) && \text{(13a)}\\
&\text{s. t.} && \Psi_T^{\mathrm{sur}} \in \mathrm{TerminalConstraintSet}, && \text{(13b)}\\
& && \begin{cases} \text{Constraints on } \Psi_t^{\mathrm{sur}}, u_t, \Gamma_{i,t} \\ \text{for } \|p_i \cdot C(\Psi_t - \Psi_t^{\mathrm{sur}})\| \leq \Gamma_{i,t} \end{cases} && \text{(13c)}\\
&t \in \mathbb{N}_{[0,T-1]}, && \Psi_{t+1} = A\Psi_t + Bu_t + \epsilon(\Psi_t, u_t), && \text{(13d)}\\
&t \in \mathbb{N}_{[0,T-1]}, && \Psi_{t+1}^{\mathrm{sur}} = A\Psi_t^{\mathrm{sur}} + Bu_t + \epsilon_t^{\mathrm{sur}}, && \text{(13e)}\\
&t \in \mathbb{N}_{[0,T-1]}, && u_t \in \mathcal{U}, && \text{(13f)}\\
&t \in \mathbb{N}_{[1,T]},\ i \in \mathbb{N}_{[1,N_S]}, && p_i \cdot (C\Psi_t^{\mathrm{sur}}) + \Gamma_{i,t} \leq q_i. && \text{(13g)}
\end{aligned}$$

We construct a tractable convex approximation of (6) by 1) characterizing a set $\mathrm{TerminalConstraintSet} \subseteq \mathbb{R}^N$ for the recursive feasibility of (13), and 2) obtaining a collection of convex constraints to enforce (13c) without relying on the unknown modeling error ϵ in (13d).

1) Recursive feasibility constraint (13b)*:* We ensure recursive feasibility of (13) by identifying an invariant set of a Koopman model (2) that is controlled by a pre-determined state-feedback controller. Recall that a set of states $\mathcal{S}$ is said to be invariant for a system $x_{t+1} = g(x_t)$ when $x_0 \in \mathcal{S}$ implies $x_t \in \mathcal{S}$ for all time $t > 0$.

Proposition 2 (Invariant set for Koopman models)**.** *Let the dynamics* (1) *have a stable equilibrium* $(\Psi^{\mathrm{st}}, u^{\mathrm{st}})$*, i.e.,* $f(x^{\mathrm{st}}, u^{\mathrm{st}}) = x^{\mathrm{st}} = C\Psi^{\mathrm{st}}$*. Choose* $K \in \mathbb{R}^{m\times N}$ *satisfying the following convex constraint,*

$$\|A + BK\|_{\mathrm{op}} + L_\epsilon \max\left(\|K\|_{\mathrm{op}}, 1\right) \leq 1. \tag{14}$$

Then, $\mathrm{Ball}(\Psi^{\mathrm{st}}, r)$ *is invariant for a Koopman model* (2) *under the time-invariant, state-feedback control law* $u_t = u^{\mathrm{inv}}(\Psi) = K(\Psi - \Psi^{\mathrm{st}}) + u^{\mathrm{st}}$ *for all time* t*.*

Proof. See Appendix B. □

The convex constraint (14) requires that the state-feedback gain K *contracts* the Koopman model ($\|A+BK\|_{\mathrm{op}} \leq 1$), while also requiring that the modeling error and its effect on the state is minimal ($L_\epsilon \leq 1$ and $L_\epsilon\|K\|_{\mathrm{op}} \leq 1$). However, in order to use Proposition 2 as the terminal constraint in (13), we additionally require that $r > 0$ be such that 1) the state-feedback control law for invariance respects the control constraints, i.e., $u^{\mathrm{inv}}(\Psi) \in \mathcal{U}$ for every $\Psi \in \mathrm{Ball}(\Psi^{\mathrm{st}}, r)$, and 2) the projection of the invariant set on the low-dimensional state space is also safe, i.e., $C\mathrm{Ball}(\Psi^{\mathrm{st}}, r) \subseteq \mathrm{SafeSet}$.

Proposition 3 (Recursive feasibility constraint)**.** *Assume that the input set* $\mathcal{U}$ *be a polytope* ($\mathcal{U} = \cap_{i=1}^{N_U}\{v : h_i \cdot v \leq g_i\}$ *for some* $N_U \in \mathbb{N}, h_i \in \mathbb{R}^m, g_i \in \mathbb{R}$)*. Let the stable equilibrium* $(\Psi^{\mathrm{st}}, u^{\mathrm{st}})$ *used in Proposition 2 be such that* $x^{\mathrm{st}} = C\Psi^{\mathrm{st}} \in \mathrm{SafeSet}$ *and* $u^{\mathrm{st}} \in \mathcal{U}$*, yielding a gain matrix* K *satisfying* (14)*. Then, for* $r_{\max}$ *defined as follows,*

$$r_{\max} = \min\left(\min_{i\in\mathbb{N}_{[1,N_S]}} \frac{q_i - p_i \cdot x^{\mathrm{st}}}{\|C^\top p_i\|}, \min_{i\in\mathbb{N}_{[1,N_U]}} \frac{g_i - h_i \cdot u^{\mathrm{st}}}{\|K^\top h_i\|}\right),$$

the set $\mathrm{TerminalConstraintSet} = \mathrm{Ball}(\Psi^{\mathrm{st}}, r_{\max})$ *ensures recursive feasibility of* (13)*.*

Proof. See Appendix C. □

Using Proposition 3, we have identified a subset of the set SafeSet within which the low-dimensional dynamics are guaranteed to remain invariant. Consequently, enforcing the terminal set constraints on Ψ_T is sufficient to ensure recursive feasibility.

2) Bounding the error between the true and the surrogate trajectories (13c)*:* To upper-bound $\|p_i \cdot C(\Psi_t - \Psi_t^{\mathrm{sur}})\|$, we characterize a sequence of auxiliary decision variables $\{\Delta_t\}_{t=0}^T$ such that $\|\Psi_t - \Psi_t^{\mathrm{sur}}\| \leq \Delta_t$ for every $t \in \mathbb{N}_{[0,T]}$ ($\Delta_0 = 0$ from $\Psi_0 = \Psi_0^{\mathrm{sur}}$). Then, we utilize Δ_t to compute the backoff variables $\Gamma_{i,t}$. Figure 1 illustrates Δ_t.

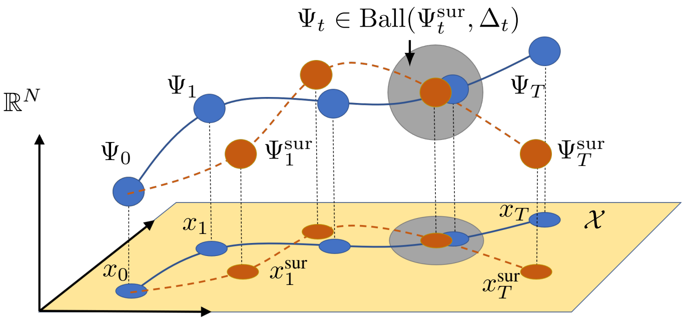

Fig. 1: Illustration of the true trajectory Ψ_t, the surrogate trajectory Ψ_t^{sur}, and the auxiliary decision variable Δ_t that bounds the error between Ψ_t and Ψ_t^{sur} from above.

Proposition 4 (Backoff variables)**.** *Given a sequence of surrogate uncertainty terms* $\{\epsilon_t^{\mathrm{sur}}\}_{t=0}^{T-1}$ *and a sequence of training points* $(\Psi_t^{\mathrm{tr}}, u_t^{\mathrm{tr}})_{t=0}^{T-1}$*, enforce* (13c) *with the following collection of convex constraints in* Ψ_t^{sur}*,* u_t*,* $\Gamma_{i,t}$ *and* Δ_t *defined for every* $i \in \mathbb{N}_{[1,N_S]}$ *and* $t \in \mathbb{N}_{[1,T]}$*,*

$$\Gamma_{i,t} \geq \|p_i \cdot C\| \left(L_\epsilon \left\|\begin{bmatrix} \Psi_{t-1}^{\mathrm{sur}} - \Psi_t^{\mathrm{tr}} \\ u_{t-1} - u_t^{\mathrm{tr}} \end{bmatrix}\right\| + \|\epsilon(\Psi_{t-1}^{\mathrm{tr}}, u_{t-1}^{\mathrm{tr}}) - \epsilon_{t-1}^{\mathrm{sur}}\|\right) + (\|p_i \cdot CA\| + L_\epsilon\|p_i\|)\Delta_{t-1}, \tag{15a}$$

$$\Delta_t \geq L_\epsilon \left\|\begin{bmatrix} \Psi_{t-1}^{\mathrm{sur}} - \Psi_{t-1}^{\mathrm{tr}} \\ u_{t-1} - u_{t-1}^{\mathrm{tr}} \end{bmatrix}\right\| + \|\epsilon(\Psi_{t-1}^{\mathrm{tr}}, u_{t-1}^{\mathrm{tr}}) - \epsilon_{t-1}^{\mathrm{sur}}\| + (\|A\| + L_\epsilon)\Delta_{t-1}. \tag{15b}$$

Then, $\|p_i \cdot C(\Psi_t - \Psi_t^{\mathrm{sur}})\| \leq \Gamma_{i,t}\,\forall\, i \in \mathbb{N}_{[1,N_S]}$ *and* $t \in \mathbb{N}_{[1,T]}$*.*

Proof. Equations (16) and (17) on page 5 follow from triangle inequality, the Lipschitz continuity of ϵ, the definition of the operator norm $\|A\|$, and the observation that $\epsilon_x = C\epsilon$. We obtain (15a) and (15b) by defining $\Gamma_{i,t}$ and Δ_t as upper bounds of the RHS of (16) and (17). □

Proposition 4 proves that (13c) can be conservatively enforced without the use of the unknown true trajectory $\{\Psi_t\}_{t=0}^T$ defined in (13d).

3) Conservative convex approximation of (4)*:* We now formulate the conservative convex approximation of (4), based on the results presented in the previous subsections. Given a sequence of surrogate uncertainty terms $\{\epsilon_t^{\mathrm{sur}}\}_{t=0}^{T-1}$ and a sequence of training points $\{(\Psi_t^{\mathrm{tr}}, u_t^{\mathrm{tr}})\}_{t=0}^{T-1}$, consider the following convex program

$$\begin{array}{ll} \underset{\substack{u_0,\ldots,u_{T-1},\Gamma_{1,1},\ldots,\Gamma_{N_S,T} \\ \Delta_1,\ldots,\Delta_T}}{\mathrm{minimize}} & \text{(13a)} \\ \text{subject to} & \Psi_T \in \mathrm{Ball}(\Psi^{\mathrm{st}}, r_{\max}), \\ & \text{(13e), (13f), (13g), (15).} \end{array} \tag{18}$$

Proposition 5. *Every feasible solution of the convex program* (18) *is feasible for* (6)*.*

Proof. Follows from Proposition 4 and (11). □

By Proposition 5, we conclude that the optimization problem (18) solves Problem 1. The problem (18) can be solved via off-the-shelf solvers [49].

For any time $t \in \mathbb{N}_{[1,T]}$,

$$\begin{aligned}
\|\Psi_t - \Psi_t^{\mathrm{sur}}\| &= \|A\Psi_{t-1} + Bu_{t-1} + \epsilon(\Psi_{t-1}, u_{t-1}) - A\Psi_{t-1}^{\mathrm{sur}} - Bu_{t-1} - \epsilon_{t-1}^{\mathrm{sur}}\| \\
&= \|A\Psi_{t-1} + Bu_{t-1} + \epsilon(\Psi_{t-1}, u_{t-1}) - A\Psi_{t-1}^{\mathrm{sur}} - Bu_{t-1} - \epsilon_{t-1}^{\mathrm{sur}} + \epsilon(\Psi_{t-1}^{\mathrm{sur}}, u_{t-1}) - \epsilon(\Psi_{t-1}^{\mathrm{sur}}, u_{t-1})\| \\
&\leq \|A(\Psi_{t-1} - \Psi_{t-1}^{\mathrm{sur}})\| + \|\epsilon(\Psi_{t-1}, u_{t-1}) - \epsilon(\Psi_{t-1}^{\mathrm{sur}}, u_{t-1})\| + \|\epsilon(\Psi_{t-1}^{\mathrm{sur}}, u_{t-1}) - \epsilon_{t-1}^{\mathrm{sur}}\| \\
&\leq (\|A\| + L_\epsilon)\|\Psi_{t-1} - \Psi_{t-1}^{\mathrm{sur}}\| + \|\epsilon(\Psi_{t-1}^{\mathrm{sur}}, u_{t-1}) - \epsilon_{t-1}^{\mathrm{sur}} + \epsilon(\Psi_{t-1}^{\mathrm{tr}}, u_{t-1}^{\mathrm{tr}}) - \epsilon(\Psi_{t-1}^{\mathrm{tr}}, u_{t-1}^{\mathrm{tr}})\| \\
&\leq (\|A\| + L_\epsilon)\|\Psi_{t-1} - \Psi_{t-1}^{\mathrm{sur}}\| + L_\epsilon \left\| \begin{bmatrix} \Psi_{t-1}^{\mathrm{sur}} - \Psi_t^{\mathrm{tr}} \\ u_{t-1} - u_t^{\mathrm{tr}} \end{bmatrix} \right\| + \|\epsilon(\Psi_{t-1}^{\mathrm{tr}}, u_{t-1}^{\mathrm{tr}}) - \epsilon_{t-1}^{\mathrm{sur}}\|.
\end{aligned} \tag{16}$$

For any time $t \in \mathbb{N}_{[1,T]}$ and $i \in \mathbb{N}_{[1,N_S]}$,

$$\begin{aligned}
\|p_i \cdot C(\Psi_t - \Psi_t^{\mathrm{sur}})\| &= \|p_i \cdot C(A\Psi_{t-1} + Bu_{t-1} + \epsilon(\Psi_{t-1}, u_{t-1}) - A\Psi_{t-1}^{\mathrm{sur}} - Bu_{t-1} - \epsilon_{t-1}^{\mathrm{sur}})\| \\
&= \|p_i \cdot C(A\Psi_{t-1} + Bu_{t-1} + \epsilon(\Psi_{t-1}, u_{t-1}) - A\Psi_{t-1}^{\mathrm{sur}} - Bu_{t-1} - \epsilon_{t-1}^{\mathrm{sur}} + \epsilon(\Psi_{t-1}^{\mathrm{sur}}, u_{t-1}) - \epsilon(\Psi_{t-1}^{\mathrm{sur}}, u_{t-1}))\| \\
&= \|p_i \cdot CA(\Psi_{t-1} - \Psi_{t-1}^{\mathrm{sur}}) + p_i \cdot (\epsilon_x(\Psi_{t-1}, u_{t-1}) - \epsilon_x(\Psi_{t-1}^{\mathrm{sur}}, u_{t-1})) + p_i \cdot C(\epsilon(\Psi_{t-1}^{\mathrm{sur}}, u_{t-1}) - \epsilon_{t-1}^{\mathrm{sur}})\| \\
&\leq \|p_i \cdot CA(\Psi_{t-1} - \Psi_{t-1}^{\mathrm{sur}})\| + \|p_i\|\|\epsilon_x(\Psi_{t-1}, u_{t-1}) - \epsilon_x(\Psi_{t-1}^{\mathrm{sur}}, u_{t-1})\| + \|p_i \cdot C\|\|\epsilon(\Psi_{t-1}^{\mathrm{sur}}, u_{t-1}) - \epsilon_{t-1}^{\mathrm{sur}}\| \\
&\leq (\|p_i \cdot CA\| + L_\epsilon\|p_i\|)\|\Psi_{t-1} - \Psi_{t-1}^{\mathrm{sur}}\| + \|p_i \cdot C\| \left(L_\epsilon \left\| \begin{bmatrix} \Psi_{t-1}^{\mathrm{sur}} - \Psi_t^{\mathrm{tr}} \\ u_{t-1} - u_t^{\mathrm{tr}} \end{bmatrix} \right\| + \|\epsilon(\Psi_{t-1}^{\mathrm{tr}}, u_{t-1}^{\mathrm{tr}}) - \epsilon_{t-1}^{\mathrm{sur}}\| \right).
\end{aligned} \tag{17}$$

Algorithm 1: Robust Koopman MPC

```
Input: Training points ${\Psi_i^{tr}, u_i^{tr}, (\Psi_i^{tr})^+}_{i=1}^{N_D}$, initial lifted
    state $\Psi_0$, SafeSet, Stable eq. $(\Psi^{st}, u^{st})$, cost function $J$.
Compute $(A, B, C)$ using ${\Psi_i^{tr}, u_i^{tr}, (\Psi_i^{tr})^+}_{i=1}^{N_D}$,
Compute TerminalConstraintSet using Prop. 3,
for each iteration of MPC ▷ Current time is set to zero
    Compute $\epsilon_t^{sur}$ and ${(\Psi_t^{tr}, u_t^{tr})}_{t=0}^{T}$ using (19)
    Solve (18) to obtain ${u_t}_{t=0}^{T-1}$
    Apply $u_0$
```

Next, we discuss a selection strategy for the sequence of training points $(\Psi_t^{\mathrm{tr}}, u_t^{\mathrm{tr}})_{t=0}^{T-1}$ and the sequence of surrogate uncertainty terms $\{\epsilon_t^{\mathrm{sur}}\}_{t=0}^{T-1}$. We first solve the following convex program,

$$\begin{aligned}
\underset{u_0,\ldots,u_{T-1}}{\text{minimize}} \quad & \sum_{t=1}^{T} (C\Psi_t - x_0) \cdot Q(C\Psi_t - x_0)) + \sum_{t=0}^{T-1} u_t \cdot (Ru_t) \\
\text{subject to} \quad & \\
t \in \mathbb{N}_{[0,T-1]}, \quad & \Psi_{t+1} = A\Psi_t + Bu_t,\ u_t \in \mathcal{U}, \\
t \in \mathbb{N}_{[1,T]}, \quad & C\Psi_t \in \text{SafeSet},
\end{aligned} \tag{19}$$

where the cost in (19) penalizes the deviations from the current state ($t = 0$) instead of the target state. We then compute the sequence of training points $\{(\Psi_t^{\mathrm{tr}}, u_t^{\mathrm{tr}})\}_{t=1}^{T}$ that is closest to the solution of (19), and set $\epsilon_t^{\mathrm{sur}} = \epsilon(\Psi_t^{\mathrm{tr}}, u_t^{\mathrm{tr}})$. The optimization problem (19) is inspired from (7), but modified to avoid the potential infeasibility when attempting to move towards the target. Instead, (19) seeks an open-loop control sequence that keeps the system safe, while minimizing deviations from its current position. Consequently, (19) typically remains feasible, even when the (7) is infeasible.

We summarize the proposed approach in Algorithm 1.

C. Strengths and weaknesses of the proposed approach

Our approach has several advantages over existing Koopman approaches [45]. First, our approach considers the modeling error ϵ in the state constraint and recursive feasibility enforcement explicitly. Second, our approach can be easily implemented for real-time control, thanks to the convex formulation (18) of the model predictive control problem (6). Third, our approach utilizes Hankel Koopman modeling, which frees the designer from the need to select/hand-craft the basis functions for the Koopman-based control.

The main shortcoming of our approach is the reliance on Lipschitz continuity of the uncertainty (Assumption 1). The assumption of Lipschitz continuity of unknown functions is becoming more prevalent in data-driven control literature for the purposes of rigorous analysis [25], [50]. However, the presented theoretical guarantees (see Proposition 4) rests on the use of an exact value or upper bound of the Lipschitz constant L_ϵ, which is typically hard to obtain. Our numerical experiements show that the lower bound on the Lipschitz constant (9) may be sufficient in practice to obtain approximate safety guarantees.

IV. Numerical results

In this section, we demonstrate that the proposed RK-MPC approach has the superior constraint satisfaction of over the nominal MPC, without compromising real-time implementability. All computations were done in Python using `CVXPY` [52], `ECOS` [53], and `CVXOPT` [49].

A. Forced Duffing Oscillator

Consider states $x = [x_p, y_p] \in \mathbb{R}^2$ and control $u \in \mathbb{R}$ with dynamics given by

$$\frac{dx}{dt} = \begin{bmatrix} y_p \\ ax_p + bx_p^3 + dy_p + u, \end{bmatrix} \tag{20}$$

where $a = 1, b = -1$, and $d = -0.5$. To train a Koopman model, we choose delay basis functions with $n_d = 3$, sampling time $t_s = 0.1$, planning horizon $T = 10$. To generate training data, we simulate 5 training trajectories by propagating the dynamics for 6 steps with random control, resulting in $N_D = 30$. Each training trajectory starts from the initial condition specified for evaluation. We compare nominal and robust Koopman MPC for two cases: initial

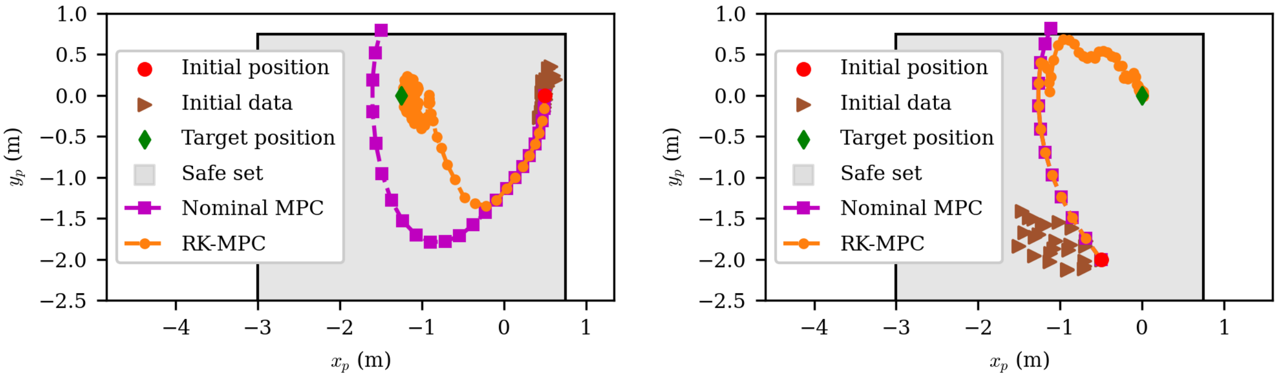

Fig. 2: Regulation of the forced Duffing oscillator using nominal and robust Koopman MPC from two initial conditions.

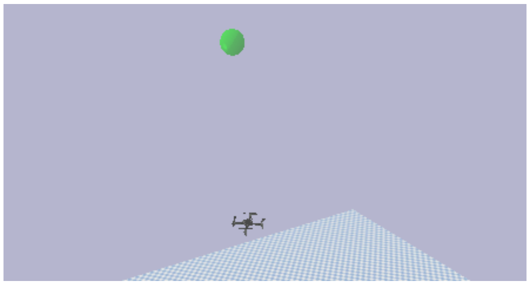

Fig. 3: Data-driven waypoint tracking for a quadrotor using robust Koopman model predictive control. We use the `gym-pybullets-drones` environment [51] for simulating the proposed data-driven approach.

Experiment	Nominal MPC			RK-MPC (Ours)		
Duffing oscillator	0.005	0.012	0.061	0.012	0.016	0.324
Quadrotor waypoint	0.027	0.036	0.217	0.062	0.079	1.184

TABLE I: Min, mean, and max compute times for MPC iterations (in seconds) measured on a standard laptop (Intel i7-10510 CPU, 16 GB RAM).

conditions $[0.5, 0]$ and $[-0.5, -2]$, and target states $[-1.25, 0]$ and $[0, 0]$ respectively. We select the state constraints are $-3 \leq x_i \leq 0.75$ for $i = 1, 2$.

Fig. 2 shows that RK-MPC reaches the target while satisfying the state constraints, whereas the nominal Koopman MPC fails. We observe that in the first case, the nominal Koopman MPC fails to reach the target, while in the second case it results in constraint violation. The superior performance of RK-MPC may be attributed to the explicit consideration of the model uncertainty, leading to a conservative state constraint enforcement.

Table I shows that RK-MPC is real-time implementable, despite the explicit consideration of the modeling errors. On average, RK-MPC solved the MPC problem at ≈ 62.5 Hz.

We emphasize that the quality of the data-driven model plays an important role in these results. The Lipschitz constant L_ϵ^- was estimated from the data to be 0.0013 and 0.0003, respectively. One can expect that poor data-driven models to have larger Lipschitz constants, which can lead to a higher degree of conservativeness in the state constraint enforcement of RK-MPC.

B. Waypoint regulation of a quadrotor

Next, we demonstrate robust Koopman model predictive control to navigate a quadrotor to a desired waypoint using only data. For the simulation, we use `gym-pybullets-drones` framework, that provides a simulation environment based on Bitcraze's CrazyFlies [51]. The quadrotor states are $x = [x_p, y_p, z_p, \theta, \psi, \phi, v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^\top \in \mathbb{R}^{12}$, where x_p, y_p, z_p are the world-frame coordinates, θ, ψ, ϕ express the world-frame Euler angles, v_x, v_y, v_z are the world-frame velocities, and $\omega_x, \omega_y, \omega_z$ are the world-frame angular velocities. To train a Koopman model, we choose delay basis functions with $n_d = 2$, sampling time $t_s = 0.05$, planning horizon $T = 15$. To generate training data, we simulate 5 training trajectories by propagating the dynamics for 10 steps with random control, resulting in $N_D = 50$. We tested the approaches to navigate the quadrotor from its initial hover position at $(0, 0, 60)$ m to a target hover position at $(5, 5, 70)$ m. We define the operating regions as follows: $-10 \leq x_p, y_p \leq 10$ and $10 \leq z_p \leq 200$.

Fig. 3 shows the resulting trajectories from RK-MPC (proposed approach) and the nominal Koopman MPC. We found that the proposed approach is able to driven the quadrotor towards the target despite the limited amount of data points (50 training points). We also found that the nominal Koopman MPC fails to satisfy the state constraints early on, possibly due to the modeling errors. Table I show the computational effort for RK-MPC does not increase substantially when more complex, high-dimensional dynamics are considered.

V. Conclusion and Future Work

This paper proposes a novel robust Koopman model predictive control approach, using Lipschitz analysis and convex optimization. Our numerical simulations show that the proposed approach is real-time implementable and accommodates modeling errors arising in low-data regimes.

Our future work will investigate potential avenues to further reduce the computational requirements as well as the conservativeness. For example, we can further reduce

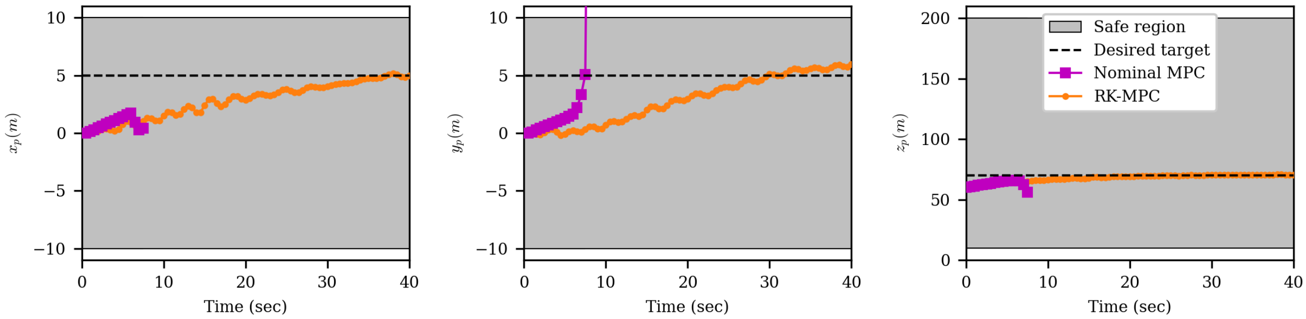

Fig. 4: Quadrotor control based on nominal Koopman MPC and the proposed RK-MPC. The proposed RK-MPC conservatively maneuvers the quadrotor and reaches the goal, without any state constraint violation. On the other hand, the nominal MPC fails to satisfy the state constraints.

the computational complexity in solving (18) by formulating it as a quadratic program instead of the second-order cone program after replacing the 2-norms with 1-norms in (15). We can also reduce the conservativeness by considering alternative heuristics for the selection of training points and the surrogate uncertainty, including iterative procedures. We will also compare the proposed approach with the recent work in tube-based approaches [34].

VI. Acknowledgement

The authors are grateful for the insightful discussions with Dr. Rien Quirynen during the early stages of the project.

Appendix

A. Proof of Proposition 1

We need to show that $\|\epsilon(\Psi,u) - \epsilon(\Psi^{\mathrm{tr}},u^{\mathrm{tr}})\| \le (L_f + \|C[A\ \ B]\|)\left\|\begin{bmatrix}\Psi-\Psi^{\mathrm{tr}}\\ u-u^{\mathrm{tr}}\end{bmatrix}\right\|$ for any two lifted-state-control tuples (Ψ,u) and $(\Psi^{\mathrm{tr}},u^{\mathrm{tr}})$. By the structure of ϵ, we have $\|\epsilon(\Psi,u) - \epsilon(\Psi^{\mathrm{tr}},u^{\mathrm{tr}})\| = \|\epsilon_x(\Psi,u) - \epsilon_x(\Psi^{\mathrm{tr}},v)\|$, which implies that the Lipschitz constants of ϵ and ϵ_x are identical. We have

$$\begin{aligned}&\|\epsilon_x(\Psi,u) - \epsilon_x(\Psi^{\mathrm{tr}},u^{\mathrm{tr}})\|\\ &= \|f(C\Psi,u) - CA\Psi - CBu - f(C\Psi^{\mathrm{tr}},u^{\mathrm{tr}}) + CA\Psi^{\mathrm{tr}} + CBv\|\\ &\le \|f(C\Psi,u) - f(C\Psi^{\mathrm{tr}},v)\| + \left\|C[A\ \ B]\begin{bmatrix}\Psi-\Psi^{\mathrm{tr}}\\ u-v\end{bmatrix}\right\|\\ &\le L_f\left\|\begin{bmatrix}C(\Psi-\Psi^{\mathrm{tr}})\\ u-v\end{bmatrix}\right\| + \|C[A\ \ B]\|\left\|\begin{bmatrix}\Psi-\Psi^{\mathrm{tr}}\\ u-v\end{bmatrix}\right\|\\ &\le (L_f + \|C[A\ \ B]\|)\left\|\begin{bmatrix}\Psi-\Psi^{\mathrm{tr}}\\ u-v\end{bmatrix}\right\|. \qquad (21)\end{aligned}$$

Here, we have used the triangle inequality, Cauchy-Schwarz inequality and the observation that $\left\|\begin{bmatrix}C(\Psi-\Psi^{\mathrm{tr}})\\ u-v\end{bmatrix}\right\| \le \left\|\begin{bmatrix}\Psi-\Psi^{\mathrm{tr}}\\ u-v\end{bmatrix}\right\|$.

B. Proof of Proposition 2

We need to show that for every $\Psi_t \in \mathrm{Ball}(\Psi^{\mathrm{st}},r)$, we have $\Psi_{t+1} \in \mathrm{Ball}(\Psi^{\mathrm{st}},r)$ under the control law $u = K(\Psi - \Psi^{\mathrm{st}}) + u^{\mathrm{st}}$ and dynamics (2a) at all times t. By the stability assumption of $(\Psi^{\mathrm{st}},u^{\mathrm{st}})$, we have

$$\Psi^{\mathrm{st}} = f(\Psi^{\mathrm{st}},u^{\mathrm{st}}) = A\Psi^{\mathrm{st}} + Bu^{\mathrm{st}} + \epsilon(\Psi^{\mathrm{st}},u^{\mathrm{st}}). \qquad (22)$$

We obtain a bound on $\|\Psi_{t+1} - \Psi^{\mathrm{st}}\|$ by considering the application of u^{inv} on the linear dynamics (2), Lipschitz continuity of the modeling error ϵ, and Cauchy-Schwarz inequality,

$$\begin{aligned}&\|\Psi_{t+1} - \Psi^{\mathrm{st}}\|\\ &= \Big\|\Big(A(\Psi_t - \Psi^{\mathrm{st}}) + B(u^{\mathrm{inv}}(\Psi_t) - u^{\mathrm{st}})\\ &\qquad\qquad + \epsilon(\Psi_t,u^{\mathrm{inv}}(\Psi_t)) - \epsilon(\Psi^{\mathrm{st}},u^{\mathrm{st}})\Big)\Big\|\\ &= \|(A+BK)(\Psi_t - \Psi^{\mathrm{st}}) + \epsilon(\Psi_t,u^{\mathrm{inv}}(\Psi_t)) - \epsilon(\Psi^{\mathrm{st}},u^{\mathrm{st}}))\|\\ &\le \|(A+BK)(\Psi_t - \Psi^{\mathrm{st}})\| + L_\epsilon\left\|\begin{bmatrix}\Psi_t - \Psi^{\mathrm{st}}\\ K(\Psi_t - \Psi^{\mathrm{st}})\end{bmatrix}\right\|\\ &= \|(A+BK)(\Psi_t - \Psi^{\mathrm{st}})\| + L_\epsilon\left\|\begin{bmatrix}I_N & 0\\ 0 & K\end{bmatrix}(\Psi_t - \Psi^{\mathrm{st}})\right\|\\ &\le \left(\|A+BK\|_{\mathrm{op}} + L_\epsilon\left\|\begin{bmatrix}I_N & 0\\ 0 & K\end{bmatrix}\right\|_{\mathrm{op}}\right)\|\Psi_t - \Psi^{\mathrm{st}}\|.\end{aligned}$$

Recall that $\left\|\begin{bmatrix}I_N & 0\\ 0 & K\end{bmatrix}\right\|_{\mathrm{op}} \le \max\left(\|K\|_{\mathrm{op}},1\right)$ due to its block diagonal structure. Since K satisfies (14), we conclude that $\|\Psi_{t+1} - \Psi^{\mathrm{st}}\| \le \|\Psi_t - \Psi^{\mathrm{st}}\|$. Consequently, the set $\mathrm{Ball}(\Psi^{\mathrm{st}},r)$ with any $r > 0$ is a control invariant set for the Koopman model (2).

C. Proof of Proposition 3

We compute $r_{\max}$ by solving the following problem,

$$\begin{aligned}\max.\quad & r\\ \mathrm{s.\ t.}\quad & r \ge 0,\\ & u^{\mathrm{inv}}(\Psi) \in \mathcal{U}, \qquad \forall\Psi \in \mathrm{Ball}(\Psi^{\mathrm{st}},r),\\ & C\Psi \in \mathsf{SafeSet}, \qquad \forall\Psi \in \mathrm{Ball}(\Psi^{\mathrm{st}},r).\end{aligned} \qquad (23)$$

Using arguments similar to Chebyshev centering of polytopes [54], we obtain the following linear program in r by

$$\begin{aligned}\max.\quad & r\\ \mathrm{s.\ t.}\quad & r \ge 0,\\ & p_i \cdot (C\Psi^{\mathrm{st}}) + \|C^\top p_i\| r \le q_i,\ \ \forall i \in \mathbb{N}_{[1,N_S]},\\ & h_i \cdot u^{\mathrm{st}} + \|K^\top h_i\| r \le g_i,\ \ \forall i \in \mathbb{N}_{[1,N_U]}.\end{aligned} \qquad (24)$$

Since (24) is a linear program in a scalar variable, the optimal solution of (24) is characterized by the active constraint. This completes the proof.

References

[1] A. Verl, A. Albu-Schäffer, O. Brock, and A. Raatz, *Soft Robotics*. Springer, 2015.

[2] D. Haggerty, M. Banks, P. Curtis, I. Mezić, and E. Hawkes, "Modeling, reduction, and control of a helically actuated inertial soft robotic arm via the Koopman operator," *arXiv preprint arXiv:2011.07939*, 2020.

[3] A. Broad, T. D. Murphey, and B. D. Argall, "Learning models for shared control of human-machine systems with unknown dynamics," in *Rob.: Sci. & Syst.*, 2017.

[4] W. Wang, X. Dai, L. Li, B. Gheneti, Y. Ding, J. Yu, and G. Xie, "Three-dimensional modeling of a fin-actuated robotic fish with multimodal swimming," *IEEE Trans. Mech.*, vol. 23, no. 4, pp. 1641–1652, 2018.

[5] M. Zefran, J. Desai, and V. Kumar, "Continuous motion plans for robotic systems with changing dynamic behavior," *Robotic motion and manipulation*, pp. 113–128, 1997.

[6] M. Zefran and J. W. Burdick, "Design of switching controllers for systems with changing dynamics," in *Conf. Dec. Ctrl.*, vol. 2, 1998.

[7] T. Kunz, U. Reiser, M. Stilman, and A. Verl, "Real-time path planning for a robot arm in changing environments," in *Int'l Conf. Intelligent Rob. Syst.*, 2010, pp. 5906–5911.

[8] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Info. Sci.*, 2013.

[9] S. Brunton, J. Proctor, and N. Kutz, "Sparse identification of nonlinear dynamics with control (SINDYc)," *IFAC-PapersOnLine*, 2016.

[10] B. Lusch, N. Kutz, and S. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, 2018.

[11] E.-M. Hong, Y. Pachepsky, G. Whelan, and T. Nicholson, "Simpler models in environmental studies and predictions," *Critical Rev. Environ. Sci. Tech.*, vol. 47, no. 18, pp. 1669–1712, 2017.

[12] S. Kamthe and M. Deisenroth, "Data-efficient reinforcement learning with probabilistic model predictive control," in *Int'l Conf. Artificial Intelligence Stat.*, 2018, pp. 1701–1710.

[13] S. Dean, S. Tu, N. Matni, and B. Recht, "Safely learning to control the constrained linear quadratic regulator," in *Amer. Ctrl. Conf.*, 2019.

[14] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A Lyapunov-based approach to safe reinforcement learning," in *Neural Info. Proc. Syst.*, 2018, pp. 8103–8112.

[15] R. Cheng, G. Orosz, R. Murray, and J. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *AAAI Conf. Artificial Intelligence*, 2019.

[16] B. Lütjens, M. Everett, and J. P. How, "Safe reinforcement learning with model uncertainty estimates," in *Int'l Conf. Rob. Autom.*, 2019.

[17] M. Wen and U. Topcu, "Constrained cross-entropy method for safe reinforcement learning," *Trans. Autom. Ctrl.*, 2020.

[18] M. Zanon and S. Gros, "Safe reinforcement learning using robust MPC," *Trans. Autom. Ctrl.*, 2020.

[19] Z. Marvi and B. Kiumarsi, "Safe reinforcement learning: A control barrier function optimization approach," *Int'l J. Rob. Nonlin. Ctrl.*, vol. 31, no. 6, pp. 1923–1940, 2021.

[20] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *Conf. Dec. Ctrl.*, 2020, pp. 3717–3724.

[21] 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, pp. 708–717.

[22] A. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Euro. Ctrl. Conf.*, 2019, pp. 3420–3431.

[23] M. Srinivasan, A. Dabholkar, S. Coogan, and P. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," *arXiv preprint arXiv:2003.04950*, 2020.

[24] C. Folkestad, Y. Chen, A. Ames, and J. Burdick, "Data-driven safety-critical control: Synthesizing control barrier functions with Koopman operators," *Ctrl. Syst. Lett.*, 2020.

[25] C. Knuth, G. Chou, N. Ozay, and D. Berenson, "Planning with learned dynamics: Probabilistic guarantees on safety and reachability via lipschitz constants," *Rob. Autom. Lett.*, pp. 5129–5136, 2021.

[26] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*. Springer, 1999.

[28] M. Bujarbaruah, S. Nair, and F. Borrelli, "A semi-definite programming approach to robust adaptive MPC under state dependent uncertainty," in *Euro. Ctrl. Conf.*, 2020, pp. 960–965.

[27] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Robust constraint satisfaction in data-driven MPC," in *Conf. Dec. Ctrl.*, 2020.

[29] J. Köhler, R. Soloperto, M. Muller, and F. Allgower, "A computationally efficient robust model predictive control framework for uncertain nonlinear systems," *Trans. Autom. Ctrl.*, 2020.

[30] M. Lorenzen, M. Cannon, and F. Allgöwer, "Robust MPC with recursive model update," *Automatica*, vol. 103, pp. 461–471, 2019.

[31] J. Berberich, J. Köhler, M. Muller, and F. Allgower, "Data-driven model predictive control with stability and robustness guarantees," *Trans. Autom. Ctrl.*, 2020.

[32] J. Köhler, P. Kötting, R. Soloperto, F. Allgöwer, and M. Müller, "A robust adaptive model predictive control framework for nonlinear uncertain systems," *Int'l J. Robust Nonlin. Ctrl.*, 2020.

[33] J. Manzano, D. Limon, D. Peña, and J.-P. Calliess, "Robust learning-based MPC for nonlinear constrained systems," *Autom.*, 2020.

[34] X. Zhang, W. Pan, R. Scattolini, S. Yu, and X. Xu, "Robust tube-based model predictive control with Koopman operators–extended version," *arXiv preprint arXiv:2108.13011*, 2021.

[35] J. Proctor, S. Brunton, and N. Kutz, "Generalizing Koopman theory to allow for inputs and control," *J. App. Dyn. Syst.*, vol. 17, 2018.

[36] E. Kaiser, N. Kutz, and S. Brunton, "Data-driven discovery of Koopman eigenfunctions for control," *Machine Learn.: Sci. Tech.*, 2021.

[37] S. Brunton, B. Brunton, J. Proctor, and N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PloS one*, vol. 11, no. 2, 2016.

[38] N. Takeishi, Y. Kawahara, and T. Yairi, "Learning Koopman invariant subspaces for dynamic mode decomposition," in *Neural Info. Proc. Syst.*, 2017, pp. 1130–1140.

[39] S. Pan, N. Arnold-Medabalimi, and K. Duraisamy, "Sparsity-promoting algorithms for the discovery of informative Koopman-invariant subspaces," *J. Fluid Mech.*, vol. 917, 2021.

[40] M. Bonnert and U. Konigorski, "Estimating Koopman invariant subspaces of excited systems using artificial neural networks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1156–1162, 2020.

[41] M. Haseli and J. Cortés, "Parallel learning of Koopman eigenfunctions and invariant subspaces for accurate long-term prediction," *Trans. Ctrl Net. Syst.*, 2021.

[42] R. Rabben, S. Ray, and M. Weber, "ISOKANN: Invariant subspaces of Koopman operators learned by neural network," *J. Chem. Phy.*, 2020.

[43] M. Haseli and J. Cortés, "Data-driven approximation of Koopman-invariant subspaces with tunable accuracy," in *Amer. Ctrl. Conf.*, 2021.

[44] G. Mamakoukas, M. Castano, X. Tan, and T. Murphey, "Derivative-based Koopman operators for real-time control of robotic systems," *Trans. Rob.*, 2021.

[45] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[46] D. Bruder, B. Gillespie, D. Remy, and R. Vasudevan, "Modeling and control of soft robots using the Koopman operator and model predictive control," *Rob. Sci. Syst.*, 2019.

[47] M. Budišić, R. Mohr, and I. Mezić, "Applied Koopmanism," *Chaos: An Interdisciplinary J. Nonlin. Sci.*, vol. 22, no. 4, 2012.

[48] H. Arbabi and I. Mezic, "Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator," *J. App. Dyn. Syst.*, vol. 16, no. 4, pp. 2096–2126, 2017.

[49] M. Andersen, J. Dahl, L. Vandenberghe *et al.*, "CVXOPT: A python package for convex optimization." [Online]. Available: cvxopt.org

[50] F. Djeumou, A. Vinod, E. Goubault, S. Putot, and U. Topcu, "On-the-fly control of unknown smooth systems from limited data," in *Amer. Ctrl. Conf.*, 2021, pp. 3656–3663.

[51] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. Schoellig, "Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control," in *Int'l Conf. Intelligent Rob. Syst.*, 2021.

[52] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Machine Learn. Res.*, 2016.

[53] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *Euro. Ctrl. Conf.*, 2013, pp. 3071–3076.

[54] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ. Press, 2004.