Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-40 (2022) 199–204

Learning Nonlinear Model Predictive Controllers and Virtual Sensors with Koopman Operators

Sergio Vanegas * **Fredy Ruiz** **

* *Dipartimento di Matematica, Politecnico di Milano, Milan, Italy (e-mail: sergiomauricio.vanegas@mail.polimi.it).*
** *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy (e-mail: fredy.ruiz@polimi.it).*

Abstract: Model Predictive Control is an industry-standard technique used to drive systems based on their internal dynamics. When not all states are available for feedback, a state estimator, such as an Extended Kalman Filter, is employed to achieve control over the complete system state. Nevertheless, when the system under control is nonlinear, these two combined methods can result in a computationally heavy control strategy, raising significantly the cost of implementing it online. In this paper, a data-driven strategy based on the Koopman Operator theory is presented to identify and replicate the dynamics of the Kalman Filter plus Model Predictive Controller pair in a resource-efficient scheme. First, a closed-loop operation data-set is generated from a pre-calibrated reference controller; then, a finite-dimensional approximation is derived for the Koopman Operator of the filter plus controller dynamics in the lifted space of observables; finally, the stability of the identified controller is evaluated through closed-loop simulations; in case the desired response has not been achieved, the identification process is performed iteratively with a progressively increasing regularization coefficient. A simulated example applied to the Van der Pol oscillator is presented to illustrate the effectiveness of the approach.

Keywords: Data-driven, Extended Kalman filter, Koopman Operator, Nonlinear systems, Model Predictive Control

1. INTRODUCTION

The *Koopman Operator* (KO) theory provides a convenient framework to represent finite dimensional nonlinear systems through infinite dimensional linear models, by transforming the state space into a infinite linear space of observables (Budišić et al. (2012)). In contrast to conventional linearization methods that depend on local derivatives, it provides an exact description (at least in theory) of the system's dynamics and, more importantly from an applied control point of view, it is easily adaptable to a fully data-driven pipeline (gray-box framework; see Huang (2020) for an in-depth exploration of data-driven modeling through the KO). The original theory was first developed in the 1930's through the work of Koopman (1931), but it was not until decades later when Korda and Mézic(see Mezić (2005); Korda and Mezić (2018)) revived this technique, first making use of it as a spectral analysis method and later on as a data-driven system identification framework for black and gray-box systems, which will be the focus of this paper.

On the other hand, there is *Model Predictive Control* (MPC): a mature framework which consists on an iterative approach to dynamical systems control, solving an optimization problem over a prediction horizon for the explicit model of the controlled system and determining the optimal driver signal that minimizes an adequate cost function. In a situation where not all the states are measured, an observer or virtual sensor (e.g., *Extended Kalman Filter*) is employed to recover the full state information. This process is repeated at each sample time, with up to date information about the state of the process (Camacho and Alba (2013)). The main disadvantage of this approach should be evident: for a nonlinear system, a complex, nonlinear, and non-convex optimization problem has to be solved at each step. Some machine-learning techniques have been used to attenuate this problem with relative success (Hertneck et al. (2018)), mainly using deep-learning techniques such as Graph Neural-Networks (Gama et al. (2022)) or Bayesian Neural-Networks (Chen et al. (2021)), meaning that a data-driven approach for MPC replication has already been tested and deemed a valid alternative to address the complexity of the original controller, approximating the static map from the optimal control problem solved at each step (without considering the filtering step).

Until now, the KO has been used alongside MPC to learn the dynamics of the plant in open-loop and use the resulting lifted-model as an internal model to predict future states (Korda and Mezić (2018); Zhang et al. (2022)), to an accuracy level where the extension of the model to continuous-time can be used in a Lyapunov-based Model Predictive Control (LMPC) framework as a virtual sensor in black-box scenarios (Narasingam et al. (2022)). Nevertheless, if a full-state MPC is required,

10.1016/j.ifacol.2023.01.072

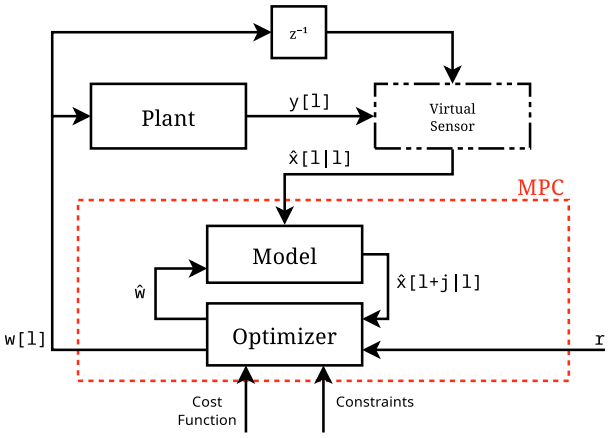

Fig. 1. State-aware MPC

whatever performance is gained through the substitution of the nonlinear model with the semi-linear Koopman dynamics is immediately overshadowed by the integration of a virtual sensor into the control loop. Fortunately, KO theory is applicable to any dynamical system, and noticing that the entire controller topology (Virtual Sensor + MPC) is a single nonlinear system, its dynamics can be represented as the linear combination of a library of observables. This paper proposes to use the KO as a technique to learn the internal dynamics of a pre-existing MPC controller working in conjunction with a state estimator by extending the available information over a lifted space of observables of the estimated state values. Since the coefficients required to evolve the output of the controller are calculated *offline*, the Koopman interpretation of the controller becomes significantly more efficient, making it cheaper to implement on lower-end hardware. The contributions of this paper are:

- A KO formulation of the MPC + Virtual Sensor dynamics-learning algorithm
- An efficient real-time implementation of the Koopman-identified Kalman Filter-informed MPC

The methodology was tested on a nonlinear second order system, the *forced Van der Pol oscillator*, comparing its accuracy and execution time against a reference nonlinear MPC and a Koopman-informed linear MPC.

2. PROBLEM FORMULATION

Consider a discrete-time dynamical system with sampling time T_s, representing the plant to be controlled, with state $\mathbf{x}[l] \in X \subseteq \mathbb{R}^{n_x}$ and dynamics described by

$$\begin{cases} \mathbf{x}[l+1] = \mathbf{f}_\mathbf{x}(\mathbf{x}[l], \mathbf{w}[l]) + \eta_\mathbf{x}[l] \\ \mathbf{y}[l] = \mathbf{f}_\mathbf{y}(\mathbf{x}[l]) + \eta_\mathbf{y}[l] \end{cases}, \tag{1}$$

where l is time (i.e., sample number), $\mathbf{x} \in X$ are the plant's states, $\mathbf{w} \in \mathbb{R}^{n_w}$ are the input signals of the **plant**, $\mathbf{y} \in \mathbb{R}^{n_y}$ are the measured output signals, and $\eta_\mathbf{x} \in \mathbb{R}^{n_x}$, $\eta_\mathbf{y} \in \mathbb{R}^{n_y}$ are random processes associated to the *process* and *measurement* noise respectively. $\mathbf{f}_\mathbf{x} : X \times \mathbb{R}^{n_w} \mapsto X$ is (assumed to be) a nonlinear *state propagator* corresponding to the smooth deterministic motion that would be expected if the random disturbances due to process noise ($\eta_\mathbf{x}$) were absent. A state-aware MPC strategy (Maiworm et al. (2021)) is integrated in closed-loop as in Fig. 1 [1], where the *Optimizer* tries to minimize the expression in

$$\begin{aligned} J(\mathbf{x}, \mathbf{w}; H_x, H_w, \mathbf{r})[l] = \\ \sum_{j=1}^{H_x} \sum_{i=1}^{n_x} \delta_{j,i} \left[\hat{x}_i[l+j|l] - r_i[l+j]\right]^2 \\ + \sum_{j=1}^{H_w} \sum_{i=1}^{n_w} \lambda_{j,i} \left[\hat{w}_i[l+j-1] - \hat{w}_i[l+j-2]\right]^2 \end{aligned} \tag{2}$$

$$\begin{aligned} \textbf{s.t. } & \forall j \in 1, \dots, H_x : \\ & \hat{\mathbf{x}}[l+j|l] = \mathbf{f}_\mathbf{x}(\hat{\mathbf{x}}[l+j-1|l], \hat{\mathbf{w}}[l+j-1|l]) \\ & \hat{\mathbf{x}}[l|l] = \mathbf{x}[l] \\ & \hat{\mathbf{w}}[l+j|l] \in \mathcal{W} \quad , \quad \hat{\mathbf{x}}[l+j|l] \in \mathcal{X} \end{aligned} \tag{3}$$

subjected to the following set of control signal and state value/rate parameters:

- Prediction horizon ($H_x \in \mathbb{N}$)
- State weights ($\delta_{j,i} \in \mathbb{R}_+$)
- Control horizon ($H_w < H_x \in \mathbb{N}$)
- Control-rate weights ($\lambda_{j,i} \in \mathbb{R}_+$)
- State constraints ($\mathcal{X} \subseteq X$)
- Control constraints ($\mathcal{W} \subseteq \mathbb{R}^{n_w}$)

The current internal states $\mathbf{x}[l]$ are in turn measured through a virtual sensor, the *Extended Kalman Filter* (EKF), which yields an estimate based on the measured output $\mathbf{y}[l]$. In its most standard form, given a nonlinear system of the form in (1), the EKF is able to recover the internal states of the system from the input and the measured output; i.e.,

$$\hat{\mathbf{x}}[l] = \mathbf{F}_{\text{EKF}}(\hat{\mathbf{x}}[l-1], \mathbf{y}[l], \mathbf{w}[l-1], P[l]) \tag{4}$$

$$P[l] = \mathbf{\Psi}_{\text{EKF}}(P[l-1], \mathbf{x}[l-1]), \tag{5}$$

where $\hat{\mathbf{x}}[l]$is the state estimate at time l and $P[l]$ is the estimated covariance matrix of the state estimation error.

It is worth mentioning that the EKF must be fully informed of the internal dynamics of the system (or, at least, must be equipped with a good approximation). For further details on the specifics of the EKF, see Crassidis and Junkins (2004). The problem with this solution is its inherent linearization: since the EKF extends the Kalman filtering theory for linear systems through a local linearization, it generates a trade-off situation between accuracy and complexity.

3. THE KOOPMAN OPERATOR

We follow the infinite-dimensional formulation of the KO from Arbabi and Mezic (2017) and Budišić et al. (2012). Let $g : X \mapsto \mathbb{C}$ be an *arbitrary* (at least for now) complex-valued function; g is called an *observable* of the system (autonomous for now)

$$\begin{cases} \mathbf{x}[l+1] = \mathbf{f}_\mathbf{x}(\bar{\mathbf{x}}[l]) + \eta_\mathbf{x}[l+1] \\ \mathbf{y}[l] = \mathbf{f}_\mathbf{y}(\bar{\mathbf{x}}[l]) + \eta_\mathbf{y}[l] \end{cases}, \tag{6}$$

with

$$\mathbf{x}[l] = \bar{\mathbf{x}}[l] + \eta_\mathbf{x}[l], \tag{7}$$

[1] In practice, past control actions are externally fed-back to the MPC in case the applied control action is different from the optimal one (e.g., because of measured non-modeled actuator effects).

where $\bar{\mathbf{x}}$ is the real value of the state, and whose value observed over a single step starting from $\bar{\mathbf{x}}[l]$ evolves in time according to the map

$$g\left(\bar{\mathbf{x}}[l+1]\right) = g \circ \mathbf{f}_{\mathbf{x}}\left(\bar{\mathbf{x}}[l]\right). \tag{8}$$

The space $\mathcal{G}$ of all observables g is a linear functional space, and thus a linear operator $K : \mathcal{G} \mapsto \mathcal{G}$ (hereinafter referred to as the *Koopman Operator*) can be defined through

$$g\left(\mathbf{x}[l+1]\right) = K \circ g\left(\mathbf{x}[l]\right), \tag{9}$$

meaning that the dynamics of an arbitrary nonlinear system can be linearly replicated when expanded over an infinite-dimensional space.

Now that the theoretical framework has been laid, the next step is describing the data-driven algorithm for the finite-dimensional KO approximation (Mauroy and Goncalves (2020)). Consider L snapshot pairs $(\mathbf{x}[l], \mathbf{x}[l+1])$ obtained from noisy measurements, and also

$$\begin{aligned}\mathbf{x}[l+1] &= \mathbf{f}_{\mathbf{x}}\left(\mathbf{x}[l] - \eta_{\mathbf{x}}[l]\right) + \eta_{\mathbf{x}}[l+1] \\ &= \bar{\mathbf{x}}[l+1] + \eta_{\mathbf{x}}[l+1],\end{aligned} \tag{10}$$

which is associated to the dynamical system described in Eq. (6). The controller-learning algorithm can be split into two main stages as follows:

Dimensional lifting: At this stage, the snapshot pairs $(\mathbf{x}[l], \mathbf{x}[l+1])$ described in (10) are lifted to the space of observables by constructing new pairs of the form $(\mathbf{g}\left(\mathbf{x}[l]\right), \mathbf{g}\left(\mathbf{x}[l+1]\right))$ for some $\mathbf{g} = \{g_j\}_{j=1}^{\infty} \subset \mathcal{G}$. The relation

$$\begin{aligned}\mathbf{g}\left(\mathbf{x}[l+1]\right) &= \mathbf{g}\left(\mathbf{f}_{\mathbf{x}}\left(\mathbf{x}[l] - \eta_{\mathbf{x}}[l]\right) + \eta_{\mathbf{x}}[l+1]\right) \\ &\approx K\mathbf{g}\left(\mathbf{x}[l]\right) + \mathcal{O}\left(||\eta_{\mathbf{x}}||\right)\end{aligned} \tag{11}$$

follows from (7), (10), and the linearity of the KO.

The selection of the observable functions can be completely arbitrary or even physically-informed, granted an infinite collection of linearly independent observables can be generated from its structure. For the sake of generality, the algorithm is presented using a collection of thin-plate spline basis functions of the form

$$g_j(\mathbf{x}) = ||\mathbf{x} - \mathbf{z}_j||^2 \log\left(\mathbf{x} - \mathbf{z}_j\right), \tag{12}$$

where $\mathbf{z}_j \in \mathbb{R}^n$ is a (randomly) pre-selected centroid. These are not only generic enough so that they can be easily applied to an arbitrary system, but also facilitate order scalability and online implementation.

Matrix identification: At this stage, a truncated finite-dimensional projection of the KO is obtained through a regularized error minimization problem, similar to the Extended DMD algorithm in Williams et al. (2015), but blinded against measurement noise using the Frobenius-norm of the identified operator.

Let $\mathcal{G}_N \subset \mathcal{G}$ be the linear subspace of observables generated by the original states and the basis $\{g_j\}_{j=1}^{n_g}$ of thin-plate spline functions. For each snapshot pair $(\mathbf{x}[l], \mathbf{x}[l+1]) \in \mathbb{R}^{n_x \times 2}$, $l \in \{1, \dots, L\}$, a new pair

$$\left(\tilde{\mathbf{g}}\left(\mathbf{x}[l]\right), \tilde{\mathbf{g}}\left(\mathbf{x}[l+1]\right)\right) \in \mathbb{R}^{N \times 2} \tag{13}$$

$$\tilde{\mathbf{g}}\left(\mathbf{x}[l]\right) = \left[\mathbf{x}[l] \;\; \mathbf{g}\left(\mathbf{x}[l]\right)\right]^T \in \mathbb{R}^N \tag{14}$$

is constructed, where $\mathbf{g}\left(\mathbf{x}\right) = \left(g_1\left(\mathbf{x}\right), \dots g_N\left(\mathbf{x}\right)\right)^T$ denotes the vector of basis functions following the definition in (12) and $N \coloneqq n_x + n_g$.

An identification of the KO K is now executed; specifically, the finite-rank truncation $K_N : \mathcal{G}_N \mapsto \mathcal{G}_N$ is identified. To this end, a regularized least-squares fit for the orthogonal projection of the *identity* observable dynamics is performed as proposed in Zhang et al. (2022). The definition of the optimization problem per observable is

$$\begin{aligned}\left(\overline{\mathbf{K}}_N\right)_i = \arg\min_{\mathbf{k} \in \mathbb{R}^N} \frac{1}{L} \sum_{l=1}^{L} |\mathbf{k}^T \tilde{\mathbf{g}}(\mathbf{x}[l]) - \left(\mathbf{x}[l+1]\right)_i|^2 \\ + \frac{\alpha}{N} ||\mathbf{k}||_F^2,\end{aligned} \tag{15}$$

where $i \in \{1, \dots, N\}$ corresponds to each row from the matrix representation of the truncated KO and $\alpha \geq 0$ defines the regularization coefficient of the Frobenius norm term. Thus, the dynamics of the original states can now be recovered through

$$\mathbf{x}[l+1] = \mathbf{A}\tilde{\mathbf{g}}\left(\mathbf{x}[l]\right), \tag{16}$$

where $\mathbf{A} = \left(\overline{\mathbf{K}}_N\right)_{1:n_x}$. It is worth noting that this is not the traditional definition of the Koopman snapshots since, for control applications, approximating only the dynamics of the original states as a linear combination of the elements in the lifted space and then re-observing the predicted states at each sample yields a more robust implementation with relatively low overhead. More complex correction strategies for plant-model mismatch caused by the truncation of the KO have been explored in the literature, such as disturbance-estimator-based approaches for LMPC (Son et al. (2020)); nevertheless, for the sake of simplicity, the re-observation approach was preferred in this paper.

For a non-autonomous system, the definition of the objective function can be easily extended by concatenating the observable library with the recorded exogenous input (see Mauroy and Goncalves (2020)) and solving

$$\begin{aligned}\left([\mathbf{A} \quad \mathbf{B}]\right)_i = \\ \arg\min_{\mathbf{a} \in \mathbb{R}^N, \mathbf{b} \in \mathbb{R}^{n_u}} \frac{1}{L} \sum_{l=1}^{L} |\mathbf{a}^T \tilde{\mathbf{g}}(\mathbf{x}[l]) + \mathbf{b}^T \mathbf{u}[l] - \left(\mathbf{x}[l+1]\right)_i|^2 \\ + \frac{\alpha}{N}(||\mathbf{a}||_F^2 + ||\mathbf{b}||_F^2)\end{aligned} \tag{17}$$

instead, where $\mathbf{u} \in \mathbb{R}^{n_u}$ are the input signals of the system [2], and then splitting the coefficients accordingly to recover the dynamics just as in

$$\mathbf{x}[l+1] = \mathbf{A}\tilde{\mathbf{g}}\left(\mathbf{x}[l]\right) + \mathbf{B}\mathbf{u}[l], \tag{18}$$

where $\mathbf{B} \in \mathbb{R}^{n \times n_u}$ is the collection of newly obtained columns.

4. LEARNING MPC CONTROL LAWS

The formulation is as follows:

- The Kalman filter estimate at sample l is taken as state for the MPC control law, defining n_x.
- The reference states at sample l, the measured plant output at sample l, and the control signal sent to the plant at sample $l-1$ are taken as inputs, defining

$$\mathbf{u}[l] = \left[\mathbf{r}[l] \;\; \mathbf{y}[l] \;\; \mathbf{w}[l-1]\right]^T \in \mathbb{R}^{n_u}, \tag{19}$$

$$n_u = n_x + n_y + n_w \tag{20}$$

[2] A different notation for the input is used because the target system to identify in this paper is not the plant, but the EKF-MPC pair

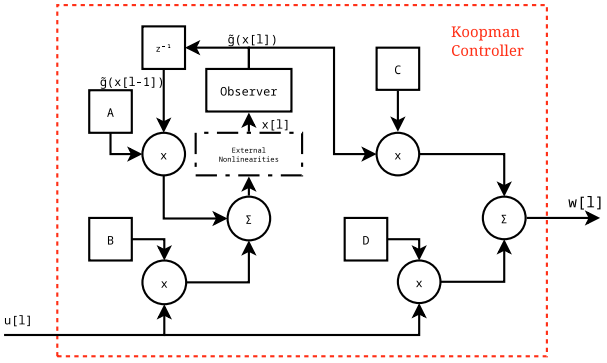

Fig. 2. Koopman-identified EKF-informed MPC

- The minimization problem

$$([\mathbf{C} \quad \mathbf{D}])_i = \arg\min_{\mathbf{c}\in\mathbb{R}^N, \mathbf{d}\in\mathbb{R}^{n_u}} \frac{1}{L}\sum_{l=1}^{L} |\mathbf{c}^T\tilde{\mathbf{g}}(\mathbf{x}[l]) + \mathbf{d}^T\mathbf{u}[l] - (\mathbf{w}[l])_i|^2 + \frac{\alpha}{N}(||\mathbf{c}||_F^2 + ||\mathbf{d}||_F^2) \tag{21}$$

  yields the coefficients defining the control law at sample l, which in turn becomes the output of our identified controller.

The newly defined matrices $\mathbf{C} \in \mathbb{R}^{n_w \times N}$ and $\mathbf{D} \in \mathbb{R}^{n_w \times n_u}$, with $i \in \{1, ..., n_w\}$ can be used to recover the control signal, resulting in the *proposed KO controller topology*

$$\begin{cases} \mathbf{x}[l] = \mathbf{A}\tilde{\mathbf{g}}(\mathbf{x}[l-1]) + \mathbf{B}\mathbf{u}[l] \\ \mathbf{w}[l] = \mathbf{C}\tilde{\mathbf{g}}(\mathbf{x}[l]) + \mathbf{D}\mathbf{u}[l] \end{cases} \tag{22}$$

At this point, the similarity between the proposed schemes and a traditional State-Space representation is more than evident; the only differences being the nonlinear observation performed over each predicted state and the fact that the state values are being calculated for sample l instead of $l+1$, as shown in Fig. 2.

In order to generate the training data-set for the KO, an informative reference signal for the controlled states has to be given to the optimization based MPC; this signal must be rich enough to avoid identification bias. The proposed methodology is as follows:

(1) Generate a series of steps covering the entirety of the possible equilibrium states where the system could be found while online; make sure that the duration of each step will be enough for the system to stabilize while under the control of the EKF-informed MPC.
(2) Use the step signals as reference to simulate (or measure) the operation of the EKF-MPC control scheme.
(3) Record the optimal control signals and EKF-predicted states in sync with the above reference.

5. NUMERICAL EXAMPLE: THE VAN DER POL OSCILLATOR

A staple benchmark nonlinear system, the forced Van der Pol oscillator, taken from Korda and Mezić (2018), was used in order to evaluate the performance of this methodology [3]. Its dynamics are given by

$$\begin{cases} \dot{x}_1 &= 2x_2 \\ \dot{x}_2 &= -0.8x_1 + 2x_2 - 10x_1^2x_2 + w \\ y &= x1 \end{cases}, \tag{23}$$

from which we can extract the model parameters

$$(n_x, n_y, n_w) = (2, 1, 1) \tag{24}$$

The EKF was provided with a fractional-step forward-Euler discrete-time approximation of the Van der Pol oscillator. The process-noise and measurement-noise were assumed to be both equal in parameters and mutually independent: $\eta \sim \mathcal{N}(0, 0.001)$. The MPC was configured with the following parameters:

- Prediction horizon (H_x): 25 samples
- Control horizon (H_w): 5 samples
- State weights: $\delta_j = [1\ 0]^T \quad \forall j = 1, \ldots, N_y$
- Control weights: $\lambda_j = 0 \quad \forall j = 1, \ldots, N_u$
- Control bounds: $[-2, 2]$
- Second state bounds: $[-1, 1]$
- Prediction internal model: same as the EKF

Additionally, an equally configured linear MPC informed with a KO approximation ($n_g = 100$) of the Van der Pol oscillator (identified from open-loop simulations with a random input signal within the above control bounds) was designed in order to evaluate the performance of the technique proposed in this paper when compared to the literature's approach to MPC using KO. This implementation also involved lifting both the states estimated by the EKF and the reference, and then feeding these to the linear MPC.

The aforementioned nonlinear MPC was then tasked with making the plant's first state follow a series of steps, first going in ascending order through the interval $[-1, 1]$ with steps of amplitude 0.1 lasting for 5 s each, then repeating the sequence in descending order, and finally going through the same list of values in random order. The whole trajectory sums up to a total of 3.15×10^2 s, or 3.15×10^4 snapshots. Since the output weight corresponding to the second-state is equal to 0, the value of the second-state reference should not matter; nevertheless, it was approximated through a backwards finite difference as

$$x_2[l] = \frac{x_1[l] - x_1[l-1]}{2\,T_s} \tag{25}$$

(following the system's definition). The entire training data-set can be observed in Fig. 3.

These were then used to train the proposed topology of the Koopman-identified EKF-informed MPC using a collection of 10, 20, 50 and 100 observable functions. The centroids for each of these functions were randomly selected by fitting a Gaussian distribution to the values recorded for each state and then extracting the necessary samples according to the chosen library size. The regularization coefficient for each Koopman-identified controller was tuned (using the same magnitude for both (15) and (21)) in order to get the best possible step response (determined qualitatively at this stage), while keeping the system's

[3] All of the source code for the described implementation can be found in *https://github.com/sergiovaneg/Koopman*.

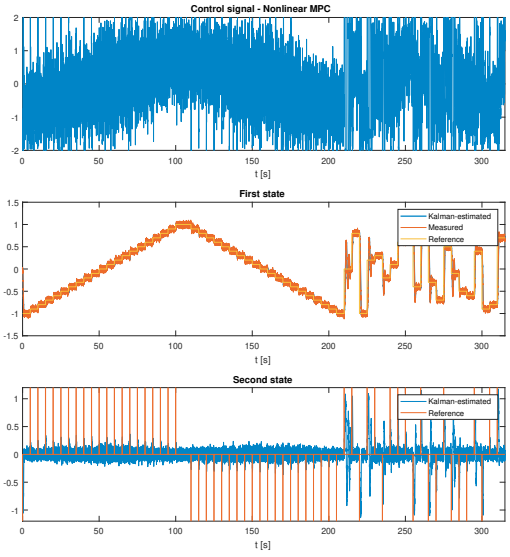

Fig. 3. Identification set

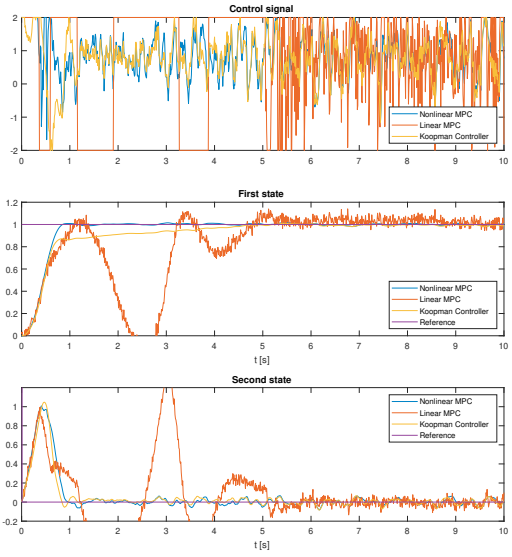

Fig. 4. Benchmark - Positive step reference

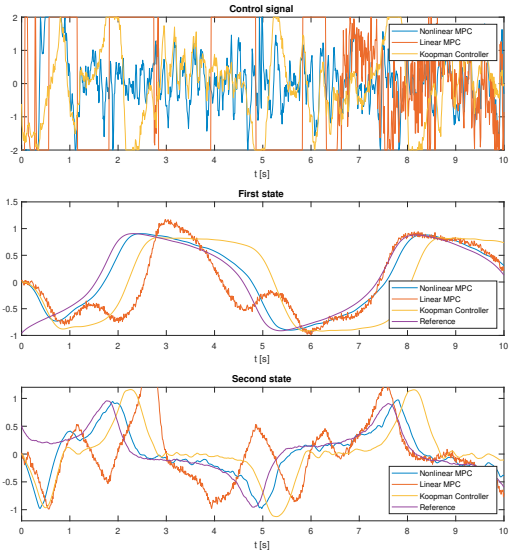

Fig. 5. Benchmark - Oscillating reference

eigenvalues within unitary magnitude [4]. All minimization problems were solved using the default options for CVX Research (Grant and Boyd (2014)) in MATLAB^{TM}.

Finally, the controllers' accuracy was numerically compared based on the Time-averaged Root Mean Square (TRMS) of the error between the reference and the internal states of the plant over the span of 1×10^2 s, while following a positive unitary step on the first state and an oscillating trajectory, generated by simulating the plant in open-loop with a random signal within the MPC's control bounds. The reference signals of the second state for both step references were calculated using (25).

Even though the reference MPC was designed to follow only the first-state reference, the error for both states, as well as the execution time for each topology and observable library size is reported in tables 1 and 2, split by the type of reference passed to the controller. In Figs. 4 and 5, the response of the system w.r.t. the reference is shown for the original MPC and the highest-observable-count Koopman-identified controllers.

Table 1. Positive step reference results

Controller	x_1 **error**	x_2 **error**	**Exec. time** $[s]$
nlMPC	5.927×10^{-2}	5.043×10^{-1}	5.391×10^{2}
lMPC	1.187×10^{-1}	5.130×10^{-1}	8.669
	Koopman controller		
10 obs.	8.060×10^{-2}	5.014×10^{-1}	4.278×10^{-1}
20 obs.	6.803×10^{-2}	5.046×10^{-1}	4.312×10^{-1}
50 obs.	6.327×10^{-2}	5.038×10^{-1}	4.357×10^{-1}
100 obs.	6.299×10^{-2}	5.039×10^{-1}	4.603×10^{-1}

The first thing that can be noticed is the performance increase in the execution time, decreasing by 3 orders of magnitude w.r.t. the nonlinear MPC and by one order of magnitude w.r.t. the Koopman-informed MPC. Moreover, the average norm of the error stays within the same order of magnitude, albeit with a noticeably slower step response (around 5 seconds as shown in Fig.4 in contrast with the nonlinear MPC's transition time of under a second). It is this very property that allows the Koopman-identified controller to have a higher measurement noise robustness, but it is also the responsible for the delayed response to the oscillating references in Fig.5. Nevertheless, the Koopman-identified controller succeeds in providing a step response that is far more stable than the Koopman-informed MPC, although suffering from the same difference in accuracy for oscillating references, as shown in Table 2.

Table 2. Oscillating reference results

Controller	x_1 **error**	x_2 **error**	**Exec. time** $[s]$
nlMPC	1.532×10^{-1}	1.840×10^{-1}	5.101×10^{2}
lMPC	1.937×10^{-1}	2.399×10^{-1}	8.292
	Koopman controller		
10 obs.	4.798×10^{-1}	3.854×10^{-1}	4.626×10^{-1}
20 obs.	6.254×10^{-1}	5.541×10^{-1}	4.621×10^{-1}
50 obs.	6.318×10^{-1}	5.565×10^{-1}	4.659×10^{-1}
100 obs.	6.201×10^{-1}	5.273×10^{-1}	6.020×10^{-1}

[4] In practice, the control bounds were found to be more effectively implemented as an external saturation. It is important to consider this external effect when performing the feedback of the control signal.d

Regarding the amount of observables, even though the higher-observable-count as-state controllers require some degree of regularization, the consistently increasing accuracy when augmenting the observable count is coherent with the Koopman theory of the exact dynamics' replication when approaching infinite observables. Nevertheless, the increase in performance is nonlinear, yielding a nearly identical response between the $n_g = 50$ and $n_g = 100$ as opposed to the jump in performance between $n_g = 10$ and $n_g = 50$. Notice also that the accuracy of the controller reduces when using $n_g = 20$, meaning that a library-size independence study should be performed before settling on a value for n_g.

6. CONCLUSIONS

As originally stated, the proposed scheme is able to successfully identify and replicate the dynamics of both an Extended Kalman Filter and a nonlinear MPC exclusively from simulation data. Furthermore, considering that the proposed observable structure does not depend on the internal model of the reference control scheme, but instead on statistical data extracted from the snapshots themselves, the technique described in this paper is proven to be model-agnostic and can be used to control any other nonlinear system, provided a pre-existing correctly calibrated EKF-MPC controller is available offline.

The main benefit from the Koopman-identified controller is, as stated from the beginning, the relatively low demand for resources of its online implementation, since the operations being performed per sample are limited to linear algebra and the *observation* procedure. As per memory, only a single sample has to be stored in order to apply the aforementioned operations. All of this makes it significantly cheaper to implement when compared to the original controller it is replicating.

It was observed that the control law is not perfectly replicated, yielding a slower response which can be explained through the limitations imposed for the eigenvalues. Also the generality of the observable library becomes a weak point as a large amount of functions (when compared to the original 2-state system) has to be used in order to approximate the controller dynamics. Moreover, since the proposed technique is fully data-driven, the entire pipeline depends on a correctly implemented reference controller (with sufficient knowledge of the internal dynamics of the plant). Lastly, in its current state the controller cannot be re-calibrated without having to perform another set of measurements on a new reference control loop. This lack of re-calibration capabilities also implies that there is no direct way of modifying the response time or over-shoot margin of the controller, so we are limited to modifying its properties during the training phase through the regularization parameter and library size. Nevertheless, some of these problems could be solved by using spectral analysis techniques, which further motivates the study of this approach.

REFERENCES

Arbabi, H. and Mezic, I. (2017). Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4), 2096–2126.

Budišić, M., Mohr, R., and Mezić, I. (2012). Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4), 047510.

Camacho, E.F. and Alba, C.B. (2013). *Model predictive control*. Springer science & business media.

Chen, H., Paoletti, N., Smolka, S.A., and Lin, S. (2021). Mpc-guided imitation learning of bayesian neural network policies for the artificial pancreas. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 2525–2532. IEEE.

Crassidis, J.L. and Junkins, J.L. (2004). *Optimal estimation of dynamic systems*. Chapman and Hall/CRC.

Gama, F., Li, Q., Tolstaya, E., Prorok, A., and Ribeiro, A. (2022). Synthesizing decentralized controllers with graph neural networks and imitation learning. *IEEE Transactions on Signal Processing*, 70, 1932–1946.

Grant, M. and Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`.

Hertneck, M., Köhler, J., Trimpe, S., and Allgöwer, F. (2018). Learning an approximate model predictive controller with guarantees. *IEEE Control Systems Letters*, 2(3), 543–548. doi:10.1109/LCSYS.2018.2843682.

Huang, B. (2020). *Data-driven modeling and control of dynamical systems using Koopman and Perron-Frobenius operators*. Ph.D. thesis, Iowa State University.

Koopman, B.O. (1931). Hamiltonian systems and transformation in hilbert space. *Proceedings of the national academy of sciences of the united states of america*, 17(5), 315.

Korda, M. and Mezić, I. (2018). Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93, 149–160.

Maiworm, M., Limon, D., and Findeisen, R. (2021). Online learning-based model predictive control with gaussian process models and stability guarantees. *International Journal of Robust and Nonlinear Control*, 31(18), 8785–8812.

Mauroy, A. and Goncalves, J. (2020). Koopman-based lifting techniques for nonlinear systems identification. *IEEE Transactions on Automatic Control*, 65(6), 2550–2565. doi:10.1109/TAC.2019.2941433.

Mezić, I. (2005). Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1), 309–325.

Narasingam, A., Son, S.H., and Kwon, J.S.I. (2022). Data-driven feedback stabilisation of nonlinear systems: Koopman-based model predictive control. *International Journal of Control*, 1–12.

Son, S.H., Narasingam, A., and Kwon, J.S.I. (2020). Handling plant-model mismatch in koopman lyapunov-based model predictive control via offset-free control framework. *arXiv preprint arXiv:2010.07239*.

Williams, M.O., Kevrekidis, I.G., and Rowley, C.W. (2015). A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6), 1307–1346.

Zhang, X., Pan, W., Scattolini, R., Yu, S., and Xu, X. (2022). Robust tube-based model predictive control with koopman operators. *Automatica*, 137, 110114.