Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-18 (2025) 133–138

Koopman-based improvement of the closed-loop response of controlled articulated soft robots

Matthias Seitz * **Annika Kirner** * **Christian Ott** *,**

* *Faculty of Electrical Engineering, TU Wien, 1040 Vienna, Austria.*
** *Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany (email: christian.ott@tuwien.ac.at).*

Abstract: To handle the complex nature of robots built from soft material, the data-based Koopman operator theory recently has been proposed as an alternative to model-based controller designs. In this paper we investigate the use of this theory for articulated soft robots, in which the elasticity is concentrated at the joints. In particular, we propose to apply the Koopman operator theory to the residual dynamics of an underlying model-based controller. For the inner loop control we utilize the Elastic Structure Preserving (ESP) control approach that has been successfully applied to a wide range of articulated soft robots. However, the control performance of the ESP control is clearly limited by the accuracy of the model, which motivates the combination with an outer data-based control approach. The concept is experimentally verified using antagonistically-driven elastic actuators with highly nonlinear stiffness characteristics.

Keywords: data-based control, flexible joint robots, robot control, Koopman operator theory

1. INTRODUCTION

The control of robotic systems with elasticity is a classical topic in robotics, which has drawn the attention of both robotics and control engineers. Originally, elastic robots have been broadly classified as flexible link robots and flexible joint robots depending on the main source of elasticity stemming from the robot links or from elastic components (e.g. gears) in the robot drive train (De Luca and Book (2016)). In both cases, traditional robot control approaches such as computed torque control are insufficient due to the underactuated nature of the dynamics, which motivated the development of a wide scope of customized model-based control approaches for different types of elastic robots (Spong (1987); De Luca (2000)).

Early works on elastic robot control typically considered the joint elasticity as a disturbance to the rigid-body dynamics. Instead, in articulated soft robots such as robots equipped with variable impedance actuators (Vanderborght et al. (2013)), the elasticity is introduced in the mechanic design deliberately and thus should not be compensated by the control (Albu-Schaeffer et al. (2008)). This type of robots motivated the development of the Elastic Structure Preserving (ESP) control framework (Keppler et al. (2018)), which aims to provide an additional virtual control input for the underactuated system coordinates in order to implement effective link side damping and other control actions. The ESP control approach has been successfully applied to various elastic joint technologies (Pollayil et al. (2022); Moyrón et al. (2025)).

Modeling robots with links made of soft material requires more complex modeling approaches based on continuum mechanics. For control purposes these models are often simplified by finite dimensional approximations (Della Santina et al. (2023)). Besides the increased computational complexity, challenges arise from the highly underactuated nature of these systems and the problem to obtain precise system parameters for the soft material components. In order to address these challenges, several works resort to data-based modeling and control approaches for robots based on soft material (Chen et al. (2024)) using concepts such as neural networks, reinforcement learning, or statistical approaches.

Recently, Koopman operator theory has been proposed as an effective tool to model and control robots based on soft material (Shi et al. (2023); Bruder et al. (2021a); Chen et al. (2022)). In this approach a finite dimensional approximation of the Koopman operator, describing the system dynamics, is learned from measured data. This finite dimensional model can be effectively used for a controller design based on linear control theory. In particular, optimization based approaches such as LQR or model predictive control (MPC) can be directly applied. This theory has been successfully applied for end-effector control of continuum robots under varying load conditions (Bruder et al. (2021b)) demonstrating that the requirement to collect measurement data in all relevant operating conditions in advance can be relaxed in certain cases.

While articulated soft robots with concentrated elasticity can be handled by model-based control approaches, model uncertainties and unmodelled dynamics clearly affect the achievable accuracy. One way to cope with this problem is to incorporate integral actions into the control (Keppler et al. (2022)). In this paper we propose a different strategy. Motivated by the successful application of the Koopman operator theory for handling the modeling challenges in

10.1016/j.ifacol.2025.10.209

soft robots (Bruder et al. (2021a)), we propose to use a data-based controller derived from Koopman operator theory for the residual dynamics of a model-based ESP controller (see Fig. 1).

The main contribution of this work, thus, is the formulation of a cascaded control approach for articulated soft robots combining the benefits of a robust model-based inner loop controller (ESP) and a data-based outer loop controller based on Koopman operator theory for improving the accuracy. We compare this solution with a straightforward application of a Koopman based controller for the open loop system dynamics and with a pure model-based ESP controller.

2. BACKGROUND

2.1 ESP Control of articulated soft robots

The Elastic Structure Preserving (ESP) control approach is a model-based control framework for articulated soft robots (Keppler et al. (2018)). Consider a robot with n joints modeled by the the system dynamics

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{K}(\boldsymbol{\theta} - \mathbf{q}) + \boldsymbol{\tau}_{ext} \tag{1a}$$

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \mathbf{K}(\boldsymbol{\theta} - \mathbf{q}) = \boldsymbol{\tau}_m + \boldsymbol{\tau}_f \ , \tag{1b}$$

with the joint $\mathbf{q} \in \mathbb{R}^n$ and motor angles $\boldsymbol{\theta} \in \mathbb{R}^n$, the rigid body inertia matrix $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n\times n}$, the nonlinear velocity dependent and gravity term summarized in $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$. Motor and link dynamics are coupled by the joint stiffness $\mathbf{K} \in \mathbb{R}^{n\times n}$. External forces act on the rigid-body part via $\boldsymbol{\tau}_{ext} \in \mathbb{R}^n$, while on the motor side we have the motor torques $\boldsymbol{\tau}_m \in \mathbb{R}^n$ as the control input as well as some friction terms $\boldsymbol{\tau}_f \in \mathbb{R}^n$ acting as a disturbance.

The dynamics (1) is under-actuated, since only the motor side has a direct control input via $\boldsymbol{\tau}_m$. The core idea of the ESP control framework is to utilize a state and input transformation, which transforms (1) in a *quasi-fully-actuated* [1] form, while preserving the system structure as an elastic robot with its intrinsic inertial properties and joint stiffness (Keppler et al. (2018)). As a desired closed loop dynamics we chose

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{K}(\boldsymbol{\eta} - \mathbf{q}) + \mathbf{u}_q + \boldsymbol{\tau}_{ext} \tag{2a}$$

$$\mathbf{B}\ddot{\boldsymbol{\eta}} + \mathbf{K}(\boldsymbol{\eta} - \mathbf{q}) = \mathbf{u}_m \ , \tag{2b}$$

with a virtual motor angle $\boldsymbol{\eta} \in \mathbb{R}^n$ and two virtual control inputs $\mathbf{u}_q \in \mathbb{R}^n$ and $\mathbf{u}_m \in \mathbb{R}^n$ acting on the link and motor side, respectively. Comparing (1a) and (2a) we obtain the state transformation

$$\boldsymbol{\eta} = \boldsymbol{\theta} + \mathbf{K}^{-1}\mathbf{u}_q \ , \tag{3}$$

while comparing (1b) and (2b) we can derive the input transformation

$$\boldsymbol{\tau}_m = \boldsymbol{\tau}_f + \mathbf{u}_m + \mathbf{u}_q + \mathbf{B}\mathbf{K}^{-1}\ddot{\mathbf{u}}_q, \tag{4}$$

which represents the final control law.

Based on this framework, the two virtual control inputs $\mathbf{u}_q$ and $\mathbf{u}_m$ can be utilized in a straight-forward way for implementing damping injection and other stabilizing control actions (Keppler et al. (2022)). Some care has to be made in the choice of $\mathbf{u}_q$ due to the necessity to compute its second time derivative in (4).

[1] We call this a *quasi-fully-actuated* dynamics, since the control input $\mathbf{u}_q$ in (2) can not be chosen arbitrarily, but must be continuously differentiable twice.

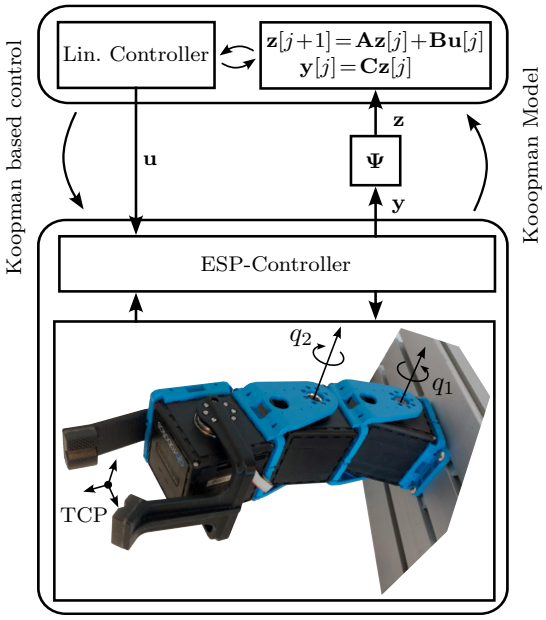

Fig. 1. Overview of the proposed control structure.

2.2 Koopman Operator Theory

Consider a discrete nonlinear dynamical system with input $\mathbf{u} \in \mathbb{R}^m$ and output $\mathbf{y} \in \mathbb{R}^p$ described by

$$\mathbf{y}[j+1] = \mathbf{F}\left(\mathbf{y}[j], \mathbf{u}[j]\right), \tag{5}$$

where $\mathbf{F}$ denotes the flow map of the system. The Koopman operator theory states that every system (5) can be exactly represented by an infinite dimensional linear dynamical system (Williams et al. (2015)). For that first the measured output is lifted to an infinite dimensional linear function space $\mathcal{F}$ with a lifting function called observable $\psi\left(\mathbf{y}[j], \mathbf{u}[j]\right) \in \mathcal{F}$. The Koopman operator $\mathcal{K}$ then is defined as an operator that advances the observable according to (5), i.e. $\mathcal{K}\psi = \psi \circ \mathbf{F}$, namely

$$\mathcal{K}\psi\left(\mathbf{y}[j], \mathbf{u}[j]\right) = \psi\left(\mathbf{y}[j+1], \mathbf{u}[j+1]\right). \tag{6}$$

For practical use, a finite dimensional approximation of the Koopman operator is necessary, which can be obtained, e.g., by the Extended Dynamic Mode Decomposition (EDMD) (Williams et al. (2015)). For that a finite dimensional subspace $\mathcal{F}_\mathcal{D} \in \mathcal{F}$ spanned by the linearly independent observables $\psi_i \in \mathcal{F}$ is considered. By definition every function $\psi \in \mathcal{F}_\mathcal{D}$ can then be written as

$$\psi\left(\mathbf{y}[j], \mathbf{u}[j]\right) = \sum_{i=1}^{N} c_i \psi_i\left(\mathbf{y}[j], \mathbf{u}[j]\right) = \mathbf{c}^{\mathrm{T}} \boldsymbol{\Psi}\left(\mathbf{y}[j], \mathbf{u}[j]\right) \tag{7}$$

with the weights $\mathbf{c} \in \mathbb{R}^N$ and the dictionary of observables $\boldsymbol{\Psi}\left(\mathbf{y}[j], \mathbf{u}[j]\right) \in \mathbb{R}^N$.

For control design a model of the form

$$\mathbf{z}[j+1] = \mathbf{A}\mathbf{z}[j] + \mathbf{B}\mathbf{u}[j] \tag{8a}$$

$$\mathbf{y}[j] = \mathbf{C}\mathbf{z}[j] \tag{8b}$$

is desired. Herein, $\mathbf{z}[j] = \boldsymbol{\Psi}\left(\mathbf{y}[j]\right)$ represents the dictionary. The choice of the dictionary is important to obtain a good approximation of the original system. To make the reconstruction of the measured output $\mathbf{y}$ easier later on, the first n entries to the dictionary can be chosen as the components of the measured output $\mathbf{y}$, i.e.

$$\boldsymbol{\Psi}\left(\mathbf{y}\right) = \left[y_1 \ \cdots \ y_p \ \psi_{p+1}\left(\mathbf{y}\right) \ \cdots \ \psi_N\left(\mathbf{y}\right)\right]^{\mathrm{T}} \in \mathbb{R}^N \ . \tag{9}$$

Note that, to learn a model where the input appears linearly, the dictionary does not contain the input at the jth time step, but, in general, it can contain inputs of previous time steps.

Next, a set of K measurements is needed to find an approximation of the Koopman operator. The EDMD method uses a data set that is ordered into snapshot pairs

$$\mathbf{a}_k = \mathbf{y}[j_k], \quad \mathbf{b}_k = \mathbf{F}\left(\mathbf{y}[j_k], \mathbf{u}[j_k]\right) = \mathbf{y}[j_k + 1], \tag{10}$$

where $\mathbf{a}_k \in \mathbb{R}^p$ is the measurement at time j_k and $\mathbf{b}_k \in \mathbb{R}^p$ is the measurement at the next time step $j_k + 1$. Unlike $\mathbf{a}_k$ and $\mathbf{b}_k$, the snapshot pairs themselves do not have to be from the same time series or be of any particular order. The snapshot pairs are then lifted into a higher dimensional space using the dictionary (9). With regards to (8) we aim to obtain a Koopman operator in the form

$$\mathbf{K} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{11}$$

where $\mathbf{I} \in \mathbb{R}^{m \times m}$ denotes the identity matrix and $\mathbf{0} \in \mathbb{R}^{m \times N}$ denotes a zero matrix. This is achieved by the minimization

$$\min_{\mathbf{A}'\,\mathbf{B}'} \sum_{k=1}^{K} \left\| \mathbf{A}'\mathbf{\Psi}\left(\mathbf{a}_k\right) + \mathbf{B}'\mathbf{u}_k - \mathbf{\Psi}\left(\mathbf{b}_k\right) \right\|_2^2. \tag{12}$$

which gives best approximation in the L^2-norm sense (Korda and Mezić (2018)).

Since in (9) the components of the measured output are the first p elements of the dictionary, the reconstruction of the states becomes trivial and $\mathbf{C} \in \mathbb{R}^{p \times N}$ is given by

$$\mathbf{C} = \left[\mathbf{I}\ \mathbf{0}\right]. \tag{13}$$

Projection Evolving the lifted output $\mathbf{\Psi}\left(\mathbf{y}[j]\right)$ with the approximated model (8a) does not give exact results. Given an output space, the dictionary defines a manifold in the lifted space. However, the evolved state $\hat{\mathbf{z}}[j+1]$ obtained from $\mathbf{\Psi}\left(\mathbf{y}[j]\right)$ with the learned linear model (8a) does not necessarily belong to the manifold. To increase the accuracy, a projection operator can be used (Bruder et al. (2021a)). Ideally, it projects $\hat{\mathbf{z}}[j+1]$ to the lifted measured output at time step $j+1$, and thus on the manifold. Writing this in terms of snapshot pairs, yields

$$\mathbf{P}\left(\mathbf{A}\mathbf{\Psi}\left(\mathbf{a}_k\right) + \mathbf{B}\mathbf{u}_k\right) = \mathbf{\Psi}\left(\mathbf{b}_k\right) \tag{14}$$

with projection operator $\mathbf{P}$. In general, this operator must be approximated. For that, at first, the matrix

$$\mathbf{\Omega}_a = \begin{bmatrix} \left(\mathbf{A}\mathbf{\Psi}\left(\mathbf{a}_1\right) + \mathbf{B}\mathbf{u}_1\right)^{\mathrm{T}} \\ \vdots \\ \left(\mathbf{A}\mathbf{\Psi}\left(\mathbf{a}_K\right) + \mathbf{B}\mathbf{u}_K\right)^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{K \times N} \tag{15}$$

is constructed and the lifted snapshots $\mathbf{\Psi}\left(\mathbf{b}_k\right)$ are compiled into the matrix

$$\mathbf{\Psi}_b = \left[\mathbf{\Psi}\left(\mathbf{b}_1\right)\ \ldots\ \mathbf{\Psi}\left(\mathbf{b}_K\right)\right]^{\mathrm{T}} \in \mathbb{R}^{K \times N}. \tag{16}$$

With these matrices, the best approximation in the L^2-norm sense of the projection operator is given by

$$\mathbf{P}^* = \left(\mathbf{\Omega}_a^{\dagger}\mathbf{\Psi}_b\right)^{\mathrm{T}}, \tag{17}$$

with $\mathbf{P}^* \in \mathbb{R}^{N \times N}$ where $\dagger$ denotes the Moore-Penrose pseudoinverse. The model used for the learning process can then be modified with the projection operator to

$$\mathbf{z}[j+1] = \hat{\mathbf{A}}\mathbf{z}[j] + \hat{\mathbf{B}}\mathbf{u}[j], \tag{18}$$

where $\hat{\mathbf{A}} = \mathbf{P}^*\mathbf{A}$ and $\hat{\mathbf{B}} = \mathbf{P}^*\mathbf{B}$.

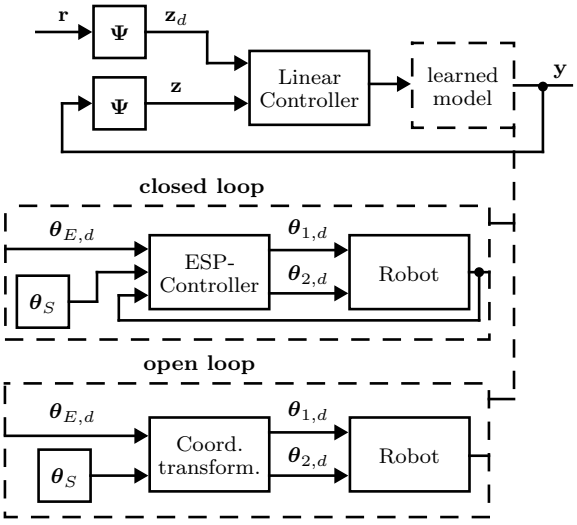

Fig. 2. Control structure of the two methods that are compared in this work. For the *open loop case* a model of the uncontrolled robot is learned, whereas for the *closed loop case* a model is learned for the robot that is controlled with an ESP controller.

3. PROPOSED CONTROL APPROACH

3.1 Considered experimental target system

In this study, we utilize a soft robot consisting of two motor position-controlled antagonistic variable stiffness actuators *qbmove advanced* (see Fig. 1). Each actuator is equipped with two servo motors, which are connected to the shaft via highly nonlinear springs. The equilibrium position $\theta_{E,d}$ of the shaft in each actuator is determined by the average of the motor positions. The overall stiffness can be adjusted via the difference θ_S between the motor positions. In the following, we set a constant relative motor distance, thus operating the actuators as nonlinear series elastic actuators with the control input $\theta_{E,d}$.

Notice that the original ESP control theory as summarized in Sec. 2.1 was developed for articulated soft robots with torque-controlled actuators. However, in (Moyrón et al. (2025)) it was shown that an application to position controlled actuators is possible. This is achieved by replacing the motor control action (4) by a motor side admittance controller. The virtual motor dynamics (2b) thus is utilized as the desired admittance. In this case, the motor angle is treated as a control input and it is computed from $\boldsymbol{\theta}_d = \boldsymbol{\eta} - \mathbf{K}^{-1}\mathbf{u}_q$.

3.2 Control structure

In this work we compare the performance of two approaches to design a controller using the Koopman operator theory. In the first approach (which we denote by *open loop case*), we learn a linear Koopman model of the robot dynamics and then design a linear Koopman controller. For the second approach (which we call *closed loop case*) we combine the Koopman-based control with an underlying model-based (ESP) controller. A linear model thus is learned for the residual dynamics of the ESP-controlled robot and a linear Koopman controller is designed for that

learned model. Figure 2 shows the control structure for these two approaches.

3.3 Selection of observables

The accuracy of the learned model highly depends on the choice of the dictionary. Table 1 gives an overview of the different dictionaries which we compare. The first four variations of the dictionary consist of only the output and time delays of the output and input. These signals represent terms of a Taylor series of the unknown dynamics function. Some variations include entries that are inspired by the physics of the system. For those, nonlinear functions that are known to be part of the system dynamics can be included as entries to the dictionary, e.g. the dynamics components of a rigid 2-DOF robot arm and the nominal elastic torque of the actuators. To make the modeling independent of constant values that are not exactly known, the first three nonzero terms of a Taylor series of those functions are chosen as entries to the dictionary.

Table 1. Overview of the used dictionaries.

Dictionary	Step size of delays	Number of delays	Physics inspired	ReLU	Size
$\boldsymbol{\Psi}_{s1d10}$	1	10	-	-	86
$\boldsymbol{\Psi}_{s1d30}$	1	30	-	-	246
$\boldsymbol{\Psi}_{s3d10}$	3	10	-	-	86
$\boldsymbol{\Psi}_{s3d30}$	3	30	-	-	246
$\boldsymbol{\Psi}_{p}$	3	10	✓	-	171
$\boldsymbol{\Psi}_{R}$	3	10	-	✓	206
$\boldsymbol{\Psi}_{pR}$	3	10	✓	✓	231

Initial measurements of step responses of the robot suggest that the system behavior varies depending on its direction of movement. To incorporate this observed behavior in the model, the angular velocities of the output are split into two entries to the dictionary in some variations. Each of them should only represent the movement in one direction. To guarantee that, the rectified linear unit (ReLU) function, defined as $\mathrm{R}(x) = \max(0, x)$, is used. Equation (19) shows as an example all the entries to the dictionary $\boldsymbol{\Psi}_R$. Note that this dictionary will be the one used for the final control design. Therein, the vector $\boldsymbol{\psi}_R$ contains the angular velocities Δq_i and $\Delta\theta_i$ of the output that are each split into a positive and a negative part using the ReLU function. The full dictionary then consists of the output, time delays of the output, the vector $\boldsymbol{\psi}_R$, time delays of $\boldsymbol{\psi}_R$, and time delays of the input:

$$\boldsymbol{\Psi}_R = \begin{bmatrix} \mathbf{y}[j] \\ \mathbf{y}[j-3] \\ \vdots \\ \mathbf{y}[j-30] \\ \boldsymbol{\psi}_R(\mathbf{y}[j]) \\ \boldsymbol{\psi}_R(\mathbf{y}[j-3]) \\ \vdots \\ \boldsymbol{\psi}_R(\mathbf{y}[j-27]) \\ \mathbf{u}[j-3] \\ \vdots \\ \mathbf{u}[j-30] \end{bmatrix}, \boldsymbol{\psi}_R = \begin{bmatrix} \mathrm{R}(\Delta q_1) \\ -\mathrm{R}(-\Delta q_1) \\ \mathrm{R}(\Delta\theta_{1,1}) \\ -\mathrm{R}(-\Delta\theta_{1,1}) \\ \mathrm{R}(\Delta\theta_{1,2}) \\ -\mathrm{R}(-\Delta\theta_{1,2}) \\ \mathrm{R}(\Delta q_2) \\ -\mathrm{R}(-\Delta q_2) \\ \mathrm{R}(\Delta\theta_{2,1}) \\ -\mathrm{R}(-\Delta\theta_{2,1}) \\ \mathrm{R}(\Delta\theta_{2,2}) \\ -\mathrm{R}(-\Delta\theta_{2,2}) \end{bmatrix} \tag{19}$$

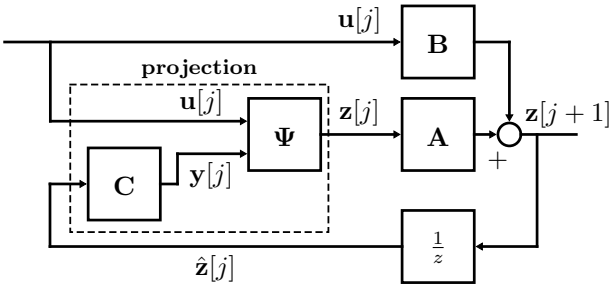

Fig. 3. Block diagram of the simulation that is used to verify the learned models.

Table 2. Average RMSE for the models with different dictionaries. All values are in radian.

Dictionary	open loop	closed loop
$\boldsymbol{\Psi}_{s1d10}$	0.0269	0.0193
$\boldsymbol{\Psi}_{s1d30}$	0.0251	0.0161
$\boldsymbol{\Psi}_{s3d10}$	0.0215	0.0148
$\boldsymbol{\Psi}_{s3d30}$	0.0211	0.0154
$\boldsymbol{\Psi}_{p}$	0.0302	0.0141
$\boldsymbol{\Psi}_{R}$	0.0219	0.0150
$\boldsymbol{\Psi}_{pR}$	0.0273	0.0137

4. KOOPMAN-BASED SYSTEM IDENTIFICATION

Learning To learn a model, a data set of multiple measured step responses is utilized. To train the model, the *ksysid.m* class provided by Bruder et al. (2021a) is used. This class is further modified to support the use of custom dictionaries.

Simulation One way to simulate the learned models is to initialize their states by lifting the initial output using the dictionary and simulating them as LTI models. However, in this way an asymmetrical behavior, like the one expected from the models with ReLU functions in the dictionary, can not be obtained. To address this, we utilize a projection as shown in Fig. 3. At every time step the state $\hat{\mathbf{z}}[j]$ is projected back on a manifold that is defined by the dictionary. This is done by first reconstructing the output $\mathbf{y}[j]$ from the state $\hat{\mathbf{z}}[j]$ and then lifting the output to the new state $\mathbf{z}[j]$ with the dictionary (Bruder et al. (2021a)).

Model Validation To validate the learned model corresponding to the different dictionaries a new measurement of the system response to sequential steps, including the step back to the original position, is used. To compare the performance of the different models, the root mean squares error (RMSE) between the simulation and the measurements is calculated for each component of the output and averaged over all outputs. Table 2 shows the resulting RMSEs for the different models and the different cases. For control design the model with the dictionary $\boldsymbol{\Psi}_R$ is chosen. This shows good results in the validation and allows to address the directional dependent behavior via the ReLU functions.

5. KOOPMAN-BASED CONTROL

For the linear controller (see Fig. 2), three different controllers are compared, namely an LQR, an LQR extended by an integrator (denoted LQR-I), and an MPC controller.

LQR The cost function for the LQR design is given as

$$J = \sum_{j=0}^{\infty} \mathbf{z}[j]^{\mathrm{T}} \mathbf{W}_z \mathbf{z}[j] + \mathbf{u}[j]^{\mathrm{T}} \mathbf{W}_u \mathbf{u}[j]. \tag{20}$$

The weights $\mathbf{W}_u$ and $\mathbf{W}_z$ are chosen as diagonal matrices. In the *open loop case* the input weights are chosen as $\mathbf{W}_u = \mathrm{diag}\,(10, 50)$ and the state weights are chosen to be $w_{z,i} = 1$ for the joint angles and $w_{z,i} = 10^3$ for the velocities. In the *closed loop case* the input weights are chosen as $\mathbf{W}_u = \mathrm{diag}\,(1, 10)$ and the state weights are $w_{z,i} = 1$ for the joint angles, $w_{z,i} = 5 \cdot 10^2$ for the velocity of the first joint angle, and $w_{z,i} = 10^2$ for the second joint angle.

LQR-I For the LQR-I controller, the model is extended by an integrator of the joint angles. Again, the cost (20) is used. In the *open loop case* the input weights are $\mathbf{W}_u = \mathrm{diag}\,(100, 100)$, the weights corresponding to the joint angles are $w_{z,i} = 1$, the ones for the velocity of the first joint angle are $w_{z,i} = 10^4$, for the second joint angle they are $w_{z,i} = 10^3$, and for the integrated joint angles they are $w_{z,i} = 10^{-2}$. For the *closed loop case* they are chosen as $\mathbf{W}_u = \mathrm{diag}\,(10, 10)$ for the input weights, $w_{z,i} = 1$ for the joint angles, $w_{z,i} = 5 \cdot 10^3$ for the velocity of the first joint, $w_{z,i} = 5 \cdot 10^2$ for the velocity of the second joint, and $w_{z,i} = 10^{-2}$ for the integrated joint angles.

MPC To guarantee an integral action for the MPC, the state space model is formulated in the differential state and augmented by the output, i.e.

$$\begin{bmatrix} \Delta\mathbf{z}[j+1] \\ \mathbf{y}[j+1] \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{CA} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{z}[j] \\ \mathbf{y}[j] \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{CB} \end{bmatrix} \Delta\mathbf{u}[j].$$

The cost function is given by

$$\begin{aligned} J\left(\{\mathbf{u}[i]\}_{i=0}^{N_p-1}\right) =& \mathbf{U}^{\mathrm{T}} \mathbf{W}_u^2 \mathbf{U} + \Delta\mathbf{U}^{\mathrm{T}} \mathbf{W}_{\Delta u}^2 \Delta\mathbf{U} \\ &+ (\mathbf{Y} - \mathbf{Y}_d)^{\mathrm{T}} \mathbf{W}_y^2 (\mathbf{Y} - \mathbf{Y}_d), \end{aligned} \tag{21}$$

where $\mathbf{W}_u, \mathbf{W}_{\Delta u} \in \mathbb{R}^{mN_p \times mN_p}$ and $\mathbf{W}_y \in \mathbb{R}^{pN_p \times pN_p}$ denote the diagonal weighting matrices for the input, the change of the input, and the error between the output and the reference, respectively. Further, $\mathbf{U} \in \mathbb{R}^{mN_p}$, $\mathbf{Y} \in \mathbb{R}^{pN_p}$, and $\mathbf{Y}_d \in \mathbb{R}^{pN_p}$ denote the vectors containing all predicted inputs, predicted outputs, and desired outputs. The sampling time of the prediction horizon is chosen as $N_p = 30$. In the *open loop case* the input weights are chosen as $\mathbf{W}_{u,i} = \mathrm{diag}\,(10, 20)$ and $\mathbf{W}_{\Delta u,i} = \mathrm{diag}\,(300, 300)$. In the *closed loop case* the input weights are chosen as $\mathbf{W}_{u,i} = \mathrm{diag}\,(10, 10)$ and $\mathbf{W}_{\Delta u,i} = \mathrm{diag}\,(1, 1)$. These weights are held constant over the whole prediction horizon. Therefore, the weighting matrices in (21) are composed of a repetition of those weights. In both cases, the weights for the output are chosen as $w_y = 1$ for each entry corresponding to a joint angle.

Step response The first experiment evaluates the step response to a step from an initial position $\mathbf{q} = \mathbf{0}$ to a desired joint position $\mathbf{q}_d = [0.3rad, -0.3rad]^T$. To show a dependency of the direction of movement, a step in the reverse direction back to $\mathbf{q} = \mathbf{0}$ is also included. The resulting behavior of the first joint angle can be seen in Fig. 4 for all controllers in the *open loop case* and in Fig. 5 for the controllers in the *closed loop case*.

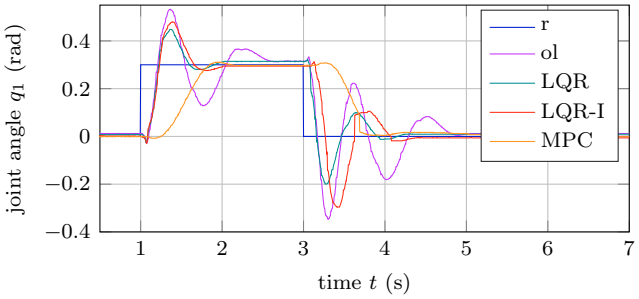

Fig. 4. Comparison of step responses of the different controllers in the *open loop case*. The blue line represents the reference (r) and the magenta line shows the pure open loop (ol) behavior.

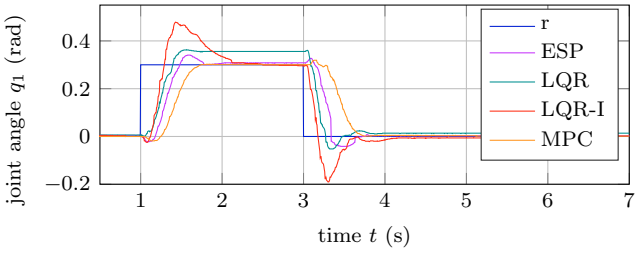

Fig. 5. Comparison of step responses of the different controllers in the *closed loop case*.

Trajectory Tracking In the second experiment, trajectory tracking is addressed. Two trajectories for the TCP of the robot are defined describing an "8" like and a rectangular shape. These should be tracked over a total time of $T = 40$s for one cycle and with a constant velocity. Since the LQR shows a considerable steady state error in the step response it is excluded for this experiment. The resulting trajectories for all remaining controllers can be seen in Fig. 6.

Discussion The results of the step response suggest that most controllers designed for the *closed loop case* improve the performance compared to the ESP controller itself. Further, the performance of all controllers designed for the *open loop case* can be improved by combining them with the ESP controller to the *closed loop case*. Also the damping of the robot that is imposed by the ESP controller is upheld for all linear controllers designed for the *closed loop case*. For trajectory tracking it can be seen that mainly the inclusion of an integrating behavior in the control design improves the performance. The outer loop Koopman controller thus can augment the missing integral action to the ESP control.

6. CONCLUSIONS

In this paper we propose a combination of data-based Koopman operator theory with an underlying model-based controller for articulated soft robots, whose accuracy clearly is affected by the quality of the used system model. This combination allows to incorporate additional control actions like integral actions to the residual dynamics. We compare this approach experimentally with a direct application of the Koopman operator theory to the open loop dynamics of the same system. The experimental

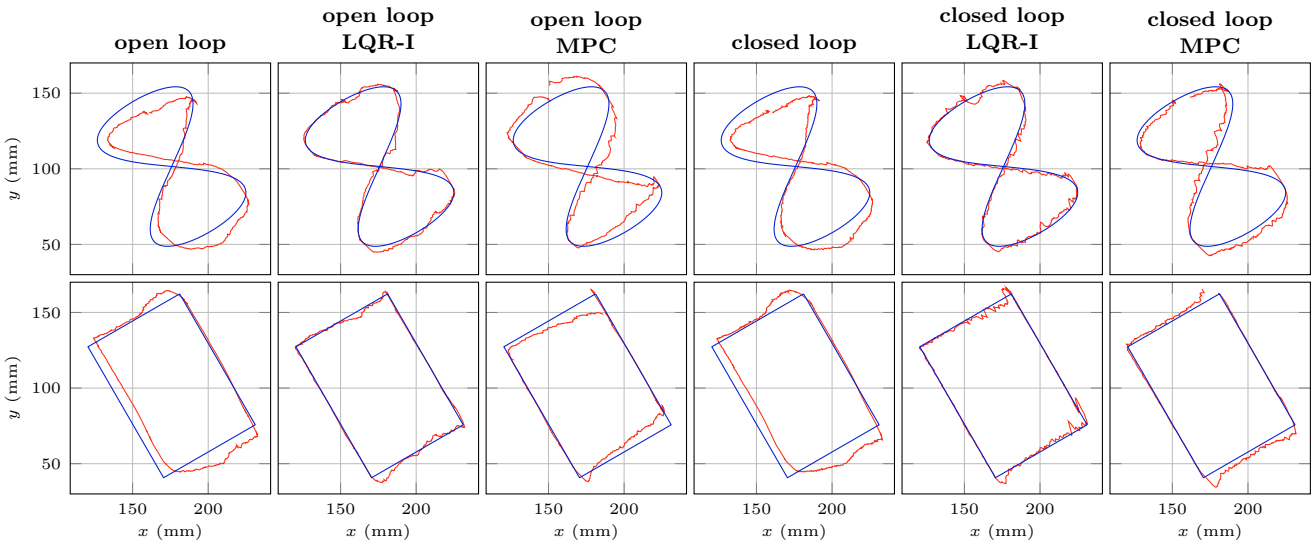

Fig. 6. Trajectory following performance of the different controllers: TCP position in red and reference in blue.

results confirm that this approach allows to improve the control performance. However, the resulting performance is limited by the accuracy of the Koopman model for the residual dynamics. In this work the observables were chosen based on physical reasoning and the observed system behavior. Further improvements may be obtained by the use of machine learning techniques for finding a different set of observables, which gives a better representation of the residual dynamics.

REFERENCES

Albu-Schaeffer, A., Eiberger, O., Grebenstein, M., Haddadin, S., Ott, C., Wimboeck, T., Wolf, S., and Hirzinger, G. (2008). Soft robotics: From torque feedback controlled lightweight robots to intrinsically compliant systems. *Robotics and Automation Magazine*.

Bruder, D., Fu, X., Gillespie, R.B., Remy, C.D., and Vasudevan, R. (2021a). Data-driven control of soft robots using koopman operator theory. *IEEE Transactions on Robotics*, 37(3), 948–961. URL `https://github.com/ramvasudevan/soft-robot-koopman`.

Bruder, D., Fu, X., Gillespie, R.B., Remy, C.D., and Vasudevan, R. (2021b). Koopman-based control of a soft continuum manipulator under variable loading conditions. *IEEE Robotics and Automation Letters*, 6(4), 6852–6859.

Chen, J., Dang, Y., and Han, J. (2022). Offset-free model predictive control of a soft manipulator using the koopman operator. *Mechatronics*, 86, 102871.

Chen, Z., Renda, F., Gall, A., Mocellin, L., Bernabei, M., Dangel, T., Ciuti, G., Cianchetti, M., and Stefanini, C. (2024). Data-driven methods applied to soft robot modeling and control: A review. *IEEE Transactions on Automation Science and Engineering*, 1–16.

De Luca, A. (2000). Feedforward/feedback laws for the control of flexible robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, 233–240.

De Luca, A. and Book, W.J. (2016). *Robots with Flexible Elements*, 243–282. Springer International Publishing.

Della Santina, C., Duriez, C., and Rus, D. (2023). Model-based control of soft robots: A survey of the state of the art and open challenges. *IEEE Control Systems Magazine*, 43(3), 30–65.

Keppler, M., Lakatos, D., Ott, C., and Albu-Schäffer, A. (2018). Elastic structure preserving (esp) control for compliantly actuated robots. *IEEE Transactions on Robotics*, 34(2), 317–335.

Keppler, M., Raschel, C., Wandinger, D., Stemmer, A., and Ott, C. (2022). Robust stabilization of elastic joint robots by esp and pid control: Theory and experiments. *IEEE Robotics and Automation Letters*, 7(3), 8283–8290.

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

Moyrón, J., Ott, C., Kirner, A., and Moreno-Valenzuela, J. (2025). Elastic structure preserving control for flexible joint robots with position-controlled actuators. *IEEE Transactions on Control Systems Technology*. doi: 10.1109/TCST.2025.3562024.

Pollayil, G.J., Meng, X., Keppler, M., Pfanne, M., Bicchi, A., and Ott, C. (2022). Elastic structure preserving impedance control for nonlinearly coupled tendon-driven systems. *IEEE Control Systems Letters*, 6, 1982–1987.

Shi, L., Liu, Z., and Karydis, K. (2023). Koopman operators for modeling and control of soft robotics. *Current Robotics Reports*, 4(2), 23–31.

Spong, M. (1987). Modeling and control of elastic joint robots. *Transactions of the ASME: Journal of Dynamic Systems, Measurement, and Control*, 109, 310–319.

Vanderborght et al. (2013). Variable impedance actuators: A review. *Robotics and Autonomous Systems*, 61(12), 1601–1614.

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, 1307–1346.