Data-driven quasi-LPV Model Predictive Control
using Koopman Operator Techniques

Pablo. S. G. Cisneros * **Adwait Datar** * **Patrick Göttsch** *
Herbert Werner *

* *Institute of Control Systems, Hamburg University of Technology, Germany*
pablo.gonzalez@tuhh.de adwait.datar@tuhh.de
patrick.goettsch@tuhh.de h.werner@tuhh.de

Abstract:
A fast data-driven extension of the velocity-based quasi linear parameter varying model predictive control (qLMPC) approach is proposed for scenarios where first principles models are not available or are computationally too expensive. We use tools from the recently proposed Koopman operator framework to identify a quasi linear parameter varying model (in Input/Output and State-Space form) by choosing the observables from physical insight. We show good tracking performance in simulation and experiment on a complex strongly non-linear plant of a Control Moment Gyroscope. We further propose an online update strategy to adapt to changes in the plant dynamics. We compare the proposed approach with standard methods from literature and show improved control performance.

Keywords: Nonlinear predictive control, linear parameter-varying systems, data-driven control, Koopman operator, adaptive control

1. INTRODUCTION

Model Predictive Control (MPC) is one of the most widely used control strategies thanks to its versatility, easy extension to multiple input-multiple output systems, and capability to explicitly consider constraints. On the other hand, its main drawback is the complex computations (when compared to other strategies) entailed by the online solution of an optimization problem. This becomes a problem for complex nonlinear systems, in which case a non-convex optimization problem needs to be solved. To tackle this issue, several algorithms have been proposed which solve an approximate problem, enabling its implementation for system with fast dynamics. One such algorithm, referred to as *quasi-Linear Model Predictive Control* (qLMPC) has been proposed in (Cisneros et al., 2016), where a nonlinear system is modeled as a quasi-Linear Parameter Varying (qLPV) system. When compared to other state-of-the-art methods, e.g. Diehl et al. (2005), qLMPC shows to have comparable numerical and closed-loop performance (Cisneros and Werner (2017)). An extension of the qLMPC approach to plant models in input-output form was presented in (Gonzalez Cisneros and Werner, 2019a). An efficient method for obtaining a qLPV model via a velocity algorithm is proposed in (Gonzalez Cisneros and Werner, 2019b).

For scenarios where first principles models are not practical to be computed, or otherwise too complex for a meaningful implementation of the control law, data-driven techniques represent a promising alternative. One such line of research, relevant for this paper, is based on the work of Koopman (Koopman, 1931). These techniques are gaining considerable attention within the control (especially MPC) community, evident from a number of recent results such as in (Korda and Mezić, 2018a), (Abraham et al., 2017), (Kaiser et al., 2018), (Korda et al., 2018), (Arbabi et al., 2018), (Proctor et al., 2016), (Hanke et al., 2018) and others. A survey on Koopman operator theory can be found in

(Budišić et al., 2012). A survey oriented towards control can be found in (Kaiser et al., 2019). The idea at the core of most of this work is to approximate a nonlinear system with a higher dimensional lifted linear system such that the state predictions can be obtained by propagating a higher dimensional linear system forward in time. As a consequence, instead of a non-convex optimization problem, a quadratic program can be posed for the higher dimensional linear system. To keep the size of the optimization problem from growing with the dimension of the lifted space, it is often posed in the so-called *dense formulation* (for example in (Korda and Mezić, 2018a)). Extensions using reproducing kernel methods can be found in (Williams et al., 2014). Other data-driven approaches are e.g. (Kadali et al., 2001), where a a predictive control law is derived by data-driven subspace identification; this limits its applicability to linear systems.

We propose an alternative approach in this paper, building on the results in (Cisneros et al., 2016) and (Gonzalez Cisneros and Werner, 2019b), and using Koopman-based identification techniques simply to obtain a velocity-based qLPV model. The identification process can be seen as applying the extended Dynamic Mode Decomposition (eDMD) algorithm with the addition of recovering a qLPV model at the end. We present our results both for the case when the identified qLPV model is in state-space form, as well as for the case where it is in input-output form. We present experimental results on a complex nonlinear MIMO plant - a Control Moment Gyroscope (CMG) - to demonstrate the practicality of the proposed approach. We also present a comparison of the closed-loop performance with other methods in the literature, and a short discussion of computational complexity. By choosing appropriate observables *a priori* based on physical insight, we show that excellent control performance can be achieved with a relatively small number of observables. Motivated by the ideas behind recursive least squares estimation (Hsia, 1977), applied to the Dynamic Mode

Decomposition (DMD) algorithm in (Zhang et al., 2019) and (Peitz and Klus, 2018), we propose a slightly modified approach where the model is updated whenever novel dynamics are encountered. This differs from the approach of online learning presented in (Li et al., 2017) where the library of observables is updated online.

This paper is organized as follows: Section 2 gives a brief overview of the Koopman operator theory. In Section 3 the algorithm for computing an approximate Koopman operator is derived. Section 4 presents how to obtain a velocity-based state space and input-output model, respectively, that can be used in an qLMPC scheme, given an approximate Koopman operator. Section 5 briefly introduces the qLMPC scheme. Section 6 details the experimental implementation, results of the controller applied to a Control Moment Gyroscope and comparison to other methods from literature. We end in section 7 with conclusions and outlook.

Notation

The notation $Co(a,b)$ with $a,b \in \mathbb{R}^n$ is used to denote the convex hull obtained from a convex combination of a,b, i.e. $Co(a,b) = \lambda a + (1-\lambda)b, \lambda \in [0,1]$. The backward time-shift operator is denoted by q^{-1}.

2. KOOPMAN BASED IDENTIFICAITON OF QLPV MODEL

Consider a system whose dynamics are governed by the discrete-time non-linear model

$$x_{k+1} = f(x_k, u_k) \qquad (1)$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ and the map $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ propagates the state forward in time. The discrete-time dynamics can be embedded in a fictitious lifted system by defining a new state $z_k \in \mathbb{R}^{n_z}$ with $n_z = n+m$ as $z_k = [x_k^T \ u_k^T]^T$ This new fictitious system is defined by

$$z_{k+1} = \begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} f(x_k, u_k) \\ g(x_k, u_k) \end{bmatrix} = F(z_k) \qquad (2)$$

where $F : \mathbb{R}^{n_z} \to \mathbb{R}^{n_z}$.

Remark: As is elaborated in Proctor et al. (2018), $g(x_k, u_k)$ is defined depending on the particular application, for example

(1) $g(x_k, u_k) = 0 \qquad \forall x_k, u_k$
(2) $g(x_k, u_k) = u_k \qquad \forall x_k, u_k$.

We wish to discover the dynamics in (1) which is characterized by the projected map $\hat{F} : \mathbb{R}^{n_z} \to \mathbb{R}^n$ defined by $\hat{F} = [I \ 0]F$. The choice of $g(x_k, u_k)$ is therefore irrelevant for our purpose.

Let $\mathcal{H}$ denote the infinite dimensional Hilbert space of observables chosen here as the space of locally integrable functions. So, for a compact $\mathcal{Z} \subset \mathbb{R}^{n_z}$, an element $\psi \in \mathcal{H}$ is a map $\psi : \mathcal{Z} \to \mathbb{R}$.

The Koopman operator $\mathcal{K} : \mathcal{H} \to \mathcal{H}$ is then defined for the dynamical system (1) by the relation

$$\mathcal{K}(\psi) = \psi \circ F \qquad \forall \psi \in \mathcal{H}. \qquad (3)$$

We will be interested in the finite dimensional approximation of the Koopman operator by choosing a finite set of observables $\{\psi_i\}_{i=1}^{n_\psi}$. With a view on approximating (3), we want to find a matrix $K \in \mathbb{R}^{n_\psi \times n_\psi}$ as an approximation of the the Koopman operator $\mathcal{K}$ such that $\forall z_k \in \mathbb{R}^{n_z}$:

$$\begin{bmatrix} \psi_1(z_{k+1}) \\ \psi_2(z_{k+1}) \\ \vdots \\ \psi_{n_\psi}(z_{k+1}) \end{bmatrix} \approx \begin{bmatrix} K_{11} & K_{12} & \dots & K_{1n_\psi} \\ K_{21} & K_{22} & \dots & K_{2n_\psi} \\ \vdots & \vdots & \ddots & \vdots \\ K_{n_\psi 1} & K_{n_\psi 2} & \dots & K_{n_\psi n_\psi} \end{bmatrix} \begin{bmatrix} \psi_1(z_k) \\ \psi_2(z_k) \\ \vdots \\ \psi_{n_\psi}(z_k) \end{bmatrix} \qquad (4)$$

Let us stack the observables to obtain $\Psi \in \mathcal{H}^{n_\psi}$ as a map $\Psi : \mathcal{Z} \to \mathbb{R}^{n_\psi}$ defined as $\Psi(z) = [\psi_1(z) \ \psi_2(z) \cdots \psi_{n_\psi}(z)]^T$. Note that the finite dimensional approximation K approaches the exact infinite dimensional operator $\mathcal{K}$ as the dimension n_ψ of the image space of Ψ is increased by stacking more linearly independent observables (in the sense of the inner product on $\mathcal{H}$) ψ_i to Ψ (Korda and Mezić, 2018b).

Consider equation (4) as a prediction equation for the observables. We can, by an appropriate choice of basis functions, obtain an expression for the prediction of the state itself. Recalling that $z = [x^T \ u^T]^T$, define the observable vector

$$\Psi(z) = [x^T \ u^T \ \psi_{n+m+1}(z) \ \dots \ \psi_{n_\psi}(z)]^T. \qquad (5)$$

Using equation (4), an approximation of the model (1) is given by (Abraham et al. (2017))

$$x_{k+1} \approx \hat{K}\Psi\left(\begin{bmatrix} x_k \\ u_k \end{bmatrix}\right), \qquad (6)$$

where $\hat{K} = [I_n \ 0]K$. Note that, as expected, the system dynamics are linear in a lifted space spanned by the basis functions. In this way, non-linearities can be captured by an appropriate choice of $\Psi(\cdot)$. This means that these functions play a crucial role in the accuracy of the approximated model, so their choice must be done with care. We select the basis functions by picking some or all of the nonlinear functions that appear in the underlying dynamical model, i.e. the R.H.S of (1), and show in simulation and in experiments that this leads to excellent results. Alternatively, polynomial, Fourier, radial or any other standard basis functions can be used, keeping in mind that the number of necessary basis functions using these bases might be considerably higher.

3. ONLINE COMPUTATION OF APPROXIMATE KOOPMAN OPERATORS

Assume that a data-set $\mathcal{Z} = \{z_i, z_i^+\}_{i=0,1,\cdots P}$ is collected, where z_i and z_i^+ are consecutive data points. Note that the data-set $\mathcal{Z}$ needs not correspond to a single trajectory, i.e. the data pairs need not be consecutive to one another. The collected data can be stacked column-wise in data matrices $\mathcal{D}, \mathcal{D}^+ \in \mathbb{R}^{n_\psi \times P}$ as

$$\mathcal{D} = [\Psi(z_0) \ \Psi(z_1) \cdots \Psi(z_P)], \ \mathcal{D}^+ = [\Psi(z_0^+) \ \Psi(z_1^+) \cdots \Psi(z_P^+)]$$

note that $\mathcal{D}^+$ is just the time-shifted version of $\mathcal{D}$. The optimization problem is thus

$$\min_K \quad \frac{1}{2}||\mathcal{D}^+ - K\mathcal{D}||_F^2 \qquad (7)$$

where $||X||_F = \text{Trace}(X^T X)$ is the Frobenius norm. The problem can be solved analytically by solving

$$K\mathcal{D}\mathcal{D}^T = \mathcal{D}^+\mathcal{D}^T$$

$$K = AG^\dagger$$

where we have defined (as is done in Abraham et al. (2017))

$$G = \frac{1}{P}\mathcal{D}\mathcal{D}^T = \frac{1}{P}\sum_{p=1}^{P}\Psi(z_p)\Psi(z_p)^T$$

$$A = \frac{1}{P}\mathcal{D}^+\mathcal{D}^T = \frac{1}{P}\sum_{p=1}^{P}\Psi(z_{p+1})\Psi(z_p)^T$$

3.1 Recursive computation

An online solution of (7) using the above definitions would require the storage and update of the data vectors $\mathcal{D}^+$, $\mathcal{D}$ which could result in memory management issues. Based on the well known recursive least squares algorithm (Hsia, 1977), we use an online update similar to the one proposed in (Peitz and Klus, 2018). Furthermore, the computation would become increasingly demanding with the number of stored data points P. Alternatively, only G, A and P can be stored and adding a data point at time step k can be done in the following way:

$$P^+ = P + 1$$
$$G^+ = \frac{1}{P^+}\left((P^+ - 1)G + \Psi(z_k)\Psi(z_k)^T\right)$$
$$A^+ = \frac{1}{P^+}\left((P^+ - 1)A + \Psi(z_k^+)\Psi(z_k)^T\right)$$

where $^+$ denotes the updated variables. Using these definitions, the Koopman operator can be updated online with consistent and relatively low computational complexity, regardless of the number of data points that have been added to the, now fictitious, data set $\mathcal{Z}$. In comparison to the online update proposed in (Peitz and Klus, 2018), we perform the update step only when the current model fails to match with observed data which is discussed next.

3.2 Koopman operator update

Although as seen in the previous section, adding new information to the model (in the form of data pairs $\{z_{k-1}, z_k\}$) involves only little computational burden, doing so at every time step is not advisable, as adding e.g. steady-state data-points would result in significant new information having little influence on updating the model (the weight $1/P$ of any new data point would make the effect negligible for large values of P). For this reason, incoming state and input data need to be analyzed in some way to determine if they represent *new* information not included in the model. There are several ways to do this, the one chosen here is motivated from the one presented in Slavakis and Theodoridis (2008) in the context of kernel based learning with projections. The principle at the core is to "learn" only if the current model disagrees (more than a certain threshold) with the observation.

Assume that state and input at the previous $h + 1$ time instants are stored in a vector

$$\phi = \begin{bmatrix} x_{k-1}^T & u_{k-1}^T & \dots & x_{k-h-2}^T & u_{k-h-2} \end{bmatrix}$$

a decision whether to add the data pair $\{z_{k-1}, z_k\}$ to the model is made by comparing the maximum prediction error of the approximated system (6) within this time window; if this value is greater than a threshold, the data point is added, i.e. add data point if

$$\left\|\begin{bmatrix} x_{k-h} \\ x_{k-h+1} \\ \vdots \\ x_k \end{bmatrix} - \begin{bmatrix} \hat{K}\Psi(z_{k-h-1}) \\ \hat{K}\Psi\left(\begin{bmatrix} \hat{K}\Psi(z_{k-h-1}) \\ u_{k-h} \end{bmatrix}\right) \\ \vdots \\ \hat{K}\Psi\left(\begin{bmatrix} \hat{K}\Psi\left(\begin{bmatrix} \hat{K}\Psi(\dots) \\ u_{k-2} \end{bmatrix}\right) \\ u_{k-1} \end{bmatrix}\right) \end{bmatrix}\right\|_\infty \geq \nu \quad (9)$$

where $\|\cdot\|_\infty$ is the vector infinity norm[1]. Note that any other norm may be used instead, however this choice offers the advantage that it is independent of h and it enables the use of physical insight to determine a suitable ν (i.e. an admissible error tolerance in physical units).

4. OBTAINING A QUASI-LPV MODEL FROM KOOPMAN OPERATOR

In the discussion that follows the following Lemma, as formulated in Zemouche et al. (2005), turns out to be useful, which can be regarded as a discrete-time version of the velocity-based linearization Leith and Leithead (1998)

Lemma 1. (Multivariable Mean Value Theorem). Let $g : \mathbb{R}^n \to \mathbb{R}^q$, assume that g is differentiable on $\mathrm{Co}(a, b)$, then there exist constant vectors $c_i \in \mathrm{Co}(a, b)$, $c_i \neq a$, $c_i \neq b$ $i = 1, \dots, q$ such that:

$$g(a) - g(b) = \left(\sum_{i,j=1}^{q,n} e_q(i)e_n(j)^T \frac{\partial g_i}{\partial x_j}(c_i)\right)(a - b)$$

where $e_s(i)$ is the ith column of I_s.

Given the approximate dynamic equations with truncated Koopman operator (6), using Lemma 1 (assuming the basis functions are differentiable) yields a velocity-based linearized dynamic equation which does not involve an additional approximation:

$$\Delta x_{k+1} \approx \hat{K}\left.\frac{\partial \Psi}{\partial x}\right|_{\tilde{x}, \tilde{u}} \Delta x_k + \hat{K}\left.\frac{\partial \Psi}{\partial u}\right|_{\tilde{x}, \tilde{u}} \Delta u_k$$

where $\tilde{x} \in \mathrm{Co}(x_k, x_{k+1})$, $\tilde{u} \in \mathrm{Co}(u_k, u_{k+1})$ which can in practice not be computed, therefore an approximation is made so that

$$\Delta x_{k+1} \approx \underbrace{\hat{K}\left.\frac{\partial \Psi}{\partial x}\right|_{x_k, u_k}}_{A(\rho_k)} \Delta x_k + \underbrace{\hat{K}\left.\frac{\partial \Psi}{\partial u}\right|_{x_k, u_k}}_{B(\rho_k)} \Delta u_k \quad (10)$$

where $\rho = H[x^T \ u^T]^T$ and H is a selection matrix. Finally (10) is augmented with the (non-incremental) state giving the dynamic equation

$$\begin{bmatrix} x_{k+1} \\ \Delta x_{k+1} \end{bmatrix} \approx \begin{bmatrix} I & A(\rho_k) \\ 0 & A(\rho_k) \end{bmatrix}\begin{bmatrix} x_k \\ \Delta x_k \end{bmatrix} + \begin{bmatrix} B(\rho_k) \\ B(\rho_k) \end{bmatrix}\Delta u_k \quad (11)$$

4.1 Input-Output Koopman-based model

Even though the goal of this framework is to obtain a data-driven model with as little a priori knowledge of the system as possible, in practice one strong assumption is made, namely that the state is known and measurable (given that the state is part of the observable functions, cf. (5)). This imposition might

---

[1] If the states have different physical units, they can be normalized; otherwise a different bound ν, one for each physical unit can be used

prove prohibitive for high order systems where it is impractical to assume even knowledge of the number of states, let alone being able to measure them. In this section we propose to tackle this issue using the input-output framework, and a new relaxed assumption is made that the order of the system is known. Assume the Input-Ouput behavior of the system is given by

$$y_{k+1} = \tilde{f}(y_k, u_k, q^{-1}) \tag{12}$$

note that (12) is a general discrete-time input-output model for a system governed by differential-algebraic equations. In this case, a fictitious state vector can be defined as

$$x_k = \begin{bmatrix} y_k^T & y_{k-1}^T & \cdots & y_{k-n_y} & u_{k-1} & \cdots & u_{k-n_u} \end{bmatrix} \tag{13}$$

where $y \in \mathbb{R}^l$ and $n = n_y + n_u$ is the assumed order of the system. This approach is commonly used in the Input-Output literature in order to use Lyapunov-like arguments to establish stability of Input-Output models (see e.g. Ali et al. (2010)). Using the state vector (13), the same definition of the basis functions (5) can be used and the Koopman operator can equally be computed recursively via (8). Analogous to Eq.(6), an approximation of system (12) is given by

$$y_{k+1} = \hat{K}\Psi(x_k, u_k) \tag{14}$$

where $\hat{K} = [I_l \quad 0]K$. A velocity-based model, following the discussion on the previous section is readily obtained by

$$\begin{bmatrix} y_{k+1} \\ \Delta x_{k+1} \end{bmatrix} \approx \begin{bmatrix} I & \hat{A}(\rho_k) \\ 0 & A(\rho_k) \end{bmatrix} \begin{bmatrix} y_k \\ \Delta x_k \end{bmatrix} + \begin{bmatrix} \hat{B}(\rho_k) \\ B(\rho_k) \end{bmatrix} \Delta u_k \tag{15}$$

where in contrast to the previously presented state-space formulation (11), the new definitions are used

$$\hat{A}(\rho_k) = \hat{K} \left. \frac{\partial \Psi}{\partial x} \right|_{x_k, u_k} \qquad \hat{B}(\rho_k) = \hat{K} \left. \frac{\partial \Psi}{\partial u} \right|_{x_k, u_k}.$$

Remark 1. In the Input-Output case, the same prediction equation (11) can be used in order to predict the full state (13). However, that would entail carrying out pointless computations to predicts future values of backward-shifted inputs and outputs.

5. PREDICTIVE CONTROLLER

We consider a finite horizon optimal controller which, at each sampling instant k minimizes the cost function

$$J_k = \sum_{i=0}^{N-1} \ell(e_{k+i}, \Delta u_{k+i}) + \zeta(x_{k+N}) \tag{16}$$

where $\zeta(x_{k+N}) = x_{k+N}^T T x_{x+N}$ is the terminal cost function, here chosen quadratic, and the usual stage cost in tracking MPC is used

$$\ell(e, \tilde{u}) = e^T Q e + \Delta u^T R \Delta u \tag{17}$$

with $e = x - r$ representing the deviation of the state from the reference. The matrices $T, Q \in \mathbb{R}^{l \times l}$ are positive semi-definite and $R \in \mathbb{R}^{m \times m}$ is positive-definite.

The optimization problem can thus be defined as

$$
\begin{aligned}
& \min_u J_k(e, \Delta u) \\
& \text{s.t.} \\
& \text{eq.(11)} \\
& u_{k+j} = u_{k-1} + \sum_{i=0}^{j-1} \Delta u_{k+i} \in \mathcal{U} \qquad j = \begin{bmatrix} 0 & N-1 \end{bmatrix} \\
& x_{k+j} \in \mathcal{X} \qquad j = \begin{bmatrix} 0 & N-1 \end{bmatrix}
\end{aligned} \tag{18}
$$

where $\mathcal{U}$, $\mathcal{X}$ is the set of admissible inputs and states, respectively.

Given a parameter (in this case state/input) dependent state space model in the form of (11) we can make use of the qLMPC algorithm [11], or MPC for quasi-LPV systems. This is an iterative algorithm which solves, at each sampling time, a series of quadratic programs by freezing the scheduling trajectory $\mathsf{P}_k = \begin{bmatrix} \rho_k^T & \rho_{k+1}^T & \cdots \rho_{k+N-1}^T \end{bmatrix}$ to the one obtained by shifting the previous solution, essentially turning the quasi-LPV model (11) into a Linear Time-Varying (LTV) model. The procedure is shown as Algorithm 1

Algorithm 1 qLMPC

Given: plant model, $\hat{Q}$, $\hat{R}$, N
1: $k \leftarrow 0$
2: Define $\mathsf{P}^0 = \mathbf{1}_N \otimes H[x_k^T \quad u_{k-1}^T]^T$
3: **repeat**
4:     $l \leftarrow 0$
5:     **repeat**
6:         Solve (18) using P_k^l to obtain U_k^l
7:         Predict state sequence given P_k^l and U_k^l
8:         Define $\mathsf{P}_k^{l+1} = \hat{H}(X^l, U^l)$
9:         $l \leftarrow l + 1$
10:     **until** stop criterion
11:     Apply u_k to the system
12:     Define $\mathsf{P}_{k+1}^0 = \hat{H}(X_k^l, U_k^l)$
13:     $k \leftarrow k + 1$
14: **until** end

Remark: This algorithm aims at finding the solution to a non-linear optimization problem by iterating (within one sampling interval) on the scheduling sequence. Even though this scheme typically converges in one or two steps even when no prior information on the scheduling sequence is available, see Cisneros et al. (2016), in practice (and in the applications reported in Section) no iteration is performed at all; instead a single QP is solved per sampling interval with the forward-shifted scheduling sequence of the previous sampling interval. Convergence is then achieved in real time over subsequent sampling intervals, similar to the *Real-Time Iteration Scheme* in Diehl et al. (2005).

6. EXPERIMENTAL RESULT: KOOPMAN-QLMPC FOR A CONTROL MOMENT GYROSCOPE

A Control Moment Gyroscope (CMG) is a mechanical device which consists of a flywheel mounted on a 3-degree-of-freedom gimbal. The flywheel and the first gimbal, bodies A and B in Figure 1, are actuated whereas the two outermost gimbals (C and D) are not, they can however be controlled by exploiting the gyroscopic effect. The goal is thus to track reference trajectories for the outer two gimbals, using the torque of the flywheel and the torque of the motor actuating the innermost gimbal as control variables. The state vector is

$$x = \begin{bmatrix} \theta_2 & \theta_3 & \theta_4 & \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 \end{bmatrix}.$$

This plant has strong nonlinear dynamics and a full model is considerably complex, see e.g. Hoffmann and Werner (2015), making it a good candidate for the Koopman framework. The CMG used for the experiments presented in this paper is the Model 750 from Educational Control Products (ECP).
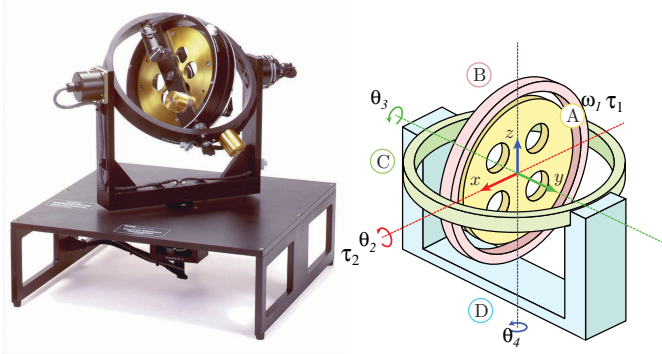

Fig. 1. 3-DOF Control Moment Gyroscope plant(left) and its CAD model(right))

6.1 Selection of basis functions

Given that a model for the CMG is available, a first meaningful approach is to select a few of the nonlinearities directly from the equations of motion to be used as basis functions. For this, all Coriolis terms (before inverting the inertia matrix) are considered i.e.

$$
\begin{aligned}
\Psi_1 = \big[x^T \ u^T \ & c_2 c_3 \dot\theta_2 \dot\theta_4 \ s_2 \dot\theta_2 \dot\theta_3 \ s_2 s_3 \dot\theta_3 \dot\theta_4 \ s_2 \dot\theta_1 \dot\theta_3 \\
& c_2 c_3 \dot\theta_1 \dot\theta_4 \ c_2 c_3^2 s_2 \dot\theta_4^2 \ c_2 s_2 \dot\theta_3^2 \ c_3 \dot\theta_3 \dot\theta_4 \ s_2^2 c_3 \dot\theta_3 \dot\theta_4 \ c_2^2 c_3 \dot\theta_3 \dot\theta_4 \\
& s_2 s_3 \dot\theta_1 \dot\theta_4 \ s_2 \dot\theta_1 \dot\theta_2 \ c_3 \dot\theta_2 \dot\theta_4 \ s_3 c_3 \dot\theta_4^2 \ c_2^2 c_3 \dot\theta_2 \dot\theta_4 \ s_2 c_2 \dot\theta_2 \dot\theta_3 \\
& s_3 c_2^2 c_3 \dot\theta_4^2 \ c_2 c_3 \dot\theta_1 \dot\theta_2 \ s_2 s_3 \dot\theta_1 \dot\theta_3 \ s_2 s_3 c_2 \dot\theta_3^2 \ s_3 c_3 \dot\theta_3 \dot\theta_4 \\
& c_2^2 c_3 \dot\theta_2 \dot\theta_3 \ s_2 c_2 c_3^2 \dot\theta_2 \dot\theta_4 \ s_3 c_2^2 c_3 \dot\theta_3 \dot\theta_4 \ c_3 \dot\theta_2 \dot\theta_3 \big]^T
\end{aligned} \tag{19}
$$

where the short-hand notations $c_i = \cos(\theta_i)$ and $s_i = \sin(\theta_i)$, $i = 2, 3$ are used. This yields a truncated Koopman operator of dimension $K_1 \in \mathbb{R}^{34 \times 34}$, where the subindex is used to denote its correspondence to Ψ_1. In order to explore the potential of the approach in cases when the nonlinear model is not available, a second set of basis functions is proposed, in which physical intuition is used: as it is expected that the contribution of the flywheel dominates the other Coriolis terms, only products of $\dot\theta_1$ with the rest of the velocities are considered; these products are then multiplied by trigonometric functions of the angles θ_2 and θ_3 (since, again from physical intuition it is clear that the rotations θ_1 and θ_4 do not have an effect on the dynamics) so that

$$
\begin{aligned}
\Psi_2 = \big[x^T \ u^T \ & c_2 \dot\theta_1 \dot\theta_2 \ c_2 \dot\theta_1 \dot\theta_3 \ c_2 \dot\theta_1 \dot\theta_4 \ c_3 \dot\theta_1 \dot\theta_2 \ c_3 \dot\theta_1 \dot\theta_3 \\
& c_3 \dot\theta_1 \dot\theta_4 \ s_2 \dot\theta_1 \dot\theta_2 \ s_2 \dot\theta_1 \dot\theta_3 \ s_2 \dot\theta_1 \dot\theta_4 \ s_3 \dot\theta_1 \dot\theta_2 \ s_3 \dot\theta_1 \dot\theta_3 \ s_3 \dot\theta_1 \dot\theta_4 \big]^T.
\end{aligned} \tag{20}
$$

yielding $K_2 \in \mathbb{R}^{21 \times 21}$.

6.2 Computation and update of the Koopman operator

The Koopman operator is computed recursively using the procedure described in Section 3. Before starting the predictive

controller, a short open-loop experiment [2] is performed using chirp test signals to start the Koopman algorithm and give a meaningful initial model to the MPC algorithm. For this training experiment, the parameter ν from (9) is set to $\nu = 0.001$ to encourage extracting information from it. When the controller is active, the factor is set to $\nu = 0.0025$ to avoid adding too many data points, which could render the model resistant to future updates.

6.3 Predictive Controller

A predictive controller according to Algorithm 1, using a model in the form of (11) determined as described in Section 5 is used to control the CMG. For this, a sampling time of $T_s = 0.01$s is used. This sampling time is also used for the sampling/update of the Koopman operator, so that both tasks (model update/control) are performed sequentially. For the predictive controller, a horizon of $N = 30$ is chosen and the tuning parameters are $Q = \text{diag}(1, 120, 120, 0.01, 5, 2, 2, 0, 0, 0, 0, 0, 0, 0)$, $T = 10Q$, $R = \text{diag}(3000, 750)$ and the constraints on the inputs are $|\tau_1| < 0.5$Nm, $|\tau_2| < 2$Nm.

Closed-loop results are shown in Figures 2 and 3, for the case where Ψ_1 and Ψ_2 are used, respectively. Each plot shows two experiments: one performed starting with the model after the training experiment, and a second one starting with the model after the first closed-loop experiment, in order to evaluate if performance improves after each iteration of a repetitive task. As expected, in both cases this is indeed the case, showing also that the second experiment needs to add fewer data points to the model, as there is less new information.
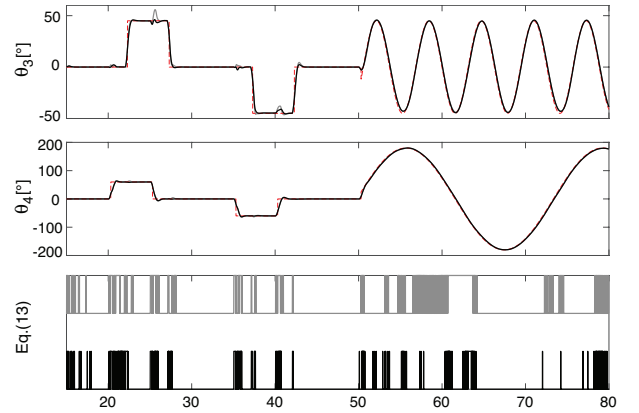

Fig. 2. Closed-loop experiment using K_1. Reference (-----), first experiment (——), second experiment (——).

Comparing both controllers, the one based on K_1 has better performance, particularly regarding cross-coupling at $t = 27$s and $t = 37$s, which is again expected given that the basis functions more closely resemble the equations of motion. It is however worth mentioning that both reference and tuning were selected to encourage an aggressive response as can be seen by the fact that inputs are driven into saturation (Figure 4); it is therefore remarkable that even with relatively simple basis functions, namely Ψ_2, the controller displays exceptional tracking performance.

---

[2] Before every experiment, both open- and closed-loop, a PI controller is used for 15s to bring the flywheel up to speed, with brakes applied on all DOF to give an initial condition for the closed-loop experiments of $\omega_1(15) = 40$rad/s, $\theta_i(15), \omega_i(15) = 0$, $i = 2, 3, 4$. At $t = 15$s control authority is switched to the proposed controller.
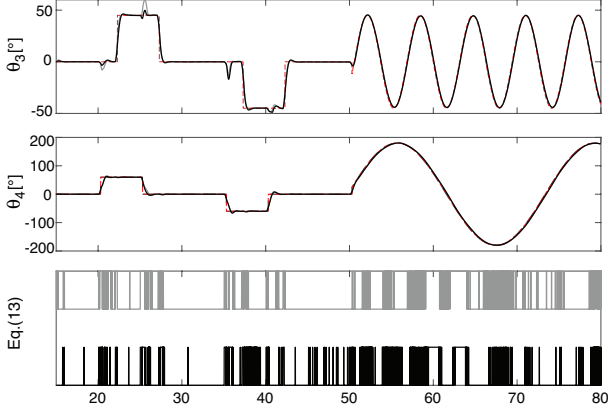
Fig. 3. Closed-loop experiment using K_2. Reference (- - - -), first experiment (——), second experiment (——).
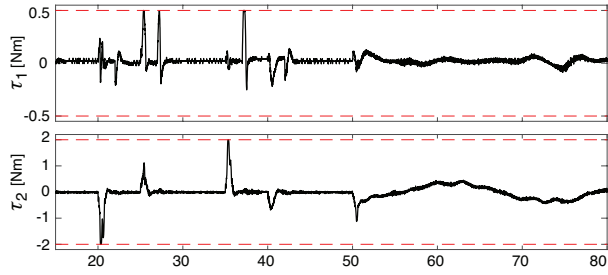

Fig. 4. Control inputs for one of the experiments

6.4 Input-Output Controller

Results of the Input-Output version of the presented approach (Section 4.1) are presented next. For this, the output is defined as $y(k) = [\theta_2(k) \quad \theta_3(k) \quad \theta_4(k) \quad \omega_1(k)]$ where $\omega_1 = \dot{\theta}_1$, and the state vector as $x_k = [y_k \ y_{k-1}]$. The basis function used are the discretized version of Ψ_2 using finite differences by replacing $\omega_1(k)\dot{\theta}_i(k)$ with $[\omega_1(k)\theta_i(k) \quad \omega_1\theta_i(k-1)]$, $i=2,3$ resulting in $K_{IO} \in \mathbb{R}^{33\times33}$, clearly a disadvantage of this approach is that it often leads to more basis functions. The tuning matrices corresponding to the outputs are discretized so that $Q = \text{diag}(1, 120, 120, 0.01, 50000, 20000, 20000, 0, 0, 0, 0, 0)$ (using the approximation $\dot{\theta}^2 = \Delta\theta^2/T_s^2$ with $T_s = 0.01$). The result of the experiment is shown in Figure 5, where compared to its state-space counterpart in Figure 3 it can be seen that the performance is comparable, with comparatively slower rise-time but noticeably lower cross-coupling.

Video footage of an experiment can be seen in `https://youtu.be/rDsuW6lncBY`, a comparison with a gain-scheduled LPV controller in `https://youtu.be/4a_dvWiBX0c`.

6.5 Comparison with (Korda and Mezić, 2018a)

The approach presented in this paper is based on the idea of identifying a linear Koopman model and implementing a linear MPC scheme based on this model. It differs from (Korda and Mezić, 2018a) in two respects:

- The method described in Sections 4 and 5 extends this idea by not using the identified linear Koopman model directly, but by converting it into a qLPV model of typically much lower order via velocity linearization.
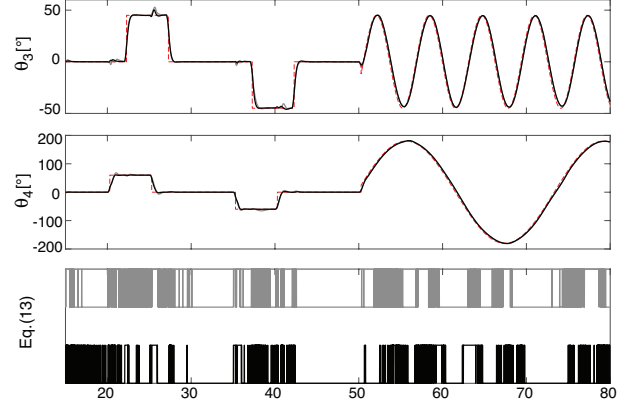

Fig. 5. Closed-loop experiment using K_{IO}. Reference (- - - -), first experiment (——), second experiment (——).

- In contrast to (Korda and Mezić, 2018a) we propose to not use RBF as observables when forming a basis for the nonlinear plant dynamics, but to exploit *a priori* knowledge about the plant when constructing the observables.

In order to investigate the benefits of the method proposed here with respect to both of the above aspects, it was compared in simulation with the method in (Korda and Mezić, 2018a) on the Control Moment Gyroscope. To have a meaningful comparison, the basis of observables Ψ_1 (the Coriolis terms) used here have also been applied when using the direct Koopman method of (Korda and Mezić, 2018a). While this is ongoing work, initial studies where both approaches (the one proposed here and the one in Korda and Mezić (2018a)) are using a basis formed by RBF suggest that even with a large number of basis functions (> 1000) no satisfactory performance can be achieved. (We expect to be able to provide evidence and details about this in the final version of the paper).

Fig. 6 shows a comparison of closed-loop performance on the same tracking scenario between the two methods. Whereas the tracking responses of θ_3 are reasonably close, tracking performance of θ_4 is much poorer with the direct Koopman approach. It should also be noted that tuning the controller via the cost function is much more challenging in the direct Koopman approach, as there are many more "lifted states" (25 artificial observables) to be penalized than in the qLPV approach, where only the seven physical states (and possibly their derivatives) are penalized. (Even though the authors of (Korda and Mezić, 2018a) propose not to penalize the additional states, this amounts to not using available degrees of freedom when tuning the controller.) The aim of tuning was in both cases to achieve the fastest possible rise time with no or little overshoot. It was also observed in the tuning process that in the direct Koopman approach of (Korda and Mezić, 2018a) the closed-loop performance is very sensitive to even small changes in the weights, i.e. that small perturbations in the weights can lead to serious performance degradation.

6.6 Computational complexity

For a fair comparison in simulation studies of computational efficiency on the CMG, the algorithm presented in (Korda and Mezić, 2018a) was adapted to the online update strategy as explained in section 3. Table 1 shows the computation times for both approaches. The "matrix creation" column shows the time needed to construct the matrices required for the "dense
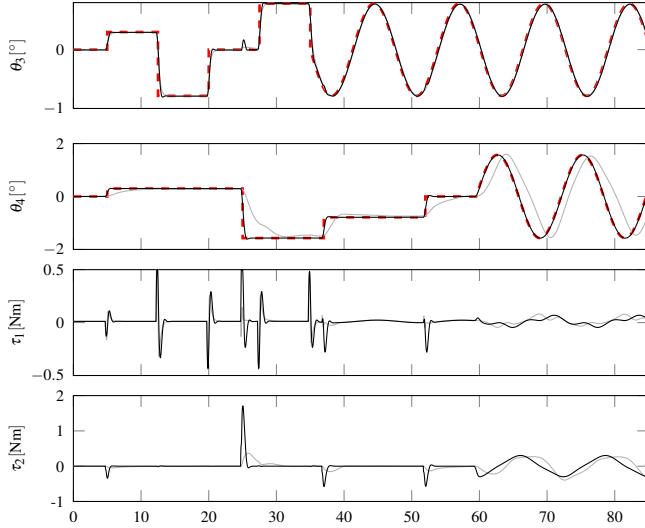
Fig. 6. Closed-loop perfomance comparison for Reference (- - - - -) with KoopmanMPC (——) and KqLMPC (——).

formulation" of the MPC problem, whereas the column QP shows the time consumed for solving the quadratic program. One can observe that forming the required block matrices takes more time than solving the QP. One can also see that both the time for matrix forming and the solver time in the qLMPC approach are significantly shorter than in the direct Koopman approach of (Korda and Mezić, 2018a). At least the former observation can be attributed to the fact that the lifted system used in the direct Koopman approach is much larger than the qLPV system obtained via velocity linearization.

Table 1. Mean Values of computational times for the two algorithms over 5,000 iterations

Algorithm	Forming matrices	Solving QP
direct Koopman	2.67 ms	1.99 ms
Koopman-based qLMPC	1.12 ms	0.29 ms

7. CONCLUSIONS & OUTLOOK

A data-driven nonlinear model predictive control approach is proposed based on the Koopman operator framework. The model used in the qLMPC algorithm can be updated online with low computational burden. The proposed scheme has been validated experimentally on a highly nonlinear MIMO system - a Control Moment Gyroscope with four degrees of freeedom. In comparison with a recently proposed approach that employs a linear Koopman model directly in LTI-based MPC, simulation studies suggest that converting the (typically very high dimensional) Koopman model via velocity linearization into a much lower dimensional qLPV system and then using the qLMPC approach, leads to superior control performance and to higher numerical efficiency. Another feature of the approach poposed here is to incorporate available physical insight into the plant to be controlled when choosing the observables; this has been demonstrated experimantally on the Control Moment Gyroscope.

REFERENCES

Abraham, I., De La Torre, G., and Murphey, T.D. (2017). Model-based control using koopman operators. *arXiv preprint arXiv:1709.01568*.

Ali, M., Abbas, H., and Werner, H. (2010). Controller synthesis for input-output LPV models. In *49th IEEE Conference on Decision and Control (CDC)*, 7694–7699. doi:10.1109/CDC.2010.5717576.

Arbabi, H., Korda, M., and Mezić, I. (2018). A data-driven koopman model predictive control framework for nonlinear partial differential equations. In *2018 IEEE Conference on Decision and Control (CDC)*, 6409–6414. IEEE.

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

Cisneros, P.S.G. and Werner, H. (2017). Parameter-dependent stability conditions for quasi-lpv model predictive control. In *2017 American Control Conference (ACC)*, 5032–5037. doi:10.23919/ACC.2017.7963735.

Cisneros, P.S.G., Voss, S., and Werner, H. (2016). Efficient nonlinear model predictive control via quasi-LPV representation. *55th IEEE Conference in Decision and Control*.

Diehl, M., Bock, H.G., and Schlöder, J.P. (2005). A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on control and optimization*, 43(5), 1714–1736.

Gonzalez Cisneros, P. and Werner, H. (2019a). Stabilizing Model Predictive Control for Nonlinear Systems in Input-Output quasi-LPV form. In *American Control Conference, 2019. Philadelphia, PA*. Available online: https://www.tuhh.de/t3resources/ics/user_upload/CiWe19b.pdf.

Gonzalez Cisneros, P.S. and Werner, H. (2019b). A velocity algorithm for nonlinear model predictive control". *submitted to IEEE Transactions on Control Systems Technology*. Available online: https://www.tuhh.de/t3resources/ics/user_upload/CiWe19a.pdf.

Hanke, S., Peitz, S., Wallscheid, O., Klus, S., Böcker, J., and Dellnitz, M. (2018). Koopman operator based finite-set model predictive control for electrical drives. *arXiv preprint arXiv:1804.00854*.

Hoffmann, C. and Werner, H. (2015). LFT-LPV modeling and control of a control moment gyroscope. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 5328–5333. doi:10.1109/CDC.2015.7403053.

Hsia, T. (1977). System identification: Least-squares methods. *Lexington, Mass.: Lexington Books*.

Kadali, R., Huang, B., and Rossiter, A. (2001). A data driven subspace approach to predictive controller design. *IFAC Proceedings Volumes*, 34(25), 365 – 370. 6th IFAC Symposium on Dynamics and Control of Process Systems 2001, Jejudo Island, Korea, 4-6 June 2001.

Kaiser, E., Kutz, J.N., and Brunton, S.L. (2018). Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A*, 474(2219), 20180335.

Kaiser, E., Kutz, J.N., and Brunton, S.L. (2019). Data-driven approximations of dynamical systems operators for control. *arXiv preprint arXiv:1902.10239*.

Koopman, B.O. (1931). Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5), 315–318.

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

Korda, M. and Mezić, I. (2018b). On convergence of extended dynamic mode decomposition to the koopman opera-

tor. *Journal of Nonlinear Science*, 28(2), 687–710.

Korda, M., Susuki, Y., and Mezić, I. (2018). Power grid transient stabilization using koopman model predictive control. *IFAC-PapersOnLine*, 51(28), 297–302.

Leith, D.J. and Leithead, W.E. (1998). Gain-scheduled and nonlinear systems: dynamic analysis by velocity-based linearization families. *International Journal of Control*, 70(2), 289–317.

Li, Q., Dietrich, F., Bollt, E.M., and Kevrekidis, I.G. (2017). Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10), 103111.

Peitz, S. and Klus, S. (2018). Feedback control of nonlinear pdes using data-efficient reduced order models based on the koopman operator. *arXiv preprint arXiv:1806.09898*.

Proctor, J.L., Brunton, S.L., and 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., and Kutz, J.N. (2018). Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1), 909–930.

Slavakis, K. and Theodoridis, S. (2008). Sliding window generalized kernel affine projection algorithm using projection mappings. *EURASIP Journal on Advances in Signal Processing*, 2008(1), 735351. doi:10.1155/2008/735351. URL `https://doi.org/10.1155/2008/735351`.

Williams, M.O., Rowley, C.W., and Kevrekidis, I.G. (2014). A kernel-based approach to data-driven koopman spectral analysis. *arXiv preprint arXiv:1411.2260*.

Zemouche, A., Boutayeb, M., and Bara, G.I. (2005). Observer design for nonlinear systems: An approach based on the differential mean value theorem. In *Proceedings of the 44th IEEE Conference on Decision and Control*, 6353–6358.

Zhang, H., Rowley, C.W., Deem, E.A., and Cattafesta, L.N. (2019). Online dynamic mode decomposition for time-varying systems. *SIAM Journal on Applied Dynamical Systems*, 18(3), 1586–1609.