



Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control[☆]

Milan Korda^{*}, Igor Mezić

University of California, Santa Barbara, United States

ARTICLE INFO

Article history:

Received 17 March 2017

Received in revised form 5 September 2017

Accepted 26 January 2018

Keywords:

Koopman operator

Model predictive control

Data-driven control design

Optimal control

Lifting

ABSTRACT

This paper presents a class of linear predictors for nonlinear controlled dynamical systems. The basic idea is to lift (or embed) the nonlinear dynamics into a higher dimensional space where its evolution is approximately linear. In an uncontrolled setting, this procedure amounts to numerical approximations of the Koopman operator associated to the nonlinear dynamics. In this work, we extend the Koopman operator to controlled dynamical systems and apply the Extended Dynamic Mode Decomposition (EDMD) to compute a finite-dimensional approximation of the operator in such a way that this approximation has the form of a linear controlled dynamical system. In numerical examples, the linear predictors obtained in this way exhibit a performance superior to existing linear predictors such as those based on local linearization or the so called Carleman linearization. Importantly, the procedure to construct these linear predictors is completely data-driven and extremely simple – it boils down to a nonlinear transformation of the data (the lifting) and a linear least squares problem in the lifted space that can be readily solved for large data sets. These linear predictors can be readily used to design controllers for the nonlinear dynamical system using linear controller design methodologies. We focus in particular on model predictive control (MPC) and show that MPC controllers designed in this way enjoy computational complexity of the underlying optimization problem comparable to that of MPC for a linear dynamical system with the same number of control inputs and the same dimension of the state-space. Importantly, linear inequality constraints on the state and control inputs as well as nonlinear constraints on the state can be imposed in a linear fashion in the proposed MPC scheme. Similarly, cost functions nonlinear in the state variable can be handled in a linear fashion. We treat both the full-state measurement case and the input–output case, as well as systems with disturbances/noise. Numerical examples demonstrate the approach.¹

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

This paper presents a class of linear predictors for nonlinear controlled dynamical systems. By a predictor, we mean an artificial dynamical system that can predict the future state (or output) of a given nonlinear dynamical system based on the measurement of the current state (or output) and given the current and future inputs of the system. We focus on predictors possessing a *linear* structure that allows established linear control design methodologies to be used to design controllers for nonlinear dynamical systems.

The key step in obtaining accurate predictions of a nonlinear dynamical system as the output of a linear dynamical system is *lifting* of the state-space to a higher dimensional space, where the evolution of this lifted state is (approximately) linear. For uncontrolled dynamical systems, this idea can be rigorously justified using the Koopman operator theory (Mezić & Banaszuk, 2004; Mezić, 2005). The Koopman operator is a *linear* operator that governs the evolution of scalar functions (often referred to as observables) along trajectories of a given nonlinear dynamical system. A finite-dimensional approximation of this operator, acting on a given finite-dimensional subspace of all functions, can be viewed as a predictor of the evolution of the values of these functions along the trajectories of the nonlinear dynamical system and hence also as a predictor of the values of the state variables themselves provided that they lie in the subspace of functions the operator is truncated on. In the uncontrolled context, the idea of representing a nonlinear dynamical system by an infinite-dimensional linear operator goes back to the seminal works of Carleman (1932),

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Riccardo Scattolini under the direction of Editor Ian R. Petersen.

^{*} Corresponding author.

E-mail addresses: milan.korda@engineering.ucsb.edu (M. Korda), mezic@engineering.ucsb.edu (I. Mezić).

¹ The source code for the numerical examples is available from <https://github.com/MilanKorda/KoopmanMPC/raw/master/KoopmanMPC.zip>.

Koopman (1931) and Koopman and von Neuman (1932). The potential usefulness of such linear representations for prediction and control was suggested in Mezić and Banaszuk (2004).

In this work, we extend the definition of the Koopman operator to controlled dynamical systems by viewing the controlled dynamical system as an uncontrolled one evolving on an extended state-space given by the product of the original state-space and the space of all control sequences. Subsequently, we use a modified version of the Extended Dynamic Mode Decomposition (EDMD) (Williams, Kevrekidis, & Rowley, 2015) to compute a finite-dimensional approximation of this controlled Koopman operator. In particular, we impose a specific structure on the set of observables appearing in the EDMD such that the resulting approximation of the operator has the form of a linear controlled dynamical system. Importantly, the procedure to construct these linear predictors is completely data-driven (i.e., does not require the knowledge of the underlying dynamics) and extremely simple – it boils down to a nonlinear transformation of the data (the lifting) and a linear least squares problem in the lifted space that can readily be solved for large data sets using linear algebra. On the numerical examples tested, the linear predictors obtained in this way exhibit a predictive performance superior compared to both Carleman linearization as well as local linearization methods. For a related work on extending Koopman operator methods to controlled dynamical systems, see Brunton, Brunton, Proctor, and Kutz (2016), Proctor, Brunton, and Kutz (2016, in press), Williams, Hemati, Dawson, Kevrekidis, and Rowley (2016). See also Surana (2016), Surana and Banaszuk (2016) and Mauroy and Gonçalves (2016) for the use of Koopman operator methods for state estimation and nonlinear system identification, respectively.

Finally, we demonstrate in detail the use of these predictors for model predictive control (MPC) design; see the survey (Mayne, Rawlings, Rao, & Scokaert, 2000) or the book (Grüne & Pannek, 2011) for an overview of MPC. In particular, we show that these predictors can be used to design MPC controllers for *nonlinear* dynamical systems with computational complexity comparable to MPC controllers for *linear* dynamical systems with the same number of control inputs and states. Indeed, the resulting MPC scheme is extremely simple: In each time step of closed-loop operation it involves one evaluation of a family of nonlinear functions (the lifting) to obtain the initial condition of the linear predictor and the solution of a *convex* quadratic program affinely parametrized by this lifted initial condition. Importantly, nonlinear cost functions and constraints can be handled in a linear fashion by including all nonlinear terms appearing in these functions among the lifting functions. Therefore, the proposed scheme can be readily used for predictive control of nonlinear dynamical systems, using the tailored and extremely efficient solvers for linear MPC (in our case qpOASES Ferreau, Kirches, Potschka, Bock, & Diehl, 2014), thereby avoiding the troublesome and computationally expensive solution of nonconvex optimization problems encountered in classical nonlinear MPC schemes (Grüne & Pannek, 2011).

The paper is organized as follows: In Section 2 we describe the problem setup and the basic idea behind the use of linear predictors for nonlinear dynamical systems. In Section 3 we derive these linear predictors as finite-dimensional approximations to the Koopman operator extended to nonlinear dynamical systems. In Section 4 we describe a numerical algorithm for obtaining these linear predictors. In Section 5 we describe the use of these predictors for model predictive control. In Section 7 we discuss extensions of the approach to input–output systems (Section 7.1) and to systems with disturbances/noise (Section 7.2). In Section 8 we present numerical examples.

2. Linear predictors – basic idea

We consider a discrete-time nonlinear controlled dynamical system

$$x^+ = f(x, u), \quad (1)$$

where $x \in \mathbb{R}^n$ is the state of the system, $u \in \mathcal{U} \subset \mathbb{R}^m$ the control input, x^+ is the successor state and f the transition mapping. The input–output case is treated in Section 7.1.

The focus of this paper is the prediction of the trajectory of (1) given an initial condition x_0 and the control inputs $\{u_0, u_1, \dots\}$. In particular we are looking for simple predictors possessing a linear structure which are suitable for linear control design methodologies such as model predictive control (MPC) (Mayne et al., 2000).

The predictors investigated are assumed to be of the form of a controlled *linear* dynamical system

$$z^+ = Az + Bu, \quad (2)$$

$$\hat{x} = Cz,$$

where $z \in \mathbb{R}^N$ with (typically) $N \gg n$ and \hat{x} is the prediction of x , $B \in \mathbb{R}^{N \times m}$ and $C \in \mathbb{R}^{n \times N}$. The initial condition of the predictor (2) is given by

$$z_0 = \psi(x_0) := \begin{bmatrix} \psi_1(x_0) \\ \vdots \\ \psi_N(x_0) \end{bmatrix}, \quad (3)$$

where x_0 is the initial condition of (1) and $\psi_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, N$, are user-specified (typically nonlinear) lifting functions. The state z is referred to as the *lifted state* since it evolves on a higher-dimensional, lifted, space.² Importantly, the control input $u \in \mathcal{U}$ of (2) remains *unlifted* and hence linear constraints on the control inputs can be imposed in a linear fashion. Notice also that the predicted state \hat{x} is a *linear* function of the lifted state z and hence also linear constraints on the state can be readily imposed. Fig. 1 depicts this idea.

Predictors of this form lend themselves immediately to linear feedback control design methodologies. Importantly, however, the resulting feedback controller will be nonlinear in the original state x even though it may be (but is not required to be) linear in the lifted state z . Indeed, from any feedback controller $\kappa_{\text{lift}} : \mathbb{R}^N \rightarrow \mathbb{R}^m$ for (2), we obtain a feedback controller $\kappa : \mathbb{R}^n \rightarrow \mathbb{R}^m$ for the original system (1) defined by

$$\kappa(x) := \kappa_{\text{lift}}(\psi(x)). \quad (4)$$

The idea is that if the true trajectory of x generated by (1) and the predicted trajectory of \hat{x} generated by (2) are close for each admissible input sequence, then the optimal controller for (2) should be close to the optimal controller for (1). In Section 3 we will see how the linear predictors (2) can be derived within the Koopman operator framework extended to controlled dynamical systems.

Note, however, that in general one cannot hope that a trajectory of a linear system (2) will be an accurate prediction of the trajectory of a nonlinear system for all future times. Nevertheless, if the predictions are accurate on a long enough time interval, these predictors can facilitate the use of linear control systems

² In general, the term “lifted state” may be misleading as, in principle, the same approach can be applied to dynamical systems with states evolving on a (possibly infinite-dimensional) space \mathcal{M} , with finitely many observations (or outputs) $\mathbf{h}(x) = [h_1(x), \dots, h_p(x)]^\top$ available at each time instance; the lifting is then applied to these output measurements (and possibly their time-delayed versions), i.e., $\psi_i(x)$ in (19) becomes $\psi_i(\mathbf{h}(x))$ for some functions $\psi_i : \mathbb{R}^p \rightarrow \mathbb{R}$. In other words, rather than the state itself we lift the output of the dynamical system. For a detailed treatment of the input–output case, see Section 7.1.

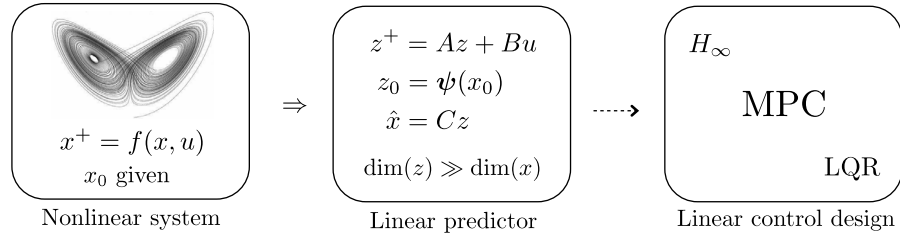


Fig. 1. Linear predictor for a nonlinear controlled dynamical system — z is the lifted state evolving on a higher-dimensional state space, \hat{x} is the prediction of the true state x and ψ is a nonlinear lifting mapping. This predictor can then be used for control design using linear methods, in our case linear MPC.

design methodologies. Especially suited for this purpose is model predictive control (MPC) that uses only *finite-time predictions* to generate the control input.

We will briefly mention in Section 3.2.2 how more complex, bi-linear, predictors of the form

$$\begin{aligned} z^+ &= Az + (Bz)u, \\ \hat{x} &= Cz \end{aligned} \quad (5)$$

can be obtained within the same framework and argue that predictors of this form can be asymptotically tight (in a well defined sense). Nevertheless, predictors of the form (5) are not immediately suited for linear control design and hence in this paper we focus on the linear predictors (2).

3. Koopman operator — rationale behind the approach

We start by recalling the Koopman operator approach for the analysis of an *uncontrolled* dynamical system

$$x^+ = f(x). \quad (6)$$

The Koopman operator $\mathcal{K} : \mathcal{F} \rightarrow \mathcal{F}$ is defined by

$$(\mathcal{K}\psi)(x) = \psi(f(x)) \quad (7)$$

for every $\psi : \mathbb{R}^n \rightarrow \mathbb{R}$ belonging to \mathcal{F} , which is a space of functions (often referred to as observables) invariant under the action of the Koopman operator. Importantly, the Koopman operator is *linear* (but typically infinite-dimensional) even if the underlying dynamical system is nonlinear. Crucially, this operator fully captures all properties of the underlying dynamical system provided that the space of observables \mathcal{F} contains the components of the state x_i , i.e., the mappings $x \mapsto x_i$ belong to \mathcal{F} for all $i \in \{1, \dots, n\}$. For a detailed survey on Koopman operator and its applications, see Budišić, Mohr, and Mezić (2012).

3.1. Koopman operator for controlled systems

There are several ways of generalizing the Koopman operator to controlled systems; see, e.g., Proctor et al. (in press) and Williams et al. (2016). In this paper we present a generalization that is both rigorous and practical. We define the Koopman operator associated to the controlled dynamical system (1) as the Koopman operator associated to the uncontrolled dynamical system evolving on the extended state-space defined as the product of the original state-space and the space of all control sequences, i.e., in our case $\mathbb{R}^n \times \ell(\mathcal{U})$, where $\ell(\mathcal{U})$ is the space of all sequences $(u_i)_{i=0}^\infty$ with $u_i \in \mathcal{U}$. Elements of $\ell(\mathcal{U})$ will be denoted by $\mathbf{u} := (u_i)_{i=0}^\infty$. The dynamics of the extended state

$$\chi = \begin{bmatrix} x \\ \mathbf{u} \end{bmatrix},$$

is described by

$$\chi^+ = F(\chi) := \begin{bmatrix} f(x, \mathbf{u}(0)) \\ S\mathbf{u} \end{bmatrix}, \quad (8)$$

where S is the left shift operator, i.e. $(S\mathbf{u})(i) = \mathbf{u}(i+1)$, and $\mathbf{u}(i)$ denotes the i th element of the sequence \mathbf{u} .

The Koopman operator $\mathcal{K} : \mathcal{H} \rightarrow \mathcal{H}$ associated to (8) is defined by

$$(\mathcal{K}\phi)(\chi) = \phi(F(\chi)) \quad (9)$$

for each $\phi : \mathbb{R}^n \times \ell(\mathcal{U}) \rightarrow \mathbb{R}$ belonging to some space of observables \mathcal{H} .

The Koopman operator (9) is a linear operator fully describing the non-linear dynamical system (1) provided that \mathcal{H} contains the components of the non-extended state³ x_i , $i = 1, \dots, n$. For example, spectral properties of the operator \mathcal{K} should provide information on spectral properties of the nonlinear dynamical system (1).

3.2. EDMD for controlled systems

In this paper, however, we are not interested in spectral properties but rather in time-domain prediction of trajectories of (1). To this end, we construct a finite-dimensional approximation to the operator \mathcal{K} which will yield a predictor of the form (2). In order to do so, we adapt the extended dynamic mode decomposition algorithm (EDMD) of Williams et al. (2015) to the controlled setting. The EDMD is a data-driven algorithm to construct finite-dimensional approximations to the Koopman operator. The algorithm assumes that a collection of data (χ_j, χ_j^+) , $j = 1, \dots, K$ satisfying $\chi_j^+ = F(\chi_j)$ is available and seeks a matrix \mathcal{A} (the transpose of the finite-dimensional approximation of \mathcal{K}) minimizing

$$\sum_{j=1}^K \|\phi(\chi_j^+) - \mathcal{A}\phi(\chi_j)\|_2^2, \quad (10)$$

where

$$\phi(\chi) = [\phi_1(\chi) \quad \dots \quad \phi_{N_\phi}(\chi)]^T$$

is a vector of lifting functions (or observables) $\phi_i : \mathbb{R}^n \times \ell(\mathcal{U}) \rightarrow \mathbb{R}$, $i \in \{1, \dots, N_\phi\}$. Note that $\chi = (x, \mathbf{u})$ is in general an infinite-dimensional object and hence the objective (10) cannot be evaluated in a finite time unless ϕ_i 's are chosen in a special way.

3.2.1. Linear predictors

In order to obtain a linear predictor (2) and a computable objective function in (10) we impose that the functions ϕ_i are of the form

$$\phi_i(x, \mathbf{u}) = \psi_i(x) + \mathcal{L}_i(\mathbf{u}), \quad (11)$$

where $\psi_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is in general nonlinear but $\mathcal{L}_i : \ell(\mathcal{U}) \rightarrow \mathbb{R}$ is linear. Without loss of generality (by linearity and causality) we

³ Note that the definition of the Koopman operator implicitly assumes that \mathcal{H} is invariant under the action of \mathcal{K} and hence, in the controlled setting, \mathcal{H} will typically automatically contain also functions depending on \mathbf{u} .

can assume that $N_\phi = N + m$ for some $N > 0$ and that the vector of lifting functions $\phi = [\phi_1, \dots, \phi_{N_\phi}]^\top$ is of the form

$$\phi(x, \mathbf{u}) = \begin{bmatrix} \psi(x) \\ \mathbf{u}(0) \end{bmatrix}, \quad (12)$$

where $\psi = [\psi_1, \dots, \psi_N]^\top$ and $\mathbf{u}(0) \in \mathbb{R}^m$ denotes the first component of the sequence \mathbf{u} . Since we are not interested in predicting future values of the control sequence, we can disregard the last m components of each term $\phi(x_j^+) - A\phi(x_j)$ in (10). Denoting \bar{A} the first N rows of A and decomposing this matrix such that $\bar{A} = [A, B]$ with $A \in \mathbb{R}^{N \times N}$, $B \in \mathbb{R}^{N \times m}$ and using the notation $x_j = (x_j, \mathbf{u}_j)$ in (10), leads to the minimization problem

$$\min_{A, B} \sum_{j=1}^K \|\psi(x_j^+) - A\psi(x_j) - B\mathbf{u}_j(0)\|_2^2. \quad (13)$$

Minimizing (13) over A and B leads to the predictor of the form (2) starting from the initial condition

$$z_0 = \psi(x_0) := \begin{bmatrix} \psi_1(x_0) \\ \vdots \\ \psi_N(x_0) \end{bmatrix}. \quad (14)$$

The matrix C is obtained simply as the best projection of x onto the span of ψ_i 's in a least squares sense, i.e., as the solution to

$$\min_C \sum_{j=1}^K \|x_j - C\psi(x_j)\|_2^2. \quad (15)$$

We emphasize that (13) and (15) are linear least squares problem that can be readily solved using linear algebra.

Remark 1. Note that the solution to (15) is trivial if the set of lifting functions $\{\psi_1, \dots, \psi_N\}$ contains the state observable, i.e., if, after possible reordering, $\psi_i(x) = x_i$ for all $i \in \{1, \dots, n\}$. In this case, the solution to (15) is $C = [I, 0]$, where I is the identity matrix of size n .

The resulting algorithm for constructing the linear predictor (2) is concisely summarized in Section 4.

3.2.2. Bilinear predictors

Predictors with a more complex structure can be obtained by imposing a structure on the functions ϕ_i different than the linear structure (11). In particular, bilinear predictors of the form (5) can be obtained by requiring that

$$\phi_i(x, \mathbf{u}) = \psi_i(x) + \xi_i(x)\mathcal{L}_i(\mathbf{u})$$

for some nonlinear functions ψ_i, ξ_i and linear operators \mathcal{L}_i .

A bilinear predictor of the form (5) can be tight (in the sense of convergence of predicted trajectories to the true ones as the number of basis functions tends to infinity) under the assumption that the discrete-time mapping (1) comes from a discretization of a continuous-time system and the discretization interval tends to zero and the underlying continuous-time dynamics is input-affine; see Section IV-C of Surana (2016) for more details. This bilinearity phenomenon is well known from the classical Carleman linearization in continuous time (Carleman, 1932). In this work, however, we focus on linear predictors since they are immediately amenable to the range of mature linear control design techniques. The use of bilinear predictors for controller design is left for further investigation.

4. Numerical algorithm – finding A, B, C

We assume that a set of data

$$\mathbf{X} = [x_1, \dots, x_K], \mathbf{Y} = [y_1, \dots, y_K], \mathbf{U} = [u_1, \dots, u_K] \quad (16)$$

satisfying the relation $y_i = f(x_i, u_i)$ is available. Note that we do not assume any temporal ordering of the data. In particular, the data is not required to come from one trajectory of (1).

Given the data $\mathbf{X}, \mathbf{Y}, \mathbf{U}$ in (16), the matrices $A \in \mathbb{R}^{N \times N}$ and $B \in \mathbb{R}^{N \times m}$ in (2) are obtained as the best linear one-step predictor in the lifted space in a least-squares sense, i.e., they are obtained as the solution to the optimization problem

$$\min_{A, B} \|\mathbf{Y}_{\text{lift}} - A\mathbf{X}_{\text{lift}} - B\mathbf{U}\|_F, \quad (17)$$

where

$$\mathbf{X}_{\text{lift}} = [\psi(x_1), \dots, \psi(x_K)], \mathbf{Y}_{\text{lift}} = [\psi(y_1), \dots, \psi(y_K)], \quad (18)$$

with

$$\psi(x) := \begin{bmatrix} \psi_1(x) \\ \vdots \\ \psi_N(x) \end{bmatrix} \quad (19)$$

being a given basis (or dictionary) of nonlinear functions. The symbol $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. The matrix $C \in \mathbb{R}^{n \times N}$ is obtained as the best linear least-squares estimate of \mathbf{X} given \mathbf{X}_{lift} , i.e., the solution to

$$\min_C \|\mathbf{X} - C\mathbf{X}_{\text{lift}}\|_F. \quad (20)$$

The analytical solution to (17) is

$$[A, B] = \mathbf{Y}_{\text{lift}}[\mathbf{X}_{\text{lift}}, \mathbf{U}]^\dagger, \quad (21)$$

where † denotes the Moore–Penrose pseudoinverse of a matrix. The analytical solution to (20) is

$$C = \mathbf{X}\mathbf{X}_{\text{lift}}^\dagger.$$

Notice the close relation of the resulting algorithm to the DMD with control proposed in Proctor et al. (2016). There, however, no lifting is applied and the least squares fit (17) is carried out on the original data, limiting the predictive power.

4.1. Practical considerations

The analytical solution (21) is not the preferred method of solving the least-squares problem (17) in practice. In particular, for larger data sets with $K \gg N$ it is beneficial to solve instead the normal equations associated to (17). The normal equations read

$$\mathbf{V} = \mathcal{M}\mathbf{G} \quad (22)$$

with variable $\mathcal{M} = [A, B]$ and data

$$\mathbf{G} = \begin{bmatrix} \mathbf{X}_{\text{lift}} \\ \mathbf{U} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\text{lift}} \\ \mathbf{U} \end{bmatrix}^\top, \quad \mathbf{V} = \mathbf{Y}_{\text{lift}} \begin{bmatrix} \mathbf{X}_{\text{lift}} \\ \mathbf{U} \end{bmatrix}^\top.$$

Any solution to (22) is a solution to (17). Importantly, the size of the matrices \mathbf{G} and \mathbf{V} is $(N + m) \times (N + m)$ respectively $N \times (N + m)$ and hence independent of the number of samples K in the data set (16). The same considerations hold for the least-squares problem (20). Note that, in practice, the lifting functions ψ_i will typically contain the state itself in which case the solution to (20) is just the selection of appropriate indices of \mathbf{X}_{lift} , i.e., after possible reordering, $C = [I, 0]$.

If lifting to a very high dimensional space is required, it may be worth exploring the so called kernel methods known from

machine learning which do not require an explicit evaluation of the lifting mapping ψ . These methods were successfully applied to the standard EDMD algorithm in Williams, Rowley, and Kevrekidis (2015), leading to substantial computational savings.

5. Model predictive control

In this section we describe how the linear predictor (2) can be used to design an MPC controller for the nonlinear system (1) with computational complexity comparable to that of an MPC controller for a linear system of the same state-space dimension and number of control inputs. We recall that MPC is a control strategy where the control input at each time step of the closed-loop operation is obtained by solving an optimization problem where a user-specified cost function (e.g., the energy or tracking error) is minimized along a prediction horizon subject to constraints on the control inputs and state variables. Traditionally, linear MPC solves a convex quadratic program, thereby allowing for an extremely fast evaluation of the control input. Nonlinear MPC, on the other hand, solves a difficult non-convex optimization problem, thereby requiring far more computational resources and/or relying on local solutions only; see, e.g., Grüne and Pannek (2011) for an overview of nonlinear MPC. Just as linear MPC, the lifting-based MPC for nonlinear systems developed here relies on convex quadratic programming, thereby avoiding all issues associated with non-convex optimization and allowing for an extremely fast evaluation of the control input. We first describe the proposed MPC controller in its most general form and subsequently, in Section 5.2, describe how a traditional nonlinear MPC problem translates to the proposed one.

The proposed model predictive controller solves at each time instance k of the closed-loop operation the optimization problem

$$\begin{aligned} & \underset{u_i, z_i}{\text{minimize}} && J\left((u_i)_{i=0}^{N_p-1}, (z_i)_{i=0}^{N_p}\right) \\ & \text{subject to} && z_{i+1} = Az_i + Bu_i, \quad i = 0, \dots, N_p - 1 \\ & && E_i z_i + F_i u_i \leq b_i, \quad i = 0, \dots, N_p - 1 \\ & && E_{N_p} z_{N_p} \leq b_{N_p} \\ & \text{parameter} && z_0 = \psi(x_k), \end{aligned} \quad (23)$$

where N_p is the prediction horizon and the convex quadratic cost function J is given by

$$\begin{aligned} J\left((u_i)_{i=0}^{N_p-1}, (z_i)_{i=0}^{N_p}\right) &= z_{N_p}^\top Q_{N_p} z_{N_p} + q_{N_p}^\top z_{N_p} \\ &+ \sum_{i=0}^{N_p-1} z_i^\top Q_i z_i + u_i^\top R_i u_i + q_i^\top z_i + r_i^\top u_i \end{aligned}$$

with $Q_i \in \mathbb{R}^{N \times N}$ and $R_i \in \mathbb{R}^{m \times m}$ positive semidefinite. The matrices $E_i \in \mathbb{R}^{n_c \times N}$ and $F_i \in \mathbb{R}^{n_c \times m}$ and the vector $b_i \in \mathbb{R}^{n_c}$ define state and input polyhedral constraints. The optimization problem (23) is parametrized by the current state of the nonlinear dynamical system x_k . This optimization problem defines a feedback controller

$$\kappa(x_k) = u_0^*(x_k),$$

where $u_0^*(x_k)$ denotes an optimal solution to problem (23) parametrized by the current state x_k .

Several observations are in order:

- (1) The optimization problem (23) is a convex quadratic programming problem (QP).
- (2) At each time step k , the predictions are initialized from the lifted state $\psi(x_k)$.
- (3) Nonlinear functions of the original state x can be penalized in the cost function and included among the constraints by including these nonlinear functions among the lifting functions ψ_i . For example, if one wished for some reason to minimize $\sum_{i=0}^{N_p-1} \cos(\|x_i\|_\infty)$, one could simply set $\psi_1 = \cos(\|x\|_\infty)$ and $q = [1, 0, 0, \dots, 0]^\top$. See Section 5.2 for more details.

5.1. Eliminating dependence on the lifting dimension

In this section we show that the computational complexity of solving the MPC problem (23) can be rendered independent of the dimension of the lifted state N . This is achieved by transforming (23) in the so-called *dense form*

$$\begin{aligned} & \underset{U \in \mathbb{R}^{mN_p}}{\text{minimize}} && U^\top H U + h^\top U + z_0^\top G U \\ & \text{subject to} && LU + Mz_0 \leq c \\ & \text{parameter} && z_0 = \psi(x_k), \end{aligned} \quad (24)$$

for some positive-semidefinite matrix $H \in \mathbb{R}^{mN_p \times mN_p}$ and some matrices and vectors $h \in \mathbb{R}^{mN_p}$, $G \in \mathbb{R}^{N \times mN_p}$, $L \in \mathbb{R}^{n_c N_p \times mN_p}$, $M \in \mathbb{R}^{n_c N_p \times N}$ and $c \in \mathbb{R}^{n_c N_p}$. The optimization is over the vector of predicted control inputs $U = [u_0^\top, u_1^\top, \dots, u_{N_p-1}^\top]^\top$. This “dense” formulation can be readily derived from the “sparse” formulation (23) by solving explicitly for z_i ’s and concatenating the point-wise-in-time stage costs and constraints; see the Appendix for explicit expressions for the data matrices of (24) in terms of those of (23).

Notice that, crucially, the size of the Hessian H or the number of the constraints n_c in the dense formulation (24) is independent of the size of the lift N . Hence, once the nonlinear mapping $z_0 = \psi(x_k)$ is evaluated, the computational cost of solving (24) is comparable to solving a standard linear MPC on the same prediction horizon, with the same number of control inputs and with the dimension of the state-space equal to n rather than N . This comes from the fact that the cost of solving an MPC problem in a dense form is independent of the dimension of the state-space, once the data matrices in (24) are formed. Importantly, these matrices are *fixed* and *precomputed offline* before deploying the controller (with the exception of the inexpensive matrix-vector multiplication $z_0^\top G$). This is in contrast with other MPC schemes for nonlinear systems where these matrices have to be re-computed at each time step of the closed-loop operation, thereby greatly increasing the computational cost.

The closed-loop operation of the lifting-based MPC can be summarized by the following algorithm, where $U_{1:m}^*$ denotes the first m components of U^* :

Algorithm 1 Lifting MPC – closed-loop operation

- 1: **for** $k = 0, 1, \dots$ **do**
 - 2: Set $z_0 := \psi(x_k)$
 - 3: Solve (24) to get an optimal solution U^*
 - 4: Set $u_k = U_{1:m}^*$
 - 5: $x_{k+1} = f(x_k, u_k)$ [= apply u_k to system (1)]
-

5.2. Transforming NMPC to Koopman MPC

In this section we describe in detail how a traditional nonlinear MPC problem translates.⁴ to the proposed MPC (23). We assume a nonlinear MPC problem which at each time step k of the closed-loop operation solves the optimization problem

$$\begin{aligned} & \underset{u_i, \bar{x}_i}{\text{minimize}} && l_{N_p}(x_{N_p}) + \sum_{i=0}^{N_p-1} l_i(\bar{x}_i) + u_i^\top \bar{R}_i u_i + \bar{r}_i^\top u_i \\ & \text{subject to} && \bar{x}_{i+1} = f(\bar{x}_i, u_i), \quad i = 0, \dots, N_p - 1 \\ & && c_{x_i}(\bar{x}_i) + c_{u_i}^\top u_i \leq 0, \quad i = 0, \dots, N_p - 1 \\ & && c_{x_{N_p}}(\bar{x}_{N_p}) \leq 0 \\ & && \bar{x}_0 = x_k, \end{aligned} \quad (25)$$

⁴ By translate, we do not mean to rewrite in an equivalent form. The problem (23) is of course only an approximation to (25) since the lifted linear predictor is not exact unless the dynamical system is linear.

where the notation \bar{x} is used to distinguish the predicted state \bar{x} , used only within the optimization problem (25), from the true measured state x of the dynamical system (1). Notice that the true nonlinear dynamics $x^+ = f(x, u)$ appears as a constraint of (25); in addition, the functions l_i and c_{x_i} can be nonlinear and hence the optimization problem (25) is in general nonconvex and extremely hard to solve to global optimality.

In order to translate (25) to the proposed form (23) we assume that a predictor of the form (2) with matrices A and B has been constructed as described in Section 4, using a lifting mapping $\psi = [\psi_1, \dots, \psi_N]^T$. The matrices A, B appear in the first constraint of (23) and the lifting mapping ψ is used for initialization $z_0 = \psi(x_k)$. In order to obtain the remaining data matrices of (23) we assume without loss of generality that $\psi_i(x) = l_i(x)$, $i = 0, \dots, N_p$ and $\psi_{N_p+i}(x) = c_{x_i}(x)$, $i = 0, \dots, N_p$. With this assumption, the remaining data is given by $Q_i = 0$, $R_i = \bar{R}_i$, $r = \bar{r}_i$, $q_i = [0_{1 \times i}, 1, 0_{1 \times N-1-i}]$, $E_i = [0_{1 \times N_p+i}, 1, 0_{1 \times N-N_p-1-i}]$, $F_i = c_{u_i}^T$, $b_i = 0$, where $0_{i \times j}$ denotes the matrix of zeros of size $i \times j$. Note that this derivation assumed that the constraint functions c_{x_i} and c_{u_i} are scalar-valued; for vector valued constraint functions, the approach is analogous, setting the lifting functions ψ_i equal to the individual components of the constraint functions c_{x_i} .

We also note that this canonical approach always leads to a linear cost function in (23). However, in special cases, when some of the cost functions $l_i(x_i)$ are convex quadratic, one may want to use the freedom of the formulation (23) and instead of setting $\psi_i = l_i$, use the quadratic terms in the cost function of (23), thereby reducing the dimension of the lift. See Section 8.2 for an example.

6. Theoretical analysis

In this section we discuss theoretical properties of the EDMD algorithm of Section 3.2. Full exposition of the theoretical analysis of EDMD is beyond the scope of this paper and therefore we only summarize the authors' results obtained concurrently in Korda and Mezić (2018), where the reader is referred for proofs of the theorems stated in this section and additional results. Note that the results of Theorems 1 and 2 are not new and were, to the best of our knowledge, first rigorously proven in Klus, Koltai, and Schütte (2016, Section 3.4) and alluded to already in the original EDMD paper (Williams et al., 2015); here we state them in a language suitable for our purposes.

We work in an abstract setting with a dynamical system

$$x^+ = F(x)$$

with $F : \mathcal{M} \rightarrow \mathcal{M}$, where \mathcal{M} is a given separable topological space. This encompasses both the finite-dimensional setting with $\mathcal{M} = \mathbb{R}^n$ and the infinite-dimensional controlled setting of Section 3.1 with $\mathcal{M} = \mathbb{R}^n \times l(\mathcal{U})$. The Koopman operator $\mathcal{K} : \mathcal{H} \rightarrow \mathcal{H}$ on a space of observables \mathcal{H} (with $\phi : \mathcal{M} \rightarrow \mathbb{R}$ for all $\phi \in \mathcal{H}$) is defined as in (9) by $\mathcal{K}\phi = \phi \circ F$ for all $\phi \in \mathcal{H}$. We assume the EDMD algorithm of Section 3.2, i.e., we solve the optimization problem

$$\min_{\mathcal{A} \in \mathbb{R}^{N \times N}} \sum_{i=1}^K \|\phi(x_i^+) - \mathcal{A}\phi(x_i)\|_2^2, \quad (26)$$

where

$$\phi(x) = [\phi_1(x), \dots, \phi_N(x)]^T$$

with $\phi_i \in \mathcal{H}$, $i = 1, \dots, N$, being linearly independent basis functions. Denoting \mathcal{H}_N the span of ϕ_1, \dots, ϕ_N , the finite-dimensional approximation of the Koopman operator $\mathcal{K}_{N,K} : \mathcal{H}_N \rightarrow \mathcal{H}_N$ obtained from (26) is defined for any $g = c^T \phi \in \mathcal{H}_N$ by

$$\mathcal{K}_{N,K}g = c^T \mathcal{A}_{N,K} \phi, \quad (27)$$

where $\mathcal{A}_{N,K}$ is the optimal solution of (26).

6.1. EDMD as L_2 projection

First we give a characterization of the EDMD algorithm as an L_2 projection. Given an arbitrary nonnegative measure μ on \mathcal{M} , we define the $L_2(\mu)$ projection⁵ of a function g onto \mathcal{H}_N as

$$\begin{aligned} P_N^\mu g &= \arg \min_{h \in \mathcal{H}_N} \|h - g\|_{L_2(\mu)} = \arg \min_{h \in \mathcal{H}_N} \int_{\mathcal{M}} (h - g)^2 d\mu \\ &= \phi^T \arg \min_{c \in \mathbb{R}^N} \int_{\mathcal{M}} (c^T \phi - g)^2 d\mu. \end{aligned} \quad (28)$$

We have the following characterization of $\mathcal{K}_{N,K}$.

Theorem 1. Let $\hat{\mu}_K$ denote the empirical measure associated to the points x_1, \dots, x_K , i.e., $\hat{\mu}_K = \frac{1}{K} \sum_{i=1}^K \delta_{x_i}$, where δ_{x_i} denotes the Dirac measure at x_i . Then for any $g \in \mathcal{H}_N$

$$\mathcal{K}_{N,K}g = P_N^{\hat{\mu}_K} \mathcal{K}g = \arg \min_{h \in \mathcal{H}_N} \|h - \mathcal{K}g\|_{L_2(\hat{\mu}_K)}, \quad (29)$$

i.e.,

$$\mathcal{K}_{N,K} = P_N^{\hat{\mu}_K} \mathcal{K}|_{\mathcal{H}_N}, \quad (30)$$

where $\mathcal{K}|_{\mathcal{H}_N} : \mathcal{H}_N \rightarrow \mathcal{H}$ is the restriction of the Koopman operator to the subspace \mathcal{H}_N .

In words, the operator $\mathcal{K}_{N,K}$ is the L_2 projection of the operator \mathcal{K} on the span of ϕ_1, \dots, ϕ_N with respect to the empirical measure supported on the samples x_1, \dots, x_K .

6.2. Convergence of EDMD

Now we turn into analyzing convergence $\mathcal{K}_{N,K}$ to \mathcal{K} as the number of samples K and the number of basis functions N tend to infinity.

First we analyze convergence as K tends to infinity. For this we assume a probabilistic sampling model. That is, we assume that the space \mathcal{M} is endowed with a probability measure μ and that the samples x_1, \dots, x_K are independent identically distributed (iid) samples from the distribution μ and we assume that $\mathcal{H} = L_2(\mu)$. We invoke the following non-restrictive assumption.

Assumption 1 (μ Independence). The basis functions ϕ_1, \dots, ϕ_N are such that

$$\mu(\{x \in \mathcal{M} \mid c^T \phi(x) = 0\}) = 0$$

for all nonzero $c \in \mathbb{R}^N$.

This is a natural assumption ensuring that the measure μ is not supported on a zero level set of a nontrivial linear combination of the basis functions used.

Finally we define

$$\mathcal{K}_N = P_N^\mu \mathcal{K}|_{\mathcal{H}_N}, \quad (31)$$

i.e., the $L_2(\mu)$ projection of the Koopman operator \mathcal{K} on \mathcal{H}_N . Then we have the following result:

Theorem 2. If Assumption 1 holds, then with probability one

$$\lim_{K \rightarrow \infty} \|\mathcal{K}_{N,K} - \mathcal{K}_N\| = 0, \quad (32)$$

where $\|\cdot\|$ is any operator norm and

$$\lim_{K \rightarrow \infty} \text{dist}(\sigma(\mathcal{K}_{N,K}), \sigma(\mathcal{K}_N)) = 0, \quad (33)$$

where $\sigma(\cdot) \subset \mathbb{C}$ denotes the spectrum of an operator and $\text{dist}(\cdot, \cdot)$ the Hausdorff metric on subsets of \mathbb{C} .

⁵ The Hilbert space $L_2(\mu)$ is the space of all square integrable functions with respect to the measure μ .

Theorem 2 says that the EDMD approximations $\mathcal{K}_{N,K}$ converge in the operator norm to the $L_2(\mu)$ projection of the Koopman operator onto the span of the basis functions ϕ_1, \dots, ϕ_N .

Having established convergence of $\mathcal{K}_{N,K}$ to \mathcal{K} we turn to studying convergence of \mathcal{K}_N to \mathcal{K} . Since the operator $\mathcal{K}_{N,K}$ is defined only on \mathcal{H}_N we extend it to all of \mathcal{H} by precomposing with the projection on \mathcal{H}_N , i.e., we study convergence of $\mathcal{K}_N P_N^\mu : \mathcal{H} \rightarrow \mathcal{H}$ to $\mathcal{K} : \mathcal{H} \rightarrow \mathcal{H}$.

We have the following result:

Theorem 3. *If $(\phi_i)_{i=1}^\infty$ forms an orthonormal basis of $\mathcal{H} = L_2(\mu)$ and if $\mathcal{K} : \mathcal{H} \rightarrow \mathcal{H}$ is continuous, then the sequence of operators $\mathcal{K}_N P_N^\mu = P_N^\mu \mathcal{K} P_N^\mu$ converges to \mathcal{K} as $N \rightarrow \infty$ in the strong operator topology, i.e., for all $g \in \mathcal{H}$*

$$\lim_{N \rightarrow \infty} \|\mathcal{K}_N P_N^\mu g - \mathcal{K}g\|_{L_2(\mu)} = 0.$$

In particular, if $g \in \mathcal{H}_{N_0}$ for some $N_0 \in \mathbb{N}$, then

$$\lim_{N \rightarrow \infty} \|\mathcal{K}_N g - \mathcal{K}g\|_{L_2(\mu)} = 0.$$

Theorem 3 tells us that the sequence of operators $\mathcal{K}_N P_N^\mu$ converges strongly to \mathcal{K} . For additional results on spectral convergence of \mathcal{K}_N to \mathcal{K} , weak spectral convergence of $\mathcal{K}_{N,N}$ (i.e., with $K = N$) to \mathcal{K} and for a method to construct \mathcal{K}_N directly, without the need for sampling, see [Korda and Mezić \(2018\)](#).

Now we use **Theorem 3** to establish convergence of finite-horizon predictions of a given vector-valued observable $g \in \mathcal{H}_{N_0}^{n_g}$. For our purposes, the most pertinent situation is when g is the state observable, i.e., $g(x) = x$ in which case the following result pertains to predictions of the state itself. We let $\mathcal{A}_{N,K}$ denote the solution to (26) and set $\mathcal{A}_N = \lim_{K \rightarrow \infty} \mathcal{A}_{N,K}$. Since $g \in \mathcal{H}_{N_0}^{n_g}$, it follows that for every $N \geq N_0$ there exists a matrix $C_N \in \mathbb{R}^{n_g \times N}$ such that $g = C_N \phi_N$, where $\phi_N = [\phi_1, \dots, \phi_N]^\top$. With this notation the following result holds:

Corollary 1. *If the assumptions of [Theorem 3](#) hold and $g \in \mathcal{H}_{N_0}^{n_g}$ for some $N_0 \in \mathbb{N}$, then for any finite prediction horizon $N_p \in \mathbb{N}$*

$$\lim_{N \rightarrow \infty} \sup_{i \in \{0, \dots, N_p\}} \int_{\mathcal{M}} (C_N \mathcal{A}_N^i \phi_N - g \circ F^i)^2 d\mu = 0. \quad (34)$$

In words, predictions over any finite horizon converge in the $L_2(\mu)$ norm. Unfortunately, **Corollary 1** does not immediately apply to the linear predictors designed in Section 3.2 as the set of basis functions (12) does not form an orthonormal basis of \mathcal{H} because of the special structure of these basis functions which ensures linearity of the resulting predictors. In this setting, one can only prove convergence of \mathcal{K}_N to $P_\infty^\mu \mathcal{K}|_{\mathcal{H}_\infty}$, where P_∞^μ is the $L_2(\mu)$ projection onto $\mathcal{H}_\infty := \{\phi_i \mid i \in \mathbb{N}\}$.

7. Extensions

In this section, we describe extensions of the proposed approach to input–output dynamical systems and to systems with disturbances/noise.

7.1. Input–output dynamical systems

In this section we describe how the approach can be generalized to the case when full state measurements are not available, but rather only certain output is measured. To this end, consider the dynamical system

$$\begin{aligned} x^+ &= f(x, u), \\ y &= h(x), \end{aligned} \quad (35)$$

where y is the measured output and $h : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$.

7.1.1. Dynamics (35) is known

If the dynamics (35) is known, one can construct a predictor of the form (2) by applying the algorithm of Section 4 to data obtained from simulation of the dynamical system (35). In closed-loop operation, the predictor (2) is then used in conjunction with a state-estimator for the dynamical system (35). Alternatively, one can design, using linear observer design methodologies, a state estimator directly for the linear predictor (2). Interestingly, doing so obviates the need to evaluate the lifting mapping $z = \psi(x)$ in closed loop, as the lifted state is directly estimated. This idea is closely related to the use of the Koopman operator for state estimation ([Surana, 2016](#); [Surana & Banaszuk, 2016](#)).

7.1.2. Dynamics (35) is not known

If the dynamics (35) is not known and only input–output data is available, one could construct a predictor of the form (2) or (5) by taking as lifting functions only functions of the output y , i.e., $\psi_i(x) = \phi_i(h(x))$. This, however, would be extremely restrictive as this severely restricts the class of lifting functions available. Indeed, if, for example, $h(x) = x_1$, only functions of the first component are available. However, this problem can be circumvented by utilizing the fact that subsequent measurements, $h(x)$, $h(f(x))$, $h(f(f(x)))$, etc., are available and therefore we can define observables depending not only on h but also on repeated composition of h with f . In practice this corresponds to having the lifting functions depend not only on the current measured output but also on several previous measured outputs (and inputs, in the controlled setting). The use of time-delayed measurements is classical in system identification theory (see, e.g., [Ljung, 1998](#)) but has also appeared in the context of Koopman operator approximation (see, e.g., [Brunton, Brunton, Proctor, Kaiser, & Kutz, 2017](#); [Tu, Rowley, Luchtenburg, Brunton, & Kutz, 2014](#)).

Assume therefore that we are given a collection of data

$$\tilde{\mathbf{X}} = [\zeta_1, \dots, \zeta_K], \quad \tilde{\mathbf{Y}} = [\zeta_1^+, \dots, \zeta_K^+], \quad \tilde{\mathbf{U}} = [u_1, \dots, u_K]$$

where

$$\zeta_i = \begin{bmatrix} y_{i,n_d}^\top & \bar{u}_{i,n_d-1}^\top & y_{i,n_d-1}^\top & \dots & \bar{u}_{i,0}^\top & y_{i,0}^\top \end{bmatrix}^\top \in \mathbb{R}^{(n_d+1)n_h+n_d m} \quad (36)$$

$$\zeta_i^+ = \begin{bmatrix} y_{i,n_d+1}^\top & \bar{u}_{i,n_d}^\top & y_{i,n_d}^\top & \dots & \bar{u}_{i,1}^\top & y_{i,1}^\top \end{bmatrix}^\top \in \mathbb{R}^{(n_d+1)n_h+n_d m}$$

$$u_i = \bar{u}_{i,n_d}$$

with $(y_{i,j})_{j=0}^{n_d+1}$ being a vector of consecutive output measurements generated by $(\bar{u}_{i,j})_{j=0}^{n_d}$ consecutive inputs. We note that there does not need to be any temporal relation between ζ_i and ζ_{i+1} . If, however, ζ_i and ζ_{i+1} are in fact successors, then the matrices $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ have the familiar (quasi)-Hankel structure known from system identification theory.

Computation of a linear predictor then proceeds in the same way as for the full-state measurement: We lift the collected data and look for the best one-step predictor in the lifted space. This leads to the optimization problem

$$\min_{A,B} \|\tilde{\mathbf{Y}}_{\text{lift}} - A\tilde{\mathbf{X}}_{\text{lift}} - B\tilde{\mathbf{U}}\|_F, \quad (37)$$

where

$$\tilde{\mathbf{X}}_{\text{lift}} = [\psi(\zeta_1), \dots, \psi(\zeta_K)], \quad \tilde{\mathbf{Y}}_{\text{lift}} = [\psi(\zeta_1^+), \dots, \psi(\zeta_K^+)] \quad (38)$$

and

$$\psi(\zeta) = \begin{bmatrix} \psi_1(\zeta) \\ \vdots \\ \psi_N(\zeta) \end{bmatrix}$$

is a vector of real or complex-valued, possibly nonlinear, lifting functions. This leads to a linear predictor in the lifted space

$$\begin{aligned} z^+ &= Az + Bu \\ \hat{y} &= Cz, \end{aligned} \quad (39)$$

where \hat{y} is the prediction of y and C is the solution to

$$\min_C \left\| [y_{1,n_d}, \dots, y_{K,n_d}] - C\tilde{\mathbf{X}}_{\text{lift}} \right\|_F. \quad (40)$$

The predictor (39) starts from the initial condition

$$z_0 = \psi(\zeta_0),$$

where

$$\zeta_0 = [y_0^\top \ \bar{u}_{-1}^\top \ y_{-1}^\top \ \dots \ \bar{u}_{-n_d}^\top \ y_{-n_d}^\top]^\top$$

is the vector of $n_d + 1$ most recent output measurements and n_d input measurements. We remark that the solution to (40) is trivial provided that the outputs and its delays are included among the lifting functions (e.g., if $\psi_j(\zeta) = \zeta(2j)$, $j = 0, \dots, n_d$); see Remark 1.

Remark 2 (Closed-Loop Operation). The closed-loop operation of the resulting MPC controller follows the steps of Algorithm 1, only the initialization $z_0 = \psi(x_k)$ in line 2 is replaced by $z_0 = \psi(\zeta_k)$, where $\zeta_k = [y_k^\top, u_{k-1}^\top, y_{k-1}^\top, \dots, u_{k-n_d}^\top, y_{k-n_d}^\top]^\top$.

7.2. Disturbance/Noise propagation

The approach can be readily extended to systems affected by a disturbance or noise of the form

$$x^+ = f(x, u, w), \quad (41)$$

where w is the disturbance or process noise. The goal is to construct a predictor for (41) of the form

$$\begin{aligned} z^+ &= Az + Bu + Dw \\ \hat{x} &= Cz, \end{aligned} \quad (42)$$

starting from the initial condition $z_0 = \psi(x_0)$, where $\psi(\cdot)$ is the lifting mapping defined in (19). For example, if w is a stochastic disturbance with known distribution, the linear predictor of the form (42) can be used to approximately compute the distribution of the state x at a future time instance, given a sequence of control inputs up to that time.

In order to obtain the matrices of the predictor (42), we assume that we are given data of the form

$$\mathbf{X} = [x_1, \dots, x_K], \quad \mathbf{Y} = [y_1, \dots, y_K], \quad (43a)$$

$$\mathbf{U} = [u_1, \dots, u_K], \quad \mathbf{W} = [w_1, \dots, w_K] \quad (43b)$$

satisfying $y_i = f(x_i, u_i, w_i)$ for all $i = 1, \dots, K$. As in Section 4, the matrices are then obtained using least-squares regression:

$$\min_{A,B,D} \|\mathbf{Y}_{\text{lift}} - A\mathbf{X}_{\text{lift}} - B\mathbf{U} - D\mathbf{W}\|_F, \quad \min_C \|\mathbf{X} - C\mathbf{X}_{\text{lift}}\|_F. \quad (44)$$

If measurements of the disturbance w are not available, then these must be best estimated from the available data, either using one of the nonlinear estimation techniques or using the Koopman operator-based estimator proposed in Surana (2016) and Surana and Banaszuk (2016). Alternatively, if the mapping f is known (either analytically or in the form of an algorithm) and an algorithm to draw samples from the distribution of w is available, one can obtain data (43) by simulation.

Remark 3. If full-state measurements are not available, the approach can be readily combined with the approach of Section 7.1.1 or 7.1.2.

8. Numerical examples

In this section we compare the prediction accuracy of the linear lifting-based predictor (2) with several other predictors and demonstrate the use of the lifting-based MPC proposed in Section 5 for feedback control of a bilinear model of a motor and of the Korteweg–de Vries nonlinear partial differential equation. The source code for the numerical examples is available from <https://github.com/MilanKorda/KoopmanMPC/raw/master/KoopmanMPC.zip>.

8.1. Prediction comparison

In order to evaluate the proposed predictor, we compare its prediction quality with that of several commonly used predictors. The system to compare the predictors on is the classical forced Van der Pol oscillator with dynamics given by

$$\dot{x}_1 = 2x_2$$

$$\dot{x}_2 = -0.8x_1 + 2x_2 - 10x_1^2x_2 + u.$$

The predictors compared are:

- (1) Predictor based on local linearization of the dynamics at the origin,
- (2) Predictor based on local linearization of the dynamics at a given initial condition x_0 ,
- (3) Carleman linearization predictor (Carleman, 1932),
- (4) The proposed lifting-based predictor (2).

In order to obtain the lifting-based predictor, we discretize the dynamics using the Runge–Kutta four method with discretization period $T_s = 0.01$ s and simulate 200 trajectories over 1000 sampling periods (i.e., 20 s per trajectory). The control input for each trajectory is a random signal with uniform distribution over the interval $[-1, 1]$. The trajectories start from initial conditions generated randomly with uniform distribution on the unit box $[-1, 1]^2$. This data collection process results in the matrices \mathbf{X} and \mathbf{Y} of size $2 \times 2 \cdot 10^5$ and matrix \mathbf{U} of size 1×10^5 . The lifting functions ψ_i are chosen to be the state itself (i.e., $\psi_1 = x_1$, $\psi_2 = x_2$) and 100 thin plate spline radial basis functions⁶ with centers selected randomly with uniform distribution on the unit box. The dimension of the lifted state-space is therefore $N = 102$.

The degree of the Carleman linearization is set to 14, resulting in the size of the Carleman linearization predictor of 120 (= the number of monomials of degree less than or equal to 14 in two variables). The B matrix for Carleman linearization predictor is set to $[0, 1, 0, \dots, 0]^\top$.

Fig. 2 compares the predictions starting from two initial conditions $x_0^1 = [0.5, 0.5]^\top$, $x_0^2 = [-0.1, -0.5]^\top$ generated by a control signal $u(t)$ being a square wave with unit magnitude and period 0.3 s. Table 1 reports the relative root mean squared errors (RMSE)

$$\text{RMSE} = 100 \cdot \frac{\sqrt{\sum_k \|x_{\text{pred}}(kT_s) - x_{\text{true}}(kT_s)\|_2^2}}{\sqrt{\sum_k \|x_{\text{true}}(kT_s)\|_2^2}} \quad (45)$$

for each predictor averaged over 100 randomly sampled initial conditions with the same square wave forcing. We observe that the lifting-based Koopman predictor is far superior to the remaining predictors. Finally, Table 2 reports the prediction accuracy of the lifting predictor in terms of the average RMSE error as a function of the dimension of the lift N ; we observe that, as expected, the prediction error decreases with increasing N , albeit not monotonously.

⁶ Thin plate spline radial basis function with center at x_0 is defined by $\psi(x) = \|x - x_0\|^2 \log(\|x - x_0\|)$.

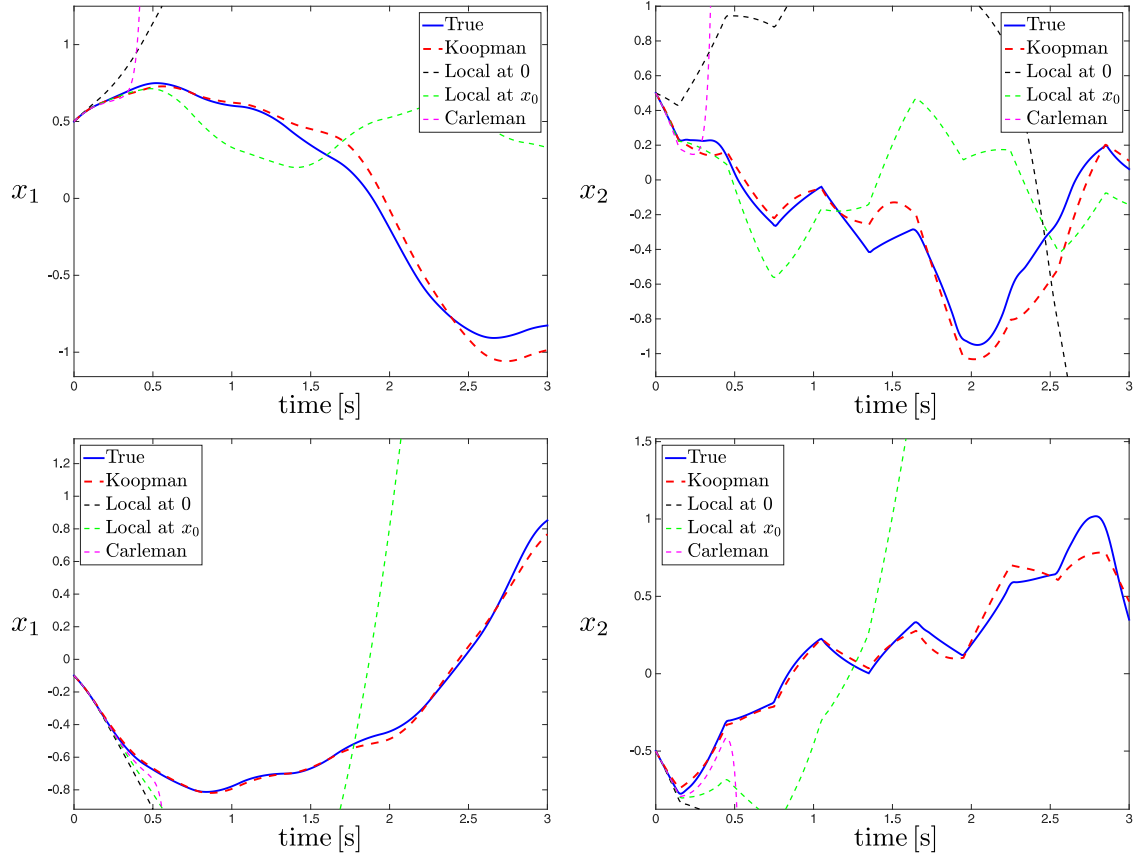


Fig. 2. Prediction comparison for the forced Van der Pol oscillator – Top: initial condition $x_0 = [0.5, 0.5]^T$. Bottom: initial condition $x_0 = [-0.1, -0.5]^T$. The forcing $u(t)$ is in both cases a square wave with unit amplitude and period 0.3 s.

Table 1

Prediction comparison – average RMSE (45) over 100 randomly sampled initial conditions: comparison among different predictors.

x_0	Average RMSE
Koopman	24.4%
Local linearization at x_0	$2.83 \cdot 10^{-3}\%$
Local linearization at 0	912.5%
Carleman	$5.08 \cdot 10^{22}\%$

Table 2

Prediction comparison – lifting-based Koopman predictor – average prediction RMSE over 100 randomly sampled initial conditions as a function of the dimension of the lift.

N	5	10	25	50	75	100
Average RMSE	66.5%	44.9%	47.0%	38.7%	30.6%	24.4%

8.2. Feedback control of a bilinear motor

In this section we apply the proposed approach to the control of a bilinear model of a DC motor (Daniel-Berhe & Unbehauen, 1998). The model reads

$$\begin{aligned}\dot{x}_1 &= -(R_a/L_a)x_1 - (k_m/L_a)x_2 u + u_a/L_a \\ \dot{x}_2 &= -(B/J)x_2 + (k_m/J)x_1 u - \tau_l/J, \\ y &= x_2\end{aligned}$$

where x_1 is the rotor current, x_2 the angular velocity and the control input u is the stator current and the output y is the angular velocity. The parameters are $L_a = 0.314$, $R_a = 12.345$, $k_m = 0.253$, $J = 0.00441$, $B = 0.00732$, $\tau_l = 1.47$, $u_a = 60$. Notice in particular the

bilinearity between the state and the control input. The physical constraints on the control input are $u \in [-4, 4]$, which we scale to $[-1, 1]$.

The goal is to design an MPC controller based on Section 7.1.2, i.e., assuming only input–output data available and no explicit knowledge of the model. In order to obtain the lifting-based predictor (39), we discretize the scaled dynamics using the Runge–Kutta four method with discretization period $T_s = 0.01$ s and simulate 200 trajectories over 1000 sampling periods (i.e., 20 s per trajectory). The control input for each trajectory is a random signal uniformly distributed on $[-1, 1]$. The trajectories start from initial conditions generated randomly with uniform distribution on the unit box $[-1, 1]^2$. We choose the number of delays $n_d = 1$. The lifting functions ψ_i are chosen to be the time-delayed vector $\zeta \in \mathbb{R}^3$, defined in (36), and 100 thin plate spline radial basis functions (see Footnote 6) with centers selected randomly with uniform distribution over $[-1, 1]^3$. The dimension of the lifted state-space is therefore $N = 103$. First, in Fig. 3, we compare the output predictions for two different, randomly chosen, initial conditions against the predictor based on local linearization at a given initial condition. The prediction accuracy of the proposed predictor is superior, especially for longer prediction times. This is documented further in Table 3 by the relative root mean-squared errors (45) over a one-second prediction horizon averaged over one hundred randomly sampled initial conditions. Both in Fig. 3 and Table 3, the control signal was a pseudo-random binary signal generated anew for each initial condition.

The control objective is to track a given angular velocity reference y_r , which translates into the objective function minimized in

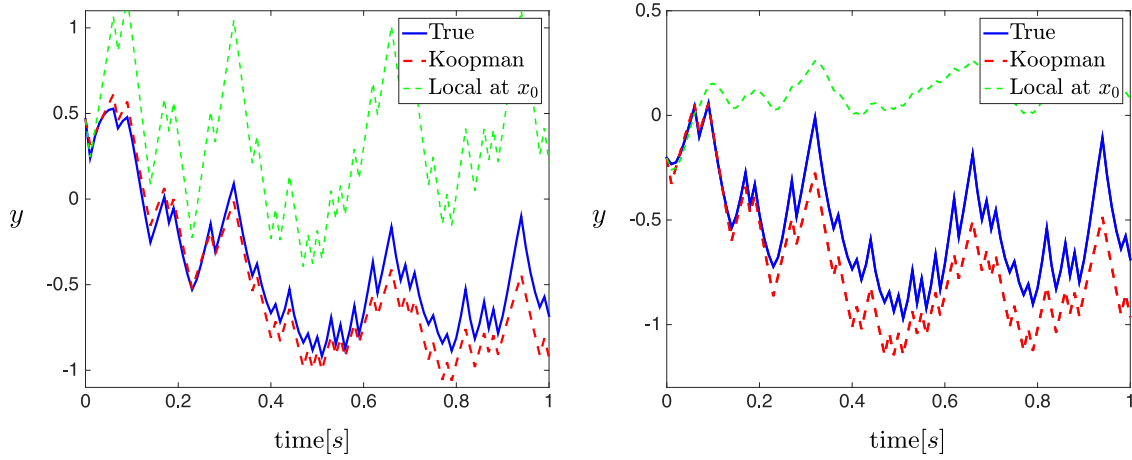


Fig. 3. Feedback control of a bilinear motor – predictor comparison. Left: initial condition $x_0 = [0.887, 0.587]^T$. Right: initial condition $x_0 = [-0.404, -0.126]^T$.

Table 3

Feedback control of a bilinear motor – prediction RMSE (45) for 100 randomly generated initial conditions.

	Koopman	Local linearization at x_0
Average RMSE	32.3 %	135.5 %

the MPC problem

$$J = (Cz_{N_p} - y_r)^T Q_{N_p} (Cz_{N_p} - y_r) + \sum_{i=0}^{N_p-1} (Cz_i - y_r)^T Q (Cz_i - y_r) + u_i^T R u_i \quad (46)$$

with $C = [1, 0, \dots, 0]$. This tracking objective function readily translates to the canonical form (23) by expanding the quadratic forms and neglecting constant terms. The cost function matrices were chosen as $Q = Q_{N_p} = 1$ and $R = 0.01$. The prediction horizon was set to one second, which results in $N_p = 100$. We compare the Koopman operator-based MPC controller (K-MPC) with an MPC controller based on local linearization (L-MPC) in two scenarios. In the first one we do not impose any constraints on the output and track a piecewise constant reference. In the second one, we impose the constraint $y \in [-0.4, 0.4]$ and track a time-varying reference $y_r(t) = 0.5 \cos(2\pi t/3)$, which violates the output constraint for some portion of the simulated period. The simulation results are shown in Fig. 4. We observe a virtually identical tracking performance in the first case. In the second case, however, the local-linearization controller becomes infeasible and hence cannot complete the entire simulation period.⁷ This infeasibility occurs due to the inaccurate predictions of the local linearization predictor over longer prediction horizons. The proposed K-MPC controller, on the other hand, does not run infeasible and completes the simulation period without violating the constraints.

Note that, even in the first scenario where the two controllers perform equally, the K-MPC controller has the benefit of being completely *data-driven* and requiring only *output measurements*, whereas the L-MPC controller requires a model (to compute the local linearization) and full state measurements. In addition, the average computation time⁸ required to evaluate the control input of the K-MPC controller was 6.86 ms (including the evaluation of the lifting mapping $\psi(\xi)$), as opposed to 103 ms for the

L-MPC controller. This discrepancy is due to the fact that the local linearization and all data defining the underlying optimization problem that depend on it have to be re-computed at every iteration, which is costly on its own and also precludes efficient warm-starting; on the other hand, all data (except for the initial condition) of the underlying optimization problem of K-MPC are precomputed offline. In both cases, the computation times could be significantly reduced with a more efficient implementation. However, we believe, that the proposed approach would still be superior in terms of computational speed.

8.3. Nonlinear PDE control

In order to demonstrate the scalability and versatility of the approach, we use it to control the nonlinear Korteweg–de Vries (KdV) equation modeling the propagation of acoustic waves in a plasma or shallow-water waves (Miura, 1976). The equation reads

$$\frac{\partial y(t, x)}{\partial t} + y(t, x) \frac{\partial y(t, x)}{\partial x} + \frac{\partial^3 y(t, x)}{\partial x^3} = u(t, x),$$

where $y(t, x)$ is the unknown function and $u(t, x)$ the control input. We consider a periodic boundary condition on the spatial variable $x \in [-\pi, \pi]$. The nonlinear PDE is discretized using the split-stepping method with spatial mesh of 128 points and time discretization of $\Delta t = 0.01$ s, resulting in a computational state-space of dimension $n = 128$. The control input u is considered to be of the form $u(t, x) = \sum_{i=1}^3 u_i(t) v_i(x)$, where the coefficients $u_i(t)$ are to be determined by the controller and v_i are fixed spatial profiles given by $v_i(x) = e^{-25(x-c_i)^2}$ with $c_1 = -\pi/2$, $c_2 = 0$, $c_3 = \pi/2$. The control inputs are constrained to $u_i(t) \in [-1, 1]$. The lifting-based predictors are constructed from data in the form of 1000 trajectories of length 200 samples. The initial conditions of the trajectories are random convex combinations of three fixed spatial profiles given by $y_0^1 = e^{-(x-\pi/2)^2}$, $y_0^2 = -\sin(x/2)^2$, $y_0^3 = e^{-(x+\pi/2)^2}$; the control inputs $u_i(t)$ are distributed uniformly in $[-1, 1]^3$. The lifting mapping ψ is composed of the state itself, the elementwise square of the state, the elementwise product of the state with its periodic shift and the constant function, resulting in the dimension of the lifted state $N = 3 \cdot 128 + 1 = 385$. The control goal is to track a constant-in-space reference that varies in time in a piecewise constant manner. In order to do so we design the lifting-based Koopman MPC (23) with the reference tracking objective (46) with $Q = Q_{N_p} = I$, $R = 0$, $C = [I_{128}, 0]$ and prediction horizon $N_p = 10$ (i.e., 0.1 s). The results are depicted in Fig. 5; we observe a fast and accurate tracking of the reference profile. The average computation time to evaluate the control input was 0.28 ms (using the dense

⁷ Infeasibility of the underlying optimization problem is a common problem encountered in predictive control with various heuristic (e.g., soft constraints) or theoretically substantiated (e.g., set invariance) approaches trying to address them. See, e.g., Grüne and Pannek (2011) and Rawlings and Mayne (2009) for more details.

⁸ The optimization problems were solved by qpOASES (Ferreau et al., 2014) running on Matlab and 2 GHz Intel Core i7 with 8 GB RAM.

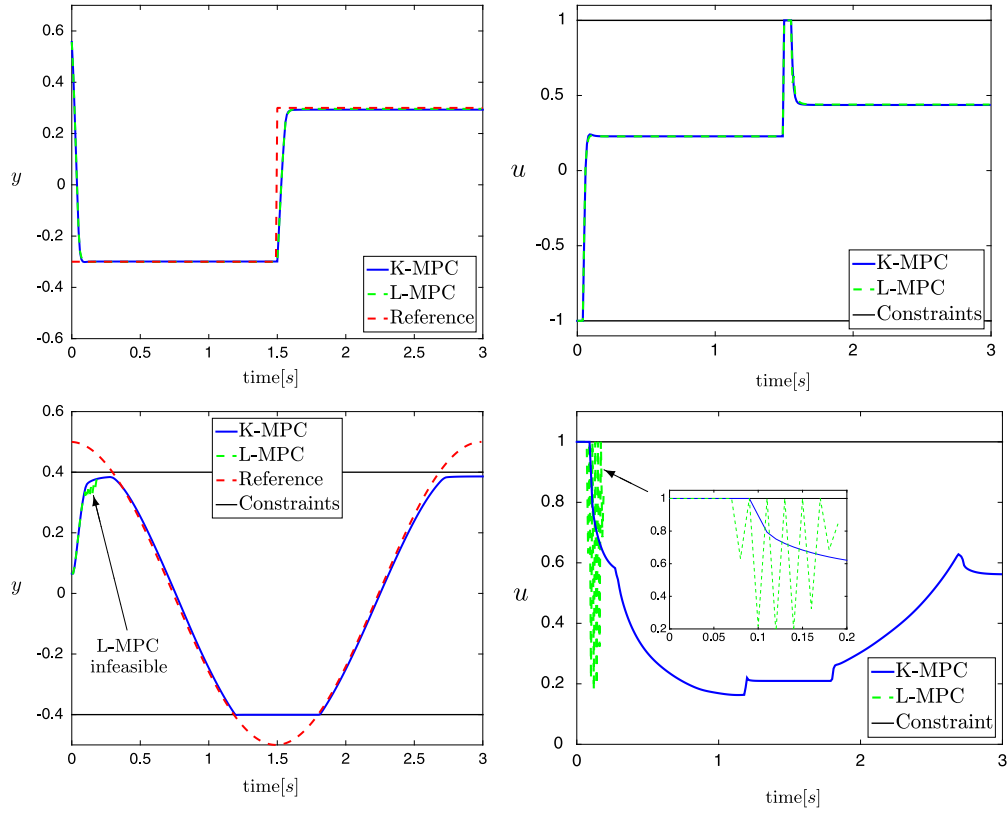


Fig. 4. Feedback control of a bilinear motor – reference tracking for a bilinear model of a DC motor. Top: piecewise constant reference, $x_0 = [0, 0.6]^\top$, no state constraints. Right: time-varying reference, $x_0 = [-0.1, 0.1]^\top$, constraints on the output imposed.

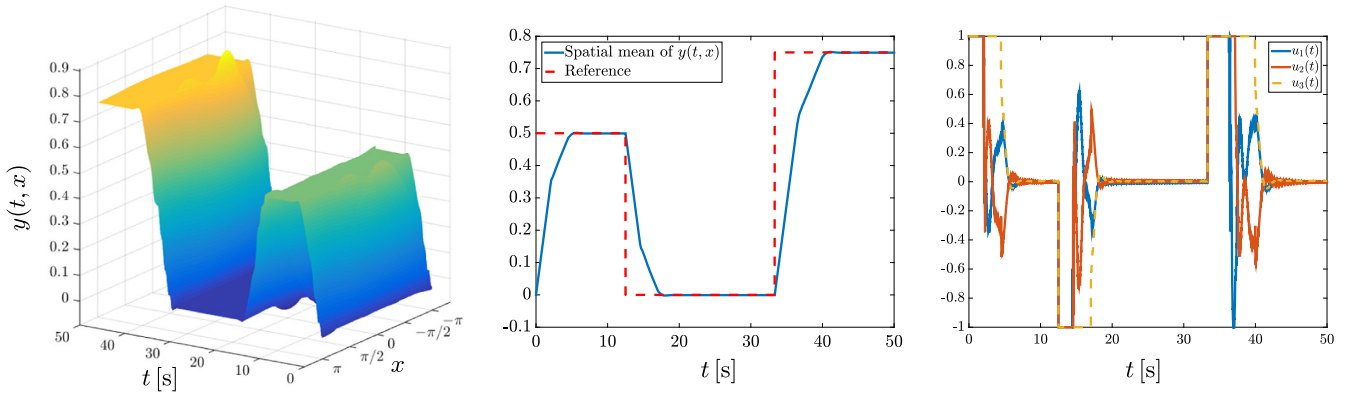


Fig. 5. Nonlinear PDE control – Tracking of a time-varying constant-in-space reference profile for the Korteweg–de Vries equation. Left: closed-loop solution. Middle: spatial mean of the solution. Right: control inputs.

form (24) and the hardware configuration described in Footnote ⁸, allowing for deployment in real-time applications requiring very fast sampling rates.

9. Conclusion and outlook

In this paper, we described a class of linear predictors for nonlinear controlled dynamical systems building on the Koopman operator framework. The underlying idea is to lift the nonlinear dynamics to a higher dimensional space where its evolution is approximately linear. The predictors exhibit superior performance on the numerical examples tested and can be readily used for feedback control design using linear control design methods. In particular, linear model predictive control (MPC) can be readily used to design controllers for the nonlinear dynamical system

without resorting to non-linear numerical optimization schemes. Linear inequality constraints on the states and control inputs as well as nonlinear constraints on the states can be imposed in a linear fashion; in addition cost functions nonlinear in the state can be handled in a linear fashion as well. Computational complexity of the underlying optimization problem is comparable to that of an MPC problem for a linear dynamical system of the same size. This is achieved by using the so-called dense form of an MPC problem whose computational complexity depends only on the number of control inputs and is virtually independent of the number of states. Importantly, the entire control design procedure is data-driven, requiring only input–output measurements.

Future work should focus on imposing or proving closed-loop guarantees (e.g., stability or degree of suboptimality) of the controller designed using the presented methodology and on optimal

selection of the lifting functions given some prior information on the dynamical system at hand.

Acknowledgments

The first author would like to thank Colin N. Jones for initial discussions on the topic and for comments on the manuscript, as well as to the three anonymous referees for their constructive comments that helped improve the manuscript. The authors would also like to thank Péter Koltai for bringing to their attention (Klus et al., 2016).

This research was supported in part by the ARO-MURI grant W911NF-14-1-0359 and the DARPA grant HR0011-16-C-0116. The research of M. Korda was supported by the Swiss National Science Foundation under grant P2ELP2_165166.

Appendix

This appendix expresses explicitly the matrices in the “dense-form” MPC problem (24) as a function of the data defining the “sparse-form” MPC problem (23). The matrices are given by

$$H = \mathbf{R} + \mathbf{B}^\top \mathbf{Q} \mathbf{B}, \quad h = r + \mathbf{B}^\top \mathbf{q}, \quad G = 2\mathbf{A}^\top \mathbf{Q} \mathbf{B},$$

$$L = \mathbf{F} + \mathbf{E} \mathbf{B}, \quad M = \mathbf{E} \mathbf{A}, \quad c = [b_0^\top, \dots, b_{N_p}^\top]^\top,$$

where

$$\mathbf{A} = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p-1}B & \dots & AB & B \end{bmatrix},$$

$$\mathbf{F} = \begin{bmatrix} F_0 & 0 & \dots & 0 \\ 0 & F_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & F_{N_p-1} \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

$$\mathbf{Q} = \text{diag}(Q_0, \dots, Q_{N_p}), \quad \mathbf{R} = \text{diag}(R_0, \dots, R_{N_p-1}),$$

$$\mathbf{E} = \text{diag}(E_0, \dots, E_{N_p}), \quad \mathbf{q} = [q_0^\top, \dots, q_{N_p}^\top]^\top, \quad \mathbf{r} = [r_0^\top, \dots, r_{N_p-1}^\top]^\top,$$

with $\text{diag}(\cdot, \dots, \cdot)$ denoting a block-diagonal matrix composed of the arguments.

References

- Brunton, S. L., Brunton, B. W., Proctor, J. L., Kaiser, E., & Kutz, J. N. (2017). Chaos as an intermittently forced linear system. *Nature communications*, 8(1), 1.
- Brunton, S. L., Brunton, B. W., Proctor, J. L., & Kutz, J. N. (2016). Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLoS One*, 11(2), e0150171.
- Budišić, M., Mohr, R., & Mezić, I. (2012). Applied koopmanism. *Chaos. An Interdisciplinary Journal of Nonlinear Science*, 22(4), 047510.
- Carleman, T. (1932). Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires. *Acta Mathematica*, 59(1), 63–87.
- Daniel-Berhe, S., & Unbehauen, H. (1998). Experimental physical parameter estimation of a thyristor driven DC-motor using the HMF-method. *Control Engineering Practice*, 6(5), 615–626.
- Ferreau, H. J., Kirches, C., Potschka, A., Bock, H. G., & Diehl, M. (2014). qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4), 327–363.
- Grüne, L., & Pannek, J. (2011). Nonlinear model predictive control. In *Nonlinear model predictive control*. Springer.
- Klus, S., Koltai, P., & Schütte, C. (2016). On the numerical approximation of the Perron-Frobenius and Koopman operator. *Journal of Computational Dynamics*, 3(1), 51–79.

- Koopman, B. (1931). Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5), 315–318.
- Koopman, B., & von Neuman, J. (1932). Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences of the United States of America*, 18(3), 255–263.
- Korda, M., & Mezić, I. (2018). *Journal of Nonlinear Science*, 28(2), 687–710.
- Ljung, L. (1998). System identification. In *Signal analysis and prediction* (pp. 163–173). Springer.
- Mauroy, A., & Goncalves, J. (2016). Linear identification of nonlinear systems: A lifting technique based on the Koopman operator. In *Conference on decision and control*.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.
- Mezić, I. (2005). Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1–3), 309–325.
- Mezić, I., & Banaszuk, A. (2004). Comparison of systems with complex behavior. *Physica D*, 197(1–2), 101–133.
- Miura, R. M. (1976). The Korteweg–deVries equation: A survey of results. *SIAM Review*, 18(3), 412–459.
- Proctor, J. L., Brunton, S. L., & Kutz, J. N. (2016). Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1), 142–161.
- Proctor, J. L., Brunton, S. L., & Kutz, J. N. (2018). Generalizing Koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, in press. ArXiv preprint arXiv:1602.07647.
- Rawlings, J. B., & Mayne, D. Q. (2009). *Model predictive control: Theory and design* (1st ed.). Nob Hill Publishing.
- Surana, A. (2016). Koopman operator based observer synthesis for control-affine nonlinear systems. In *Conference on decision and control*.
- Surana, A., & Banaszuk, A. (2016). Linear observer synthesis for nonlinear systems using Koopman operator framework. In *IFAC symposium on nonlinear control systems*.
- Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., & Kutz, J. N. (2014). On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2), 391–421.
- Williams, M. O., Hemati, M. S., Dawson, S. T. M., Kevrekidis, I. G., & Rowley, C. W. (2016). Extending data-driven Koopman analysis to actuated systems. In *IFAC symposium on nonlinear control systems (NOLCOS)*.
- Williams, M. O., Kevrekidis, I. G., & Rowley, C. W. (2015). A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6), 1307–1346.
- Williams, M. O., Rowley, C. W., & Kevrekidis, I. G. (2015). A kernel-based method for data-driven Koopman spectral analysis. *Journal of Computational Dynamics*, 2(2), 247–265.



Milan Korda received the Bachelor's degree in electrical engineering and the Master's degree in cybernetics and robotics from the Czech Technical University in Prague in 2009 and 2011, respectively. In 2016 he received the Ph.D. degree for his work on polynomial optimization and model predictive control at the Automatic Control Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. Currently he is an SNSF postdoctoral fellow at the University of California, Santa Barbara (UCSB).



Igor Mezić works on operator-theoretic methods in nonlinear dynamical systems and control theory and their applications in fluid dynamics, energy efficient design and operations and complex systems dynamics. He did his Dipl. Ing. in Mechanical Engineering in 1990 at the University of Rijeka, Croatia and his Ph.D. in Applied Mechanics at the California Institute of Technology. Dr. Mezić was a postdoctoral researcher at the Mathematics Institute, University of Warwick, UK in 1994–95. From 1995 to 1999 he was a member of Mechanical Engineering Department at the University of California, Santa Barbara where he is currently a Professor. In 2000–2001 he has worked as an Associate Professor at Harvard University in the Division of Engineering and Applied Sciences. He won the Alfred P. Sloan Fellowship, NSF CAREER Award from NSF and the George S. Axelby Outstanding Paper Award on “Control of Mixing” from IEEE. He also won the United Technologies Senior Vice President award for Science and Technology Special Achievement Prize in 2007. He was an Editor of *Physica D: Nonlinear Phenomena* and an Associate Editor of the *Journal of Applied Mechanics* and *SIAM Journal on Control and Optimization* and is a Fellow of the American Physical Society and SIAM. Dr. Mezić is the Director of the Center for Energy Efficient Design and Head of Buildings and Design Solutions Group at the Institute for Energy Efficiency at the University of California, Santa Barbara.