

Koopman Data-Driven Predictive Control with Robust Stability and Recursive Feasibility Guarantees

Citation for published version (APA):

de Jong, T., Breschi, V., Schoukens, M., & Lazar, M. (2024). *Koopman Data-Driven Predictive Control with Robust Stability and Recursive Feasibility Guarantees*. arXiv.org. <https://doi.org/10.48550/arXiv.2405.01292>

DOI:

[10.48550/arXiv.2405.01292](https://doi.org/10.48550/arXiv.2405.01292)

Document status and date:

Published: 02/05/2024

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Koopman Data-Driven Predictive Control with Robust Stability and Recursive Feasibility Guarantees

Thomas de Jong¹, Valentina Breschi¹, Maarten Schoukens¹ and Mircea Lazar¹

Abstract—In this paper, we consider the design of data-driven predictive controllers for nonlinear systems from input-output data via linear-in-control input Koopman lifted models. Instead of identifying and simulating a Koopman model to predict future outputs, we design a subspace predictive controller in the Koopman space. This allows us to learn the observables minimizing the multi-step output prediction error of the Koopman subspace predictor, preventing the propagation of prediction errors. To avoid losing feasibility of our predictive control scheme due to prediction errors, we compute a terminal cost and terminal set in the Koopman space and we obtain recursive feasibility guarantees through an interpolated initial state. As a third contribution, we introduce a novel regularization cost yielding input-to-state stability guarantees with respect to the prediction error for the resulting closed-loop system. The performance of the developed Koopman data-driven predictive control methodology is illustrated on a nonlinear benchmark example from the literature.

I. INTRODUCTION

By leveraging the linearity of the predictor in the lifted Koopman space, linear-in-control input Koopman Model Predictive Control (MPC) represents a promising and appealing approach to nonlinear MPC. Indeed, the shift to the Koopman space allows one to convert the nonlinear predictive control problem into a quadratic program (QP), which can then be solved effectively and with high precision. This approach was originally proposed in [1], where the Koopman operator is extended to controlled dynamical systems using Extended Dynamic Mode Decomposition (EDMD) and then used for predictive control purposes. It has then been extended to consider full-state measurements and input-output scenarios, as discussed in [1], [2].

Despite its advantages, approximating a nonlinear system with finite-dimensional linear-in-control input Koopman modeling has its limitations. Indeed, it leads to models that can be incomplete, as shown in [3]. Moreover, while the Koopman operator provides coordinate transformations that extend the local neighborhoods where a linear model is valid to the full basin around them [4], these models cannot capture the global dynamics of nonlinear systems with multiple equilibria, as any linear operator has either one fixed point at the origin or a subspace of infinitely many fixed points [5], [6].

This inherent incompleteness of linear-in-control input Koopman models results in discrepancies between real-life nonlinear system outputs and Koopman model predictions,

which can be detrimental to predictive control design. In particular, it is no longer possible to guarantee stability and recursive feasibility using just terminal ingredients in the Koopman space. A solution to this problem based on robust tube MPC design in the Koopman space was recently presented in [7]. While offering robust stability and recursive feasibility guarantees to Koopman model-based predictive control, this approach relies on tightened constraints in the Koopman space, which might lead to arbitrary conservatism as constraints on the observables do not relate linearly to the original constraints on outputs or states. Moreover, it requires the measurability of the controlled system state and it postulates the availability of an accurate Koopman model.

These models can be identified from data using system identification methods, that however typically look at fitting one-step linear state-space dynamics, i.e., the system matrices [1], [8], [9]. As a result, prediction errors propagate as the prediction horizon increases. This limitation is overcome in data-driven predictive control (DPC) methods, whose aim is to minimize predicted tracking errors and control efforts over the entire prediction horizon [10], [11].

A recent observation in [12] intertwines linear-in-control input Koopman Data-driven Predictive Control and DPC, highlighting that the former can be reformulated as subspace predictive control (SPC) in the Koopman space. The SPC approach to data-driven predictive control comes with several benefits. It removes the propagation of prediction errors, as separate multi-step predictors are identified jointly. The resulting prediction matrices are least squares optimal, yielding unbiased predictors. Last but not least, the resulting predictors are linear in the predicted control input sequence.

Motivated by these insights, in this work we propose a linear-in-control input SPC formulation in the Koopman space, which consists of three main design steps. Firstly, we parameterize the observables as basis or kernel functions that depend on past input and past/current output data of the nonlinear system. We solve a nonlinear least squares problem to learn the observables, aiming to minimize the multi-step output prediction error of the Koopman SPC predictors. It is important to note that our approach differs from that in reference [13]. While [13] identifies a model that requires iterating the Koopman state-space equations N times to obtain the N -step ahead prediction, our method eliminates the need for such iteration for multi-step ahead predictions. Thirdly, we use least squares identification to compute the optimal multi-step prediction matrices allowing us to predict the evolution of the full Koopman lifted state, rather than the output, over the prediction horizon. This last choice

¹ Department of Electrical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands. E-mails of the authors: t.o.d.jong@tue.nl, v.breschi@tue.nl, m.schoukens@tue.nl, m.lazar@tue.nl.

is advantageous because it enables the computation of a terminal cost and terminal set in the Koopman space. At the same time, since the linear-in-control input Koopman predictors are still incomplete, the corresponding prediction errors might still result in a loss of feasibility despite using a terminal set in the Koopman space. To circumvent this issue, inspired by the solution proposed in [14] for robust stochastic MPC, we define the initial state of the Koopman model as the interpolation of the measurements based lifted state and the previously one-step ahead predicted Koopman state. We prove that this choice leads to a recursively feasible Koopman data-driven predictive control scheme, despite the prediction error. Based on this definition of the initial state, we also develop a novel regularization cost and establish that the original nonlinear system in closed-loop with the Koopman data-driven predictive controller is input-to-state stable (ISS) [15] with respect to the prediction error.

The remainder of the paper is organized as follows. After introducing the preliminaries and our problem statement in Section II, we present methods for learning observables within the Koopman formulation in Section III. To address potential plant-model mismatches stemming from the linear-in-control input Koopman formulation, we develop robust stability and recursive feasibility conditions in Section IV. Finally, in Section V, we substantiate the effectiveness of our proposed control scheme through a nonlinear example.

Notation and basic definitions: Let \mathbb{R} , \mathbb{R}_+ and \mathbb{N} denote the set of real numbers, the set of non-negative reals, the set of non-negative integers. For every $c \in \mathbb{R}$ and $\Pi \subseteq \mathbb{R}$ define $\Pi_{\geq c} := \{k \in \Pi \mid k \geq c\}$ and let $\mathbf{w}_{[0,k]} := \{w(l)\}_{l \in \mathbb{Z}_{[0,k]}}$. Throughout this paper, for any finite number $q \in \mathbb{N}_{\geq 1}$ of column vectors or functions $\{\xi_1, \dots, \xi_q\}$ we will make use of the operator $\text{col}(\xi_1, \dots, \xi_q) := [\xi_1^\top, \dots, \xi_q^\top]^\top$ and for a square matrix A we use the operator $\text{blkdiag}(n, A)$ defined as the the block diagonal matrix with n matrices A on the diagonal. We denote by $\rho(A)$ for some square matrix A the maximum absolute value of the eigenvalues of A . For $c_1 \in \mathbb{R}_{>0}$, a function $\varphi : \mathbb{R}_{[0,c_1]} \rightarrow \mathbb{R}_+$ belongs to class \mathcal{K} if it is continuous, strictly increasing and $\varphi(0) = 0$. A function $\varphi : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ belongs to class \mathcal{K}_∞ if $\varphi \in \mathcal{K}$ and $\lim_{s \rightarrow \infty} \varphi(s) = \infty$. A function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ belongs to class \mathcal{KL} if for each fixed $k \in \mathbb{R}_+$, $\beta(\cdot, k) \in \mathcal{K}$ and for each fixed $s \in \mathbb{R}_+$, $\beta(s, \cdot)$ is decreasing and $\lim_{k \rightarrow \infty} \beta(s, k) = 0$. Given a signal $v \in \mathbb{R}^{n_v}$, a starting time $k \geq 0$ and ending instant $j \geq k + 1$, we define

$$\bar{\mathbf{v}}(k, j) = \text{col}(v(k), \dots, v(k + j - 1)). \quad (1)$$

II. PRELIMINARIES & PROBLEM STATEMENT

We consider nonlinear MIMO systems with inputs $u \in \mathbb{R}^m$ and measured outputs $y \in \mathbb{R}^p$, whose dynamics can be represented with an *unknown* discrete-time state-space model:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)), \quad k \in \mathbb{N}, \\ y(k) &= h(x(k)), \end{aligned} \quad (2)$$

$x \in \mathbb{X}$ is an unmeasured state, and the functions $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ are assumed to be *unknown*.

We further assume that the following properties hold.

Assumption 2.1: The system (2) is controllable and observable.

Assumption 2.2: The origin is a stabilizable equilibrium point for system (2).

Consider next a closed-loop counterpart of system (2), assuming that the input is a perturbed state feedback law $u(k) = \phi(x(k), e(k))$, i.e.,

$$\begin{aligned} x(k+1) &= f(x(k), \phi(x(k), e(k))), \quad k \in \mathbb{N}, \\ y(k) &= h(x(k)), \end{aligned} \quad (3)$$

where $e : \mathbb{N} \rightarrow \mathbb{R}^{n_e}$ is an unknown *inner* perturbation trajectory. We can define input-to-state stability for the closed-loop system in (3) as follows.

Definition 2.3: We call the system (3) input-to-state stable in \mathbb{X} for perturbations in \mathbb{E} , or shortly $ISS(\mathbb{X}, \mathbb{E})$, if there exists a \mathcal{K} -function β and a \mathcal{K} -function γ such that, for each $x(0) \in \mathbb{X}$ and all $\mathbf{e} = \{e(l)\}_{l \in \mathbb{Z}_+}$, it holds that the corresponding state trajectory of (3) satisfies

$$\|x(k)\| \leq \beta(\|x(0)\|, k) + \gamma(\|\mathbf{e}_{[0,k]}\|), \quad \forall k \in \mathbb{Z}_{\geq 1}.$$

In this work, we exploit the fact that we can approximate a nonlinear system of the form (2) by a higher dimensional linear model utilizing the Koopman framework [16], where the state $x(k)$ of the original system (2) is lifted to a higher dimensional space via a set of observables, i.e.,

$$z(k) := \varphi(x(k)) := \text{col}(\varphi_1(x(k)), \dots, \varphi_L(x(k))),$$

and $z \in \mathbb{R}^L$ with $L > n$. A commonly followed approach is to use an approximate Koopman model for MPC [1] which is the linear-in-control input embedding, i.e.,

$$\begin{aligned} z(k+1) &= Az(k) + Bu(k), \quad k \in \mathbb{N}, \\ z(0) &= \varphi(\mathbf{x}_{\text{ini}}(k)), \end{aligned} \quad (4)$$

where we use an input-output data embedding of the state as in [2]. This model is preferred even though it is incomplete, because it leads to a QP problem to be solved online.

Remark 2.4: In [3] it was shown that for general discrete-time nonlinear systems of the form (2) in combination with a finite set of continuously differentiable observable functions $\varphi \in \mathbb{X} \rightarrow \mathbb{R}^L$ where \mathbb{X} is convex such that $\varphi(f(\cdot, 0)) \in \text{span}\{\varphi\}$, then there exists an exact finite dimensional lifting

$$\varphi(x(k+1)) = \mathcal{A}\varphi(x(k)) + \mathcal{B}(x(k), u(k))u(k). \quad (5)$$

For alternative complete nonlinear Koopman embedding methods we refer to [17], [18]. It would be of interest to research further usage of such embedding methods in Koopman MPC.

Remark 2.5: Even if [3] proves that linear-in-control input Koopman models as (4) are incomplete, hence leading to considerable prediction errors, for relatively short prediction horizons N the system remains relatively close to the initial conditions $z(0)$. In such a case it can be assumed that the transient response dominates the forced response. This further supports the validity of using (4) for predictive control strategies.

Above in (4), $\mathbf{x}_{\text{ini}}(k) := \text{col}(\mathbf{u}_{\text{ini}}(k), \mathbf{y}_{\text{ini}}(k))$ is a collection of T_{ini} past inputs and outputs defined as

$$\begin{aligned}\mathbf{u}_{\text{ini}}(k) &:= \text{col}(u(k - T_{\text{ini}} + 1), \dots, u(k - 1)) \in \mathbb{R}^{T_{\text{ini}}m}, \\ \mathbf{y}_{\text{ini}}(k) &:= \text{col}(y(k - T_{\text{ini}} + 1), \dots, y(k)) \in \mathbb{R}^{T_{\text{ini}}p},\end{aligned}$$

while the observables satisfy the following assumption.

Assumption 2.6: The functions $\varphi(\mathbf{x}_{\text{ini}}(k))$ parameterizing the vectors of observables are continuous and are such that $\varphi(0) = 0$.

The Koopman model (4) can be used to define N -step ahead prediction matrices:

$$\Psi(A) := \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \Gamma(A, B) := \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}, \quad (6)$$

ultimately allowing one to follow the *indirect* data-driven predictive control, see e.g., [19], where typically multi-step predictors of the NARX type [20] are identified from the input-output data generated by system (2) and used to predict $\mathbf{y}_{[1,N]}(k)$ and $\mathbf{z}_{[1,N]}(k)$ form past inputs and outputs $\mathbf{u}_{\text{ini}}(k)$, $\mathbf{y}_{\text{ini}}(k)$ and $\mathbf{u}_{[0,N-1]}(k)$, with,

$$\begin{aligned}\mathbf{u}_{[0,N-1]}(k) &:= \text{col}(u_{0|k}, \dots, u_{N-1|k}), \\ \mathbf{y}_{[1,N]}(k) &:= \text{col}(y_{1|k}, \dots, y_{N|k}), \\ \mathbf{z}_{[1,N]}(k) &:= \text{col}(z_{1|k}, \dots, z_{N|k}).\end{aligned}$$

A. Problem Statement

Assume that we are given a set of noiseless input-output data $\{y(i), u(i)\}_{i \in [0,s]}$ gathered from the unknown system (2), where $s \in \mathbb{N}$ denotes the number of samples. In this context, our goal is the following.

Problem 2.7: Given a finite set of noiseless input-output data $\{y(i), u(i)\}_{i \in [0,s]}$ the objective is to utilize them *exclusively* to construct an optimal linear-in-control input multi-step prediction model of the form:

$$\mathbf{z}_{[1,N]}(k) = \Psi \mathbf{z}_{0|k} + \Gamma \mathbf{u}_{[0,N-1]}(k), \quad (7)$$

minimizing the prediction error

$$e(k) := z_{1|k-1} - \varphi(\mathbf{x}_{\text{ini}}(k)), \quad \forall k \in \mathbb{N}. \quad (8)$$

□

We approach this problem by developing a multi-step predictor that minimizes the prediction error over the entire prediction horizon, to ensure prediction errors do not propagate. Due to the availability of only input and output data, we adopt a two-step approach to learn a state space model in the Koopman lifted space, elaborated upon in Section III.

Since the linear-in-control input Koopman form is incomplete, using it for control purposes further leads to the problem formalized in the following statement.

Problem 2.8: Given a nonlinear system of the form (2) and a non-exact multi-step Koopman predictor (7) obtained by solving Problem 2.7, develop a data-driven predictive control scheme with recursive feasibility and robust stability guarantees. □

We approach this problem using an interpolated initial condition [14] in the data-driven predictive control problem,

as elaborated in Section III. This approach allows us to guarantee recursive feasibility, but it makes guaranteeing stability more challenging. To circumvent this issue, we develop robust stability (ISS) conditions (see Section IV).

III. MULTI-STEP LEARNING OF THE KOOPMAN OBSERVABLES AND PREDICTION MATRICES

Toward framing our approach to tackle Problem 2.7, let us initially define the following data Hankel matrices:

$$\begin{aligned}\mathbf{U}_p &:= [\bar{\mathbf{u}}(0, T_{\text{ini}} - 1), \dots, \bar{\mathbf{u}}(T, T_{\text{ini}} - 1)], \\ \mathbf{Y}_p &:= [\bar{\mathbf{y}}(0, T_{\text{ini}}), \dots, \bar{\mathbf{y}}(T, T_{\text{ini}})], \\ \mathbf{Z}_p &:= [\bar{\mathbf{z}}(0, 1), \dots, \bar{\mathbf{z}}(T, 1)], \\ \mathbf{U}_f &:= [\bar{\mathbf{u}}(T_{\text{ini}} - 1, N), \dots, \bar{\mathbf{u}}(T_{\text{ini}} - 1 + T, N)], \\ \mathbf{Y}_f &:= [\bar{\mathbf{y}}(T_{\text{ini}}, N), \dots, \bar{\mathbf{y}}(T_{\text{ini}} + T, N)], \\ \mathbf{Z}_f &:= [\bar{\mathbf{z}}(1, N), \dots, \bar{\mathbf{z}}(T + 1, N)],\end{aligned} \quad (9)$$

where $T \geq (m+p)T_{\text{ini}} + mN$ is the number of their columns.

Since it is not possible to capture the global dynamics of nonlinear systems with multiple equilibria using a single linear operator [5], [6], we tackle Problem 2.7 under Assumption 2.1 and by leveraging Assumption 2.2 to fix the origin as the equilibrium point of system (2).

To identify directly a multi-step output predictor from the available input-output data, we define

$$\tilde{\Psi} := \tilde{C}\Psi, \quad \tilde{\Gamma} := \tilde{C}\Gamma, \quad (10)$$

with the block diagonal matrix $\tilde{C} := \text{blkdiag}(N, C)$. Accordingly, our goal of minimizing the prediction error translates into the following loss function:

$$J(\mathbf{Y}_f, \hat{\mathbf{Y}}_f) := \|\mathbf{Y}_f - \hat{\mathbf{Y}}_f\|_2^2,$$

where $\hat{\mathbf{Y}}_f$ is the Hankel matrix of future outputs generated by the linear-in-control multi-step Koopman model predictor. Hence, the multi-step learning problem we aim to solve can be formalized as:

$$\begin{aligned}\min_{\{\varphi, \tilde{\Psi}, \tilde{\Gamma}\}} & J(\mathbf{Y}_f, \hat{\mathbf{Y}}_f) \\ \text{subject to:} & \end{aligned} \quad (11)$$

$$\hat{\mathbf{Y}}_f = \tilde{\Psi} \varphi \left(\begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \end{bmatrix} \right) + \tilde{\Gamma} \mathbf{U}_f.$$

The observables φ_i can be chosen as some basis functions (e.g., radial basis functions) or neural networks [21] and, depending on this structural choice, (11) can be a nonlinear least squares problem. As an example, this is what happens when parametrizing the observable as radial basis function and learning their centers, or when the observables are parameterized as neural networks with nonlinear activation functions (such as tanh or sigmoid).

Solving (11) results in a multi-step predictor of the form

$$\begin{aligned}\mathbf{y}_{[1,N]} &= \tilde{\Psi}^* z(k) + \tilde{\Gamma}^* \mathbf{u}_{[0,N-1]} \\ &= \tilde{\Psi}^* \varphi^*(\mathbf{x}_{\text{ini}}(k)) + \tilde{\Gamma}^* \mathbf{u}_{[0,N-1]},\end{aligned}$$

where we have introduced the definition of the state at time k according to (4).

After this first identification step, we propose to refine the estimates of $\tilde{\Psi}$ and $\tilde{\Gamma}$ by exploiting the same arguments of linear subspace identification. Specifically, we feed the input-output data through the learned observables φ^* , construct the Koopman lifted state Hankel matrices \mathbf{Z}_p and \mathbf{Z}_f from (9) and, then, compute the least-squares estimates

$$\mathbf{Z}_f \begin{bmatrix} \mathbf{Z}_p \\ \mathbf{U}_f \end{bmatrix}^\dagger = [\Psi^{LS} \quad \Gamma^{LS}]. \quad (12)$$

Note that, this problem has a unique solution if the matrix $[\mathbf{Z}_p^\top \quad \mathbf{U}_f^\top]^\top$ has full row rank. This can be achieved by choosing a persistently exciting input of sufficient order and choosing the experiment length T sufficiently long. The resulting multi-step state predictor is given by

$$\mathbf{z}_{[1,N]}(k) = \Psi^{LS} z_{0|k} + \Gamma^{LS} \mathbf{u}_{[0,N-1]}(k),$$

being a multi-step linear-in-control input Koopman prediction model for the system (2). From Ψ^{LS} and Γ^{LS} , we can further extract their first row blocks to construct the matrices \tilde{A} and \tilde{B} such that

$$z_{1|k} = \tilde{A}z_{0|k} + \tilde{B}u_0(k),$$

which we use in constructing the terminal ingredients of the proposed predictive control scheme.

IV. KOOPMAN DATA-DRIVEN PREDICTIVE CONTROL WITH INTERPOLATED INITIAL STATE

As shown in [3], linear-in-control input Koopman models frequently lead to plant-model mismatches. When employing the Koopman model in a predictive control scheme, this mismatch could cause a loss of recursive feasibility and stability guarantees even if a terminal set is used in the Koopman space. Inspired by [14], we address this challenge by proposing the following Koopman data-driven predictive control formulation with interpolated initial state:

$$\min_{\Xi(k)} l_N(z_{N|k}) + \sum_{i=0}^{N-1} l(z_{i|k}, u_{i|k}) + l_r(z_{0|k}) \quad (13a)$$

subject to:

$$\mathbf{z}_{[1,N]}(k) = \Psi^{LS} z_{0|k} + \Gamma^{LS} \mathbf{u}_{[0,N-1]}(k), \quad (13b)$$

$$z_{0|k} = (1 - \xi_k)\varphi(\mathbf{x}_{\text{ini}}(k)) + \xi_k z_{1|k-1}, \quad (13c)$$

$$(\mathbf{z}_{[1,N]}(k), \mathbf{u}_{[0,N-1]}(k)) \in \mathbb{X}_z^N \times \mathbb{U}^N, \quad (13d)$$

$$z_{N|k} \in \mathbb{X}_T \subseteq \mathbb{X}_z, \quad (13e)$$

$$\xi_k \in [0, 1]. \quad (13f)$$

where $\Xi(k) := \text{col}(\mathbf{z}_{[1,N]}(k), \mathbf{u}_{[0,N-1]}(k), z_{0|k}, \xi_k)$, $z_{1|k-1}$ is the one-step ahead optimal predicted state from the previous sampling instance, the terminal cost is $l_N(z_{N|k}) := z_{N|k}^\top P z_{N|k}$, and the stage cost is $l(z_{i|k}, u_{i|k}) := z_{i|k}^\top Q z_{i|k} + u_{i|k}^\top R u_{i|k}$. With respect to [14], we propose the alternative regularization term

$$l_r(z_{0|k}) := \lambda \|z_{0|k} - \varphi(\mathbf{x}_{\text{ini}}(k))\|_2^2, \quad \lambda > 0, \quad (14)$$

which directly penalizes the deviation of the initial state in the Koopman space from the Koopman state obtained by

lifting the most recent measured input-output data $\mathbf{x}_{\text{ini}}(k)$. This alternative regularization is instrumental in establishing input-to-state stability guarantees, see, e.g., [15], [22], with respect to the Koopman model error

$$e(k) := z_{1|k-1}^* - \varphi(\mathbf{x}_{\text{ini}}(k)), \quad (15)$$

namely, the difference between the one-step ahead predicted Koopman state and the corresponding measured Koopman state obtained at time k . This prediction error is related to the inner perturbation trajectory in (3) which will later be utilized for stability analysis. Note that in (13) the Koopman predicted states are constrained to a compact set \mathbb{X}_z within the Koopman space that contains the origin in its interior. Such a set could be constructed by computing the minimum and the maximum of $\varphi(\mathbf{x}_{\text{ini}})$ over $\mathbf{x}_{\text{ini}} \in \mathbb{U}^{T_{\text{ini}}} \times \mathbb{Y}^{T_{\text{ini}}}$. Moreover, it is worth remarking that our Assumption 2.6 is consistent with the fact that we assume the origin is an equilibrium of the original nonlinear system.

A. Properties of proposed scheme

To establish the properties of (13) we have to introduce a terminal cost and set in the Koopman space, which are outlined in the following.

Assumption 4.1: There exists a locally stabilizing Koopman state-feedback control law $u(z(k)) := Kz(k)$, a positive definite matrix P and a compact set \mathbb{X}_T with the origin in its interior such that:

$$(\tilde{A} + \tilde{B}K)^\top P(\tilde{A} + \tilde{B}K) - P + Q + K^\top R K \succeq 0 \quad (16a)$$

$$(\tilde{A} + \tilde{B}K)\mathbb{X}_T \subseteq \mathbb{X}_T, \quad K\mathbb{X}_T \subseteq \mathbb{U}, \quad \mathbb{X}_T \subseteq \mathbb{X}_z. \quad (16b)$$

With this assumption in place, we can formalize our first theoretical result, namely the recursive feasibility of (13).

Theorem 4.2: Suppose that Assumption 4.1 holds. At any time $k \in \mathbb{N}$, given $\mathbf{x}_{\text{ini}}(k)$ and $z_{1|k-1}^*$, suppose that (13) is feasible. Then the problem in (13) is feasible at time $k+1$ for $\mathbf{x}_{\text{ini}}(k+1)$ and $z_{1|k}^*$.

Proof: Consider the optimal predicted input sequence $\mathbf{u}_{[0,N-1]}^*(k)$ at k , i.e.,

$$\mathbf{u}_{[0,N-1]}^*(k) = \{u_{0|k}^*, u_{1|k}^*, \dots, u_{N-2|k}^*, u_{N-1|k}^*\}.$$

Then, at time $k+1$, choose $\tilde{\xi}_{k+1} = 1$, which yields $\tilde{z}_{0|k+1} = z_{1|k}^*$ and construct the sub-optimal input sequence:

$$\tilde{\mathbf{u}}_{[0,N-1]}(k+1) = \{u_{1|k}^*, u_{2|k}^*, \dots, u_{N-1|k}^*, \tilde{u}_{N-1|k+1}\},$$

where $\tilde{u}_{N-1|k+1} := Kx_{N|k}^*$. This results in a sub-optimal, but feasible initial condition $\tilde{z}_{0|k+1}$, as the shifted sub-optimal sequence $\tilde{\mathbf{u}}_{[0,N-1]}(k+1)$ satisfies all the constraints in (13), due to Assumption 4.1 and the fact that

$$\begin{aligned} \tilde{z}_{N|k+1} &= \tilde{A}^N z_{1|k}^* + [\tilde{A}^{N-1} \tilde{B} \quad \dots \quad \tilde{B}] \tilde{\mathbf{u}}_{[0,N-1]}(\tilde{z}_{0|k+1}) \\ &= \tilde{A}(\tilde{A}^{N-1} z_{1|k}^* + \tilde{A}^{N-2} \tilde{B} u_{1|k}^* + \dots) + \tilde{B} \tilde{u}_{N-1|k+1} \\ &= \tilde{A} z_{N|k}^* + \tilde{B} \tilde{u}_{N-1|k+1} = (\tilde{A} + \tilde{B}K) z_{N|k}^* \in \mathbb{X}_T, \end{aligned}$$

which completes the proof. \blacksquare

Remark 4.3: Theorem 4.2 also holds in the presence of noise under suitable assumptions. If Problem 13 is feasible

at time k it is also feasible at time $k+1$ since $\xi = 1$ always results in a feasible problem.

Next, we prove that the nonlinear system (2) controlled in closed-loop with the Koopman DPC control law obtained by solving (13) is ISS with respect to the Koopman model one-step ahead prediction error $e(k)$. To this end, let $\mathbb{X}_f(N)$ denote the set of feasible initial states in the Koopman space, i.e., $z_{0|k}$, which is positively invariant by Theorem 4.2 for the closed-loop system (2)-(13). Moreover, we define the optimal value function at time k corresponding to (13), i.e.,

$$\begin{aligned} V(z_{0|k}^*) &:= J(z_{[1,N]}^*(k), \mathbf{u}_{[0,N-1]}^*(k), z_{0|k}^*, \xi_k^*) \\ &= l_N(z_{N|k}^*) + \sum_{i=0}^{N-1} l(z_{i|k}^*, u_{i|k}^*) + l_r(z_{0|k}^*). \end{aligned}$$

With that, we can establish the following instrumental result along with the usual reasoning used to establish that the MPC value function is decreasing.

Lemma 4.4: Suppose that Assumption 4.1 holds. Then for all $\varphi(\mathbf{x}_{\text{ini}}(0)) \in \mathbb{X}_z$ and $z_{1|0}^* \in \mathbb{X}_z$ such that the corresponding $z_{0|0}^* \in \mathbb{X}_f(N)$, along the trajectories of the closed-loop system (2)-(13) it holds that:

$$V(z_{0|k+1}^*) - V(z_{0|k}^*) \leq -\alpha_Q(\|z_{0|k}^*\|) + \sigma_r(\|e(k+1)\|), \quad (17)$$

for all $k \in \mathbb{N}$ and for $\alpha_Q(s) := \lambda_{\min}(Q)s^2 \in \mathcal{K}_\infty$ and $\sigma_r(s) := \lambda s^2 \in \mathcal{K}_\infty$.

Proof: Since $z_{0|0}^* \in \mathbb{X}_f(N)$, by Theorem 4.2, (13) remains feasible for all $k \geq 1$. At any time $k+1 \in \mathbb{N}$ consider the sub-optimal initial state $\tilde{z}_{0|k+1} = z_{1|k}^*$ obtained for $\tilde{\xi}_{k+1} = 1$ and the corresponding sub-optimal state and input sequences $\tilde{\mathbf{z}}_{[1,N]}(k+1), \tilde{\mathbf{u}}_{[0,N-1]}(k+1)$. Then, by the optimality of $V(z_{0|k+1}^*)$ it holds that

$$\begin{aligned} &V(z_{0|k+1}^*) - V(z_{0|k}^*) \\ &\leq J(\tilde{\mathbf{z}}_{[1,N]}(k+1), \tilde{\mathbf{u}}_{[0,N-1]}(k+1), z_{1|k}^*, 1) - V(z_{0|k}^*) \\ &= z_{N|k}^{*\top} ((\tilde{A} + \tilde{B}K)^\top P (\tilde{A} + \tilde{B}K) - P + Q + K^\top R K) z_{N|k}^* \\ &\quad - l(z_{0|k}^*, u_{0|k}^*) + l_r(\tilde{z}_{0|k+1}) - l_r(z_{0|k}^*) \\ &\leq -z_{0|k}^{*\top} Q z_{0|k}^* + \lambda \|z_{1|k}^* - \varphi(\mathbf{x}_{\text{ini}}(k+1))\|^2, \end{aligned}$$

which yields the inequality in (17). \blacksquare

Typically, robust stability results for MPC, see, e.g., [22], use continuity of the value function V evaluated at the true nonlinear system state $\varphi(\mathbf{x}_{\text{ini}}(k))$ to obtain input-to-state stability results. However, in the case of (13) this is not possible because $\mathbf{x}_{\text{ini}}(k)$ may not belong to the feasible set $\mathbb{X}_f(N)$. Hence, instead, we firstly establish ISS [15] for the interpolated state trajectory $z_{0|k}^*$ and then we use the properties of comparison functions [23] to establish ISS of the lifted Koopman state based on the true nonlinear system measurements, i.e., $\varphi(\mathbf{x}_{\text{ini}}(k))$.

Theorem 4.5: Assume that there exists $\alpha_2 \in \mathcal{K}_\infty$ such that $V(z) \leq \alpha_2(\|z\|)$ for all $z \in \mathbb{X}_f(N)$. Suppose that Assumption 4.1 holds. Then for all $\varphi(\mathbf{x}_{\text{ini}}(0)) \in \mathbb{X}_z$ and $z_{1|0}^* \in \mathbb{X}_z$ such that the corresponding $z_{0|0}^* \in \mathbb{X}_f(N)$,

along the trajectories of the closed-loop system (2)-(13) it holds that

$$\|\varphi(\mathbf{x}_{\text{ini}}(k))\| \leq \beta(\|\varphi(\mathbf{x}_{\text{ini}}(0))\|, k) + \gamma(\|\mathbf{e}_{[0,k]}\|), \quad \forall k \geq 1, \quad (18)$$

for some $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}$.

Proof: First we show that for all $z_{0|0}^* \in \mathbb{X}_f(N)$ there exist $\beta_z \in \mathcal{KL}$ and $\gamma_z \in \mathcal{K}$ such that

$$\|z_{0|k}^*\| \leq \beta_z(\|z_{0|0}^*\|, k) + \gamma_z(\|\mathbf{e}_{[0,k]}\|), \quad \forall k \geq 1. \quad (19)$$

Indeed, by the hypothesis, inequality (17) and the fact that $V(z_{0|k}^*) \geq \alpha_Q(\|z_{0|k}^*\|)$ for all $k \in \mathbb{N}$, by standard ISS Lyapunov function arguments [15] we obtain (19). Next notice that by (13c) it holds that

$$z_{0|k}^* - \varphi(\mathbf{x}_{\text{ini}}(k)) = \xi_k^*(z_{1|k-1}^* - \varphi(\mathbf{x}_{\text{ini}}(k))) = \xi_k^* e(k). \quad (20)$$

Hence, it holds that

$$\begin{aligned} \|\varphi(\mathbf{x}_{\text{ini}}(k))\| &= \|\varphi(\mathbf{x}_{\text{ini}}(k)) - z_{0|k}^* + z_{0|k}^*\| \\ &\leq \|z_{0|k}^*\| + \|\varphi(\mathbf{x}_{\text{ini}}(k)) - z_{0|k}^*\| \\ &= \|z_{0|k}^*\| + \|z_{0|k}^* - \varphi(\mathbf{x}_{\text{ini}}(k))\| \\ &\leq \|z_{0|k}^*\| + \|e(k)\| \\ &\leq \beta_z(\|z_{0|0}^*\|, k) + \gamma_z(\|\mathbf{e}_{[0,k]}\|) + \|e(k)\|. \end{aligned}$$

Then due to $\beta_z \in \mathcal{KL}$ it follows that:

$$\begin{aligned} \beta_z(\|z_{0|0}^*\|, k) &= \beta_z(\|z_{0|0}^* + \varphi(\mathbf{x}_{\text{ini}}(0)) - \varphi(\mathbf{x}_{\text{ini}}(0))\|, k) \\ &= \beta_z(\|\varphi(\mathbf{x}_{\text{ini}}(0)) + \xi_0^* e(0)\|, k) \\ &\leq \beta_z(\|\varphi(\mathbf{x}_{\text{ini}}(0))\| + \|e(0)\|, k) \\ &\leq \beta_z(2\|\varphi(\mathbf{x}_{\text{ini}}(0))\|, k) + \beta_z(2\|e(0)\|, k) \\ &\leq \beta_z(2\|\varphi(\mathbf{x}_{\text{ini}}(0))\|, k) + \beta_z(2\|e(0)\|, 0). \end{aligned}$$

Above we used the fact that $\xi_k^* \in [0, 1]$, the triangle inequality and standard properties of \mathcal{K} and \mathcal{L} functions. By combining the previous inequalities and using $\|e(i)\| \leq \|\mathbf{e}_{[0,k]}\|$ for any $i \in [0, k]$, we obtain (18) with $\beta(s, k) := \beta_z(2s, k) \in \mathcal{KL}$ and $\gamma(s) := \gamma_z(s) + s + \beta_z(2s, 0) \in \mathcal{K}$, which completes the proof. \blacksquare

Remark 4.6: The ISS bound (18) holds in the presence of noisy data under certain assumptions, since ISS does not assume that the prediction error is bounded.

According to standard ISS arguments [15], the ISS bound (18) implies that when the Koopman model error tends to zero, the lifted measured state of the true nonlinear system converges to zero. Due to Assumption 2.6 we further obtain that $\|\mathbf{x}_{\text{ini}}(k)\|$ converges to zero and, under detectability, that the true state of the nonlinear system converges to zero.

Furthermore, the ISS bound on trajectories (18) implies the finite asymptotic gain property [15], which guarantees that all trajectories of the nonlinear system (13) in closed-loop with the Koopman DPC control law obtained by solving (13) are ultimately bounded in a set proportional with the Koopman model error.

Remark 4.7 (On the choice of $l_r(z_{0|k})$): Given the equality in (20), the regularization loss in (14) can be replaced with

$$l_r(z_{0|k}) := \lambda \xi_k^2 \|e(k)\|_2^2, \quad \lambda > 0, \quad (21)$$

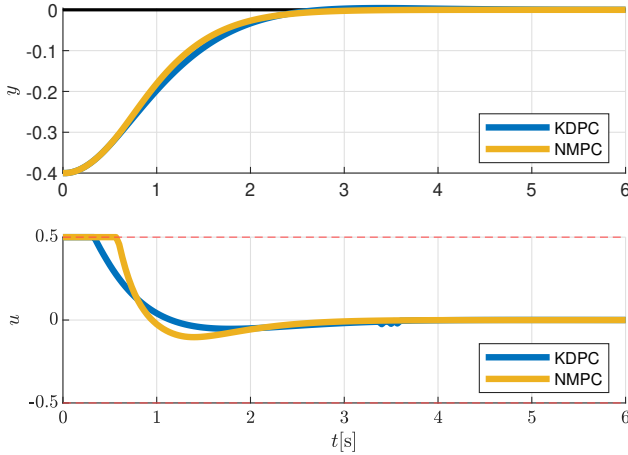


Fig. 1. Simulations chart: position x_1 , control input u , interpolation variable ξ and observables φ_i .

which directly pushes ξ_k to zero when $e(k) \neq 0$. Notice that if we were to use the original regularization cost proposed in [14], i.e., $l_r(z_{0|k}) := \lambda \xi_k^2$, the inequality (17) holds with $\sigma_r(\|e(k+1)\|)$ replaced by λ . Since λ needs to be chosen sufficiently large, this only yields bounded trajectories in a very large set, which is not very useful. Also, this choice does not yield ISS for the interpolated state trajectory, but only input-to-state practical stability.

V. ILLUSTRATIVE EXAMPLE

To assess the effectiveness of the proposed scheme, we consider two nonlinear benchmark examples. The first example is a 2D cart-spring-damper (CSD) from [24] and the second example is a nonlinear pendulum from [25]. We analyze the accuracy of the learned observables by computing the R^2 coefficient for all the future prediction steps over the prediction horizon $j = 1, \dots, N$, i.e.,

$$R^2 = 1 - \frac{\sum_{k=1}^{N_{test}} [y(k+j) - y_{j|k}]^2}{\sum_{k=1}^{N_{test}} [y(k+j) - \bar{y}_j]^2}, \quad (22)$$

where N_{test} is the length of the test data set and $\bar{y}_j = \frac{1}{N_{test}} \sum_{k=1}^{N_{test}} y(k+j)$. The training data set and test data set are shown in Appendix II. The proposed Koopman data-driven predictive control (KDPC) is compared with nonlinear model predictive control (NMPC). For NMPC, knowledge of the plants and state measurement is assumed while only input and output data is used for KDPC. For NMPC we use the input-state cost function $l(x_{i|k}, u_{i|k}) := x_{i|k}^\top Q x_{i|k} + u_{i|k}^\top R u_{i|k}$ with terminal cost $l_N(x_{N|k}) := x_{N|k}^\top P x_{N|k}$ and the exact nonlinear model of the plant is used as prediction model and the NMPC problem is solved using FMINCON. More implementation details for Koopman data-driven predictive are discussed in Appendix I. The code can be found in the github repository¹

a) *2D cart-spring-damper*: To assess the effectiveness of the proposed scheme, we consider the discretized counterpart of the 2D cart-spring-damper (CSD) from [24], i.e.,

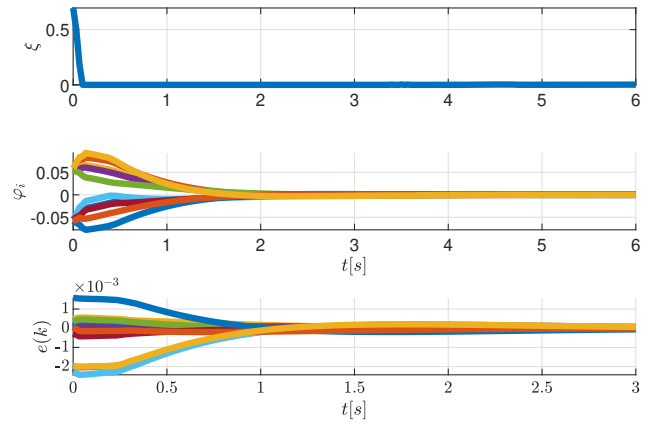


Fig. 2. Simulations chart: interpolation variable ξ , observables φ_i and Koopman model error e .

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k), \\ x_2(k+1) &= x_2(k) - \frac{T_s k_0}{M} e^{-x_1(k)} x_1(k) - \frac{T_s \bar{h}_d}{M} x_2(k) + \frac{T_s}{M} u(k), \\ y(k) &= x_1(k), \end{aligned} \quad (23)$$

considering the same parameters from [24]. An open-loop identification experiment was performed to generate the output data using a multi-sine input constructed with the Matlab function `idinput`, with the parameters Range $[-4, 4]$, Band $[0, 1]$, Period 1000, NumPeriod 1 and Sine $[25, 40, 1]$. The observables are parameterized by means of a neural network with 8 tanh activation functions and 2 hidden layers and $T_{ini} = 5$. The parameters are found by solving (11) using the Adam optimizer with the MSE loss function from the Pytorch library with a learning rate of $1 \cdot 10^{-2}$. The resulting R^2 coefficient is computed, see Table I. From this table it seems that the KDPC prediction model achieves accurate predictions on the test data with R^2 scores close to 1. Once the observables are learned the state data matrices \mathbf{Z}_p and \mathbf{Z}_f are found by feeding the same input-output data through the learned observables. The prediction model matrices Ψ^{LS} and Γ^{LS} are then retrieved as in (12). From them, we then extract the Koopman model matrices \tilde{A} and \tilde{B} as discussed at the end of Section III to construct the terminal set by solving (16). This can be done using standard techniques for linear MPC, i.e., first solve a corresponding linear matrix inequality for (\tilde{A}, \tilde{B}) that yields a solution (P, K) to (16a) and then compute a polyhedral constraints admissible and invariant terminal set for $(\tilde{A} + \tilde{B}K)$ that satisfies (16b) (e.g., using the MPT 3 Matlab Toolbox). The state of the system (23) is then initialized close to the constraints and stabilized using the Koopman DPC scheme in (13) with the tuning parameters reported in Table II. More details about the implementation for the terminal cost and constraints are given in Appendix I.

We initialize the system at $x_1(0) = -0.4\text{m}$ and $x_2(0) = 0\text{m/s}$. The attained results are shown in Fig.2. It is clear that the Koopman DPC controller remains recursively feasible and stabilizes the true nonlinear system while input constraints are active. Moreover, the interpolation variable is pushed to zero as soon as the input constraint becomes inactive, which implies that when this condition holds the

¹https://github.com/todejong/KDPC_RSFRG

TABLE I
R² COEFFICIENTS FOR CSD.

N	1	3	5	7	9	11	13	15
KDPC	0.9844	0.9879	0.9902	0.9921	0.9936	0.9947	0.9953	0.9958

TABLE II
CONTROLLER PARAMETERS AND CONSTRAINTS CSD.

	Q	R	λ	u _{min}	u _{max}	N
KDPC	10I _L	1	10 ⁹	-0.5	0.5	15
NMPC	10I _n	1	-	-0.5	0.5	15

lifted measured input-output data $x_{ini}(k)$ are used and, hence, they are feasible. The results show that the observables converge to the origin, as expected when the prediction error vanishes. This is also observed in the third subplot in Figure 2. Here it is shown that the Koopman model error defined by (15) is of the order 10^{-3} and goes to zero after initialization. From this and the ISS property, we can conclude asymptotic stability of the Koopman state. All methods are able to stabilize the system while accounting for constraints. These simulations show that the KDPC controller achieves very similar performance compared with NMPC without any assumed knowledge of the plant.

b) *Pendulum*: To assess the effectiveness of the proposed scheme, we secondly consider the discretized nonlinear pendulum model from [25], i.e.,

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} 1 - \frac{bT_s}{J} & 0 \\ T_s & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} \frac{T_s}{J} \\ 0 \end{bmatrix} u(k) \\ &\quad - \begin{bmatrix} \frac{MLgT_s}{2J} \sin(x_2(k)) \\ 0 \end{bmatrix}, \end{aligned} \quad (24)$$

$$y(k) = x_2(k),$$

where $u(k)$ and $y(k)$ are the system input torque and pendulum angle at time instant k , while $J = \frac{ML^2}{3}$, $M = 1\text{kg}$ and $L = 1\text{m}$ are the moment of inertia, mass and length of the pendulum. Moreover, $g = 9.81\text{m/s}^2$ is the gravitational acceleration, $b = 0.1$ is the friction coefficient and the sampling time $T_s = 1/30\text{s}$. The output data is again generated by an open-loop identification experiment using a multi-sine input with the exact same parameters as for the 2D cart-spring-damper, see Appendix II for the test data and

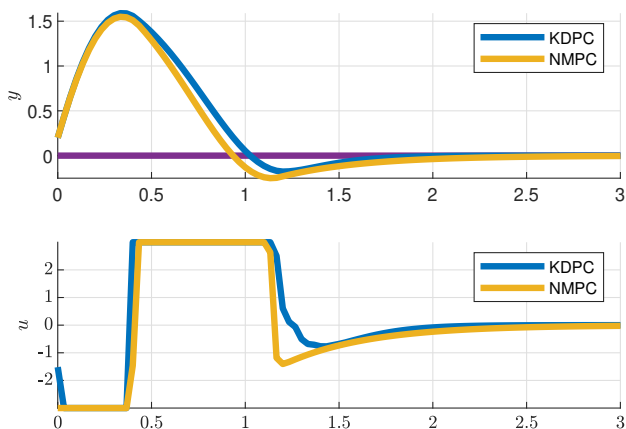


Fig. 3. Simulations pendulum: angle x_2 , control input u , interpolation variable ξ and observables φ_i .

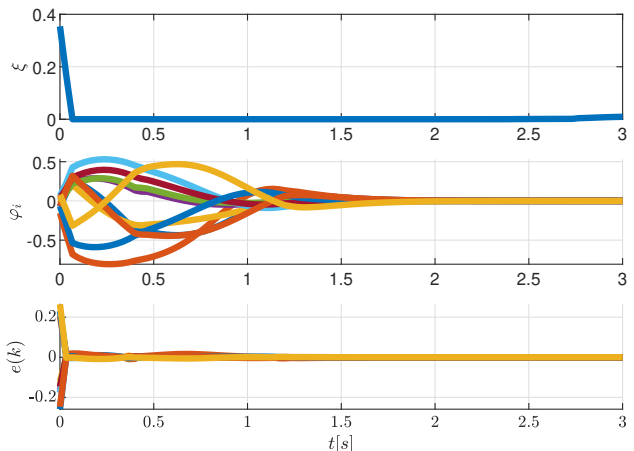


Fig. 4. Simulations pendulum: interpolation variable ξ , observables φ_i and Koopman model error e .

TABLE III

R² COEFFICIENTS FOR PENDULUM.

N	1	3	5	7	9	11	13	15
KDPC	0.9139	0.9109	0.9070	0.9016	0.8960	0.8899	0.8822	0.8717

TABLE IV

CONTROLLER PARAMETERS AND CONSTRAINTS PENDULUM.

	Q	R	λ	u _{min}	u _{max}	N
KDPC	10I _L	0.01	10 ⁹	-3	3	15
NMPC	10I _n	0.01	-	-3	3	15

training data. The observables are parameterized by means of a neural network with 2 hidden layers with 10 tanh activation functions per layer and $T_{ini} = 3$. More details about the implementations and the terminal cost and constraints are given in Appendix I.

We initialize the pendulum at $x_1(0) = 7\text{ rad/s}$ and $x_2(0) = 0.2\text{ rad}$. The attained results are shown in Fig.4. It is clear that all methods remains recursively feasible and stabilize the true nonlinear system while input constraints are active. The performance of the KDPC controller is again close to the one of NMPC. Just as for the other example, the interpolation variable is pushed to zero as soon as the input constraint becomes inactive and the observables converge to the origin. This is also observed in the third subplot in Fig.4. Here it is shown that the Koopman model error defined by (15) quickly goes to zero after initialization. From this and the ISS property, we can conclude asymptotic stability of the Koopman state.

VI. CONCLUSIONS AND FUTURE WORKS

We have introduced a novel Koopman DPC scheme aimed at stabilizing general nonlinear systems. Our methodology involves a two-step approach. Firstly, we focus on learning the observables. Subsequently, we develop a best-fit state space model within the lifted Koopman space, that simplifies to a least squares problem in the lifted space. While our use of the linear-in-control input Koopman formulation enables the control problem to be framed as a quadratic program, such formulation is still incomplete.

To address this limitation, we have devised an interpolated

initial condition strategy for the Koopman DPC problem, ensuring recursive feasibility even in the presence of plant-model mismatches. Furthermore, we established sufficient conditions under which Input-to-State Stability is guaranteed. Through a simulation example, we demonstrated the effectiveness of our Koopman DPC approach in stabilizing nonlinear systems, thereby showcasing its potential.

A. Future Works

In our paper, we assumed a noiseless environment for both learning the Koopman model and conducting robust stability analysis. While our results establish recursive feasibility and robust stability for non-exact prediction models, they have not been extensively tested in the presence of noise. However, the interpolated initial condition ensures recursive feasibility if initially feasible even in the case of noise. The ISS bound also remains valid because it does not require that the prediction error is bounded. Future work should focus on deriving stochastic robust stability guarantees tailored to specific noise assumptions.

The performance of our proposed control scheme is inherently linked to the accuracy of the prediction model. However, it has been shown that the linear-in-control input Koopman formulation utilized in this paper may introduce prediction errors [3]. To enhance the performance of our control approach, it is recommended to extend our methodology by utilizing the complete Koopman formulation [3]. By addressing this incompleteness, we can potentially improve the accuracy and robustness of our control scheme.

REFERENCES

- [1] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.
- [2] —, "Koopman model predictive control of nonlinear dynamical systems," *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*, pp. 235–255, 2020.
- [3] L. C. Jacob, R. Tóth, and M. Schoukens, "Koopman form of nonlinear systems with inputs," *Automatica*, vol. 162, p. 111525, 2024.
- [4] Y. Lan and I. Mezić, "Linearization in the large of nonlinear systems and koopman operator spectrum," *Physica D: Nonlinear Phenomena*, vol. 242, no. 1, pp. 42–53, 2013.
- [5] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern koopman theory for dynamical systems," *arXiv preprint arXiv:2102.12086*, 2021.
- [6] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PLoS one*, vol. 11, no. 2, p. e0150171, 2016.
- [7] X. Zhang, W. Pan, R. Scattolini, S. Yu, and X. Xu, "Robust tube-based model predictive control with koopman operators," *Automatica*, vol. 137, p. 110114, 2022.
- [8] Y. Lian and C. N. Jones, "Learning feature maps of the koopman operator: A subspace viewpoint," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 860–866.
- [9] A. Borghi, "Koopman subspace identification in the presence of measurement noise," 2021.
- [10] P. Verheijen, V. Breschi, and M. Lazar, "Handbook of linear data-driven predictive control: Theory, implementation and design," *Annual Reviews in Control*, vol. 56, p. 100914, 2023.
- [11] J. Köhler, K. P. Wabersich, J. Berberich, and M. N. Zeilinger, "State space models vs. multi-step predictors in predictive control: Are state space models complicating safe data-driven designs?" in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 491–498.
- [12] M. Lazar, "Basis functions nonlinear data-enabled predictive control: Consistent and computationally efficient formulations," *arXiv preprint arXiv:2311.05360*, 2023.
- [13] L. C. Jacob, G. I. Beintema, M. Schoukens, and R. Tóth, "Deep identification of nonlinear systems in koopman form," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 2288–2293.
- [14] J. Köhler and M. N. Zeilinger, "Recursively feasible stochastic predictive control using an interpolating initial state constraint," *IEEE Control Systems Letters*, vol. 6, pp. 2743–2748, 2022.
- [15] Z.-P. Jiang and Y. Wang, "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, vol. 37, pp. 857–869, 2001.
- [16] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.
- [17] D. Goswami and D. A. Paley, "Global bilinearization and controllability of control-affine nonlinear systems: A koopman spectral approach," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 6107–6112.
- [18] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing koopman theory to allow for inputs and control," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.
- [19] D. Masti, F. Smarra, A. D'Innocenzo, and A. Bemporad, "Learning affine predictors for mpc of nonlinear systems via artificial neural networks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5233–5238, 2020.
- [20] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.
- [21] Y. Lian, R. Wang, and C. N. Jones, "Koopman based data-driven predictive control," 2021.
- [22] M. Lazar, W. P. M. H. Heemels, and A. R. Teel, "Further input-to-state stability subtleties for discrete-time systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1609–1613, 2013.
- [23] C. M. Kellett, "A compendium of comparison function results," *Mathematics of Control, Signals, and Systems*, vol. 26, no. 6, pp. 339–374, 2014.
- [24] D. M. Raimondo, D. Limon, M. Lazar, L. Magni, and E. F. ndez Camacho, "Min-max model predictive control of nonlinear systems: A unifying overview on stability," *European Journal of Control*, vol. 15, no. 1, pp. 5–21, 2009.
- [25] A. Dalla Libera and G. Pillonetto, "Deep prediction networks," *Neurocomputing*, vol. 469, pp. 321–329, 2022.

APPENDIX I
COMPUTATION OF TERMINAL INGREDIENTS

The terminal cost P and terminal constraints in the examples are computed by firstly extracting \tilde{A} and \tilde{B} from Ψ^{LS} and Γ^{LS} in (12). Finding P amounts to solving inequality (16a). This is done by defining the variables $Y = KP^{-1}$ and $O = P^{-1}$, after which Schur complement is applied to obtain the LMI:

$$\begin{bmatrix} O & (\tilde{A}O + \tilde{B}Y)^\top & O & Y^\top \\ (\tilde{A}O + \tilde{B}Y) & O & 0 & 0 \\ O & 0 & Q^{-1} & 0 \\ Y & 0 & 0 & R^{-1} \end{bmatrix} \succ 0, \quad (25)$$

$O = O^\top, \quad O \succ 0.$

This LMI is solved by replacing $\succ 0$ by $\succeq I10^{-6}$ and using the MOSEK solver. The terminal cost is found by $P = O^{-1}$ and the terminal control matrix is given by $K = YO^{-1}$. Finding a terminal set $\mathbb{X}_T := \{z : M_N z \leq b_N\}$ amounts to computing a constraints admissible and invariant set for the dynamics $(\tilde{A} + \tilde{B}K)$. This is done by defining the lifted state constraint sets $\mathbb{X} := \{z : A_z z \leq b_z\}$, the input constraint set $\mathbb{U} := \{u : A_u u \leq b_u\}$ and the admissible constraint set $\mathbb{X}\mathbb{U} := \{z : A_u K z \leq b_u\} \cap \mathbb{X}$ using the Polyhedron function from the MPT 3 toolbox. The terminal constraint set \mathbb{X}_T for the dynamics $(\tilde{A} + \tilde{B}K)$ is found using the invariantSet function from the MPT 3 toolbox.

The following MATLAB commands are used to find M_N and b_N , i.e.,

- 1) model = LTISystem('A', $\tilde{A} + \tilde{B}K$)
- 2) Xset = Polyhedron(A_x, b_x)
- 3) Uset = Polyhedron(A_u, b_u)
- 4) XUset = Polyhedron($A_u K, b_u$) & Xset
- 5) InvSet = model.invariantSet('X', XUset)

APPENDIX II
IDENTIFICATION DATA

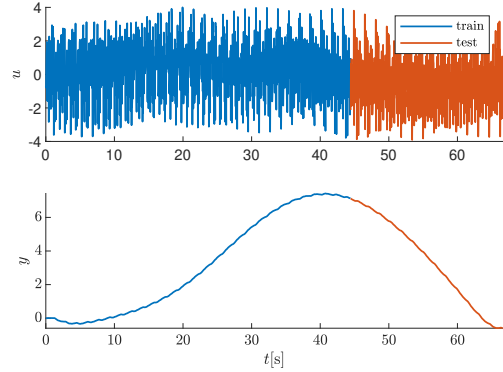


Fig. 5. Input-output data for 2D cart-spring damper.

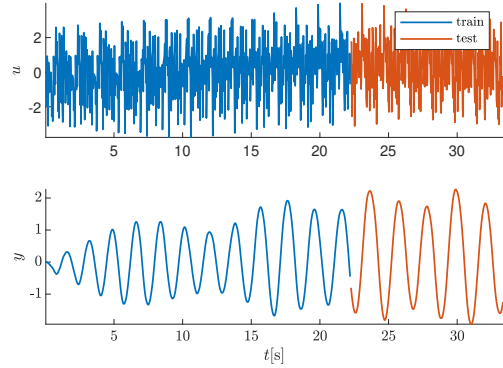


Fig. 6. Input-output data for Pendulum.