Mechanical Design and Data-Enabled Predictive Control of a Planar Soft Robot

Huanqing Wang, Kaixiang Zhang, Kyungjoon Lee, Yu Mei, *Graduate Student Member, IEEE*, Keyi Zhu, Vaibhav Srivastava, *Senior Member, IEEE*, Jun Sheng, *Member, IEEE*, and Zhaojian Li, *Senior Member, IEEE*

***Abstract*—Soft robots offer a unique combination of flexibility, adaptability, and safety, making them well-suited for a diverse range of applications. However, the inherent complexity of soft robots poses great challenges in their modeling and control. In this letter, we present the mechanical design and data-driven control of a pneumatic-driven soft planar robot. Specifically, we employ a data-enabled predictive control (DeePC) strategy that directly utilizes system input/output data to achieve safe and optimal control, eliminating the need for tedious system identification or modeling. In addition, a dimension reduction technique is introduced into the DeePC framework, resulting in significantly enhanced computational efficiency with minimal to no degradation in control performance. Comparative experiments are conducted to validate the efficacy of DeePC in the control of the fabricated soft robot.**

***Index Terms*—Modeling, control, and learning for soft robots, soft sensors and actuators, data-driven control, predictive control.**

I. Introduction

SOFT robots have gathered significant attention in recent years due to their appealing features such as flexibility, adaptability, and versatility [1]. Such robots are fabricated using soft materials like fabrics, elastic polymers, among others [1], [2], while their actuation can be achieved through pneumatic, fluids, cables, or magnetics [3]. Therefore, the inherent flexibility and adaptability render them better candidates, as compared to rigid robots, for operation in complex environments with human interaction or fragile objects.

However, due to the intrinsic high complexity and nonlinearity of soft robots, their modeling and control have presented significant challenges. Numerical approaches, such as finite element analysis [4], are accurate but not suitable for real-time control due to complex coupled dynamics and high computational overhead. Approximated models of soft robots, on the other hand, often rely on the piecewise constant curvature (PCC) assumption. The PCC-based model [5] can streamline the system depiction with manageable computational complexity. This model finds extensive application in soft continuum robots, where its margin of error remains acceptable under conditions of relatively small payloads. Additionally, when the robot operates in a horizontal plane, gravitational effects can be neglected. Under the constant curvature assumption, both static and dynamic models can be derived. The PCC technique is well-established in quasi-static modeling and control [6]; however, it only considers kinematic relationships, making it suitable for scenarios characterized by slow dynamics or static environments. A dynamic model is required for dynamic tasks and interactions with the environment. Dynamic modeling approaches for soft robots include augmented rigid links [7] and Lagrangian approaches [8], with the majority requiring extensive experiments for the identification of model parameters. Nevertheless, it is empirically challenging and time-/cost- intensive to identify the model parameters of soft robots.

To address the complex nature of modeling and control of highly nonlinear systems like soft robots, there is a growing interest in data-driven control. Some researchers have focused on reinforcement learning for continuum robot control [9]. Meanwhile, Koopman-based methods have also been investigated [10], [11], which employs the Koopman operator to lift the lower-dimensional nonlinear states to a high-dimensional linear space, yielding control-oriented linear models of nonlinear systems. Compared to Koopman-based approaches that require empirical selection of observable functions and full state information, another emerging technique is **d**ata-**e**nabl**e**d **p**redictive **c**ontrol (DeePC) [12] which only needs input and output data. DeePC leverages the fundamental lemma [13] to describe system trajectories in a non-parametric manner. DeePC has demonstrated its efficacy in predicting system behavior without relying on explicit system models, and it has been successfully applied in domains such as power systems [14], quadcopters [15], and vehicle platooning [16], [17]. However, despite its appealing features, the application of DeePC to soft robots remains unexplored. The ability to obtain a non-parametric representation from data trajectories makes DeePC an ideal candidate for soft robot systems with highly nonlinear and stochastic dynamics.

Therefore, the paper focuses on the application of the DeePC approach to a soft robot. The main contributions are threefold.

Manuscript received 19 March 2024; accepted 11 July 2024. Date of publication 29 July 2024; date of current version 1 August 2024. This article was recommended for pubbrlication by Associate Editor D. Shin and Editor C. Laschi upon evaluation of the reviewers' comments. This work was supported by the U.S. National Science Foundation Project under Grant ECCS-2024649 and Grant CMMI-2320698. *(Corresponding author: Zhaojian Li.)*

Huanqing Wang, Kaixiang Zhang, Keyi Zhu, and Zhaojian Li are with the Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: wanghu26@egr.msu.edu; zhangk64@egr.msu.edu; zhukeyi1@egr.msu.edu; lizhaoj1@egr.msu.edu).

Kyungjoon Lee and Jun Sheng are with the Department of Mechanical Engineering, University of California Riverside, Riverside, CA 92521 USA (e-mail: klee449@ucr.edu; jun.sheng@ucr.edu).

Yu Mei and Vaibhav Srivastava are with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: meiyu1@egr.msu.edu; vaibhav@egr.msu.edu).

Digital Object Identifier 10.1109/LRA.2024.3434929

First, we present the synergistic design of a cost-effective soft robot, and investigate the use of DeePC on this soft arm. While working on the specific soft arm, the approach is general and can be readily extended to other soft robots. Second, building upon our recent theoretical advancements in dimension reduction for DeePC [18], a singular value decomposition (SVD) based scheme is introduced to improve the computation efficiency when deploying DeePC on soft robots, which bridges the gap from theory to application. Third, comprehensive experiments are conducted to evaluate the performance of DeePC. The results show that the DeePC approach achieves better control performance compared to a model-based control scheme and another data-driven control method, and the SVD based scheme can significantly enhance the computation efficiency of DeePC with no to little performance degradation, making it computationally tractable and suitable for real-time control of soft robots.

The remainder of this paper is structured as follows. Section II provides an overview of the hardware design of the planar soft robot and pneumatic control board. Section III introduces the control objective and the DeePC approach for dynamic control of the soft robot. Section IV presents the experimental setup and results. Finally, Section V draws conclusions based on the findings.

II. Hardware Design

This section introduces the hardware modules of the soft robot, including the design and fabrication, along with an overview of the components constituting the pneumatic control board.

A. Planar Soft Robot Design and Fabrication

The design of the soft robot encompasses two planar bending chambers, and each chamber consists of three core components: a fabric sleeve, a pneumatic bladder, and two end caps. This design strategy ensures that the soft robot is both low-cost and easy to fabricate, utilizing off-the-shelf materials alongside 3D printed parts. The fabric sleeve is crafted from two specific types of fabrics: an anisotropic fabric (White Heavy Stretch High Elasticity Knit Elastic Band, Cisone), which permits stretching in the wale direction while restricting expansion in the course direction, and a strain-limiting fabric (1050 Denier Coated Ballistic Nylon Fabric, Magna Fabrics), designed to prevent stretching in all directions. The directions are indicated in Fig. 1(a).

During assembly, the anisotropic fabric is oriented to enable stretching along its length. Both fabrics are centrally overlaid and joined together using a straight stitch, employing a polyester string (UV Resistant High Strength Polyester Thread 69 T70 Size 210D/3, Selric). The excess material from the strain-limiting fabric is then folded over the edges and stitched along the existing stitch line, repeating this process thrice for added durability. This edge reinforcement is crucial to prevent delamination under higher pressures. Two such fabric sleeves are assembled in total.

Detailed illustrations of this fabrication process are depicted in Fig. 1. Each sleeve houses a latex bladder inside. Subsequently, a customized collar, manufactured using a 3D printer (Ender-3 Pro 3D Printer, Creality), connected to a pneumatic tube (4 mm OD × 2 mm ID, Uxcell) is inserted into both ends of the bladder. These collars are enclosed within customized, 3D-printed end caps which are further secured with zip-ties to guarantee stability during pressurization [19], [20]. To ensure a more consistent bending motion, the sleeves are oriented such that the anisotropic fabric faces outward from the center. The completed assembly is shown in Fig. 1(b). When one chamber is pressurized by compressed air, the chamber will bend due to a strain difference caused by the anisotropic fabric stretching while the strain-limiting fabric remains unchanged, leading to a bending motion. The planar bending capability of the soft robot is illustrated in Fig. 1(c).

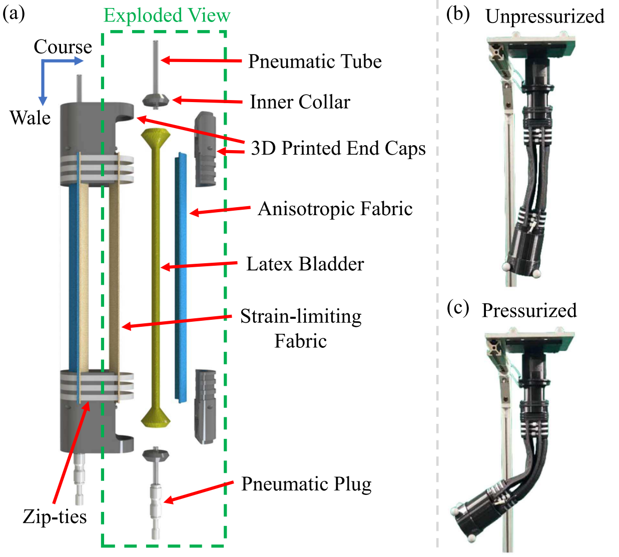

Fig. 1. Design of soft robot: (a) Exploded view of the components; (b) Robot in unpressurized state; and (c) Bending movement of robot when pressurized.

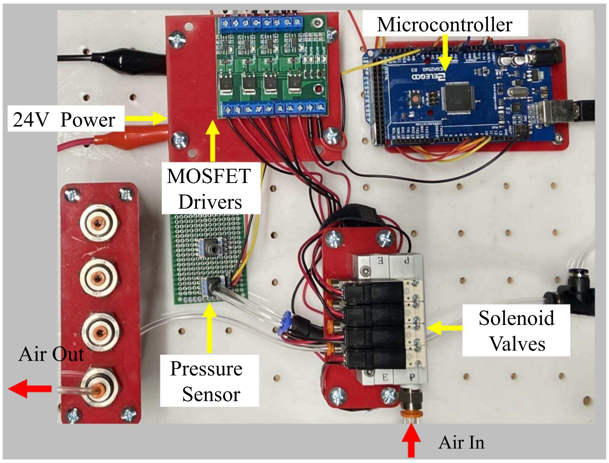

Fig. 2. Illustration of the pneumatic control board.

B. Pneumatic Control Board

The soft robot used is pneumatically driven, requiring an air source and an air control scheme. The air source is a portable pressure tank with a regulator output. The pneumatic control is referenced from the open soft robot toolkit [21], and the setup of the pneumatic control board is shown in Fig. 2. The Arduino

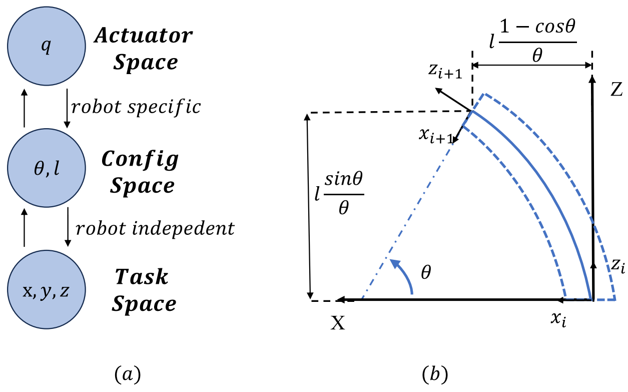

Fig. 3. (a) The three spaces of constant-curvature robots. (b) Configuration space parameters of PCC representation for a single segment.

acts as the low-level microcontroller to control the fast-switching solenoid valves (SMC VQ110U-5 M) through MOSFET drivers powered by 24 V. The air supply is from an air compressor and is directed to the manifold. Then, the electronic solenoid runs at a high frequency and adjusts the amount of air by regulating its duty cycle. The chamber pressure is measured by a pressure sensor (Honeywell SSCDANN100PGAA5) and fed back into the Arduino. The pneumatic control board can act as the low-level controller that receives pressure commands from the high-level controller and generates the necessary output to actuate the soft robot.

III. Control Objective and DeePC Formulation

In the following, we first state the control objectives for the soft robot in Section III-A. Section III-B presents the DeePC approach, illustrating its application to address the considered control problem. To enhance computational efficiency, Section III-C further incorporates a dimension reduction technique into the DeePC framework.

A. Soft Robot Control Problem

The widely embraced modeling approach for soft robots is based on the piecewise constant curvature (PCC) assumption. In this approach, the robot's configuration is considered as consisting of a certain number of segments, each characterized by a constant curvature. As illustrated in Fig. 3(a), the kinematics of a soft robot can be decomposed into two mappings among three spaces under the PCC assumption [6]. The first mapping is from actuator space to configuration space and the second one is from configuration space to task space. Specifically, actuator variables are often characterized by robot-specific properties such as cable length and air pressure. Configuration spaces consist of arc parameters that describe the configuration of the soft robot, which determines the task space position by geometric relationship. As shown in Fig. 3(b), the configuration space of a segment can be described by parameters (θ, ℓ), where θ is the bending angle, and ℓ is the arc length. The task space is generally defined with Cartesian coordinates, with the tip position of the segment being denoted by (p_x, p_z). The relationship between (θ, ℓ) and (p_x, p_z) can be described by $p_x = \ell \cdot \frac{1-\cos(\theta)}{\theta}$, $p_z = \ell \cdot \frac{\sin(\theta)}{\theta}$. The homogeneous transformation matrix between the robot tip and the robot base can be written as follows [22]:

$$H = \begin{bmatrix} \cos(\theta) & \sin(\theta) & \ell\frac{1-\cos(\theta)}{\theta} \\ -\sin(\theta) & \cos(\theta) & \ell\frac{\sin(\theta)}{\theta} \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

In the PCC approach, the kinematics of each segment can be formulated by (1). By aggregating all segments, the kinematics of the whole soft robot can be obtained. Subsequently, the dynamic model can be further developed through various techniques based on the kinematic relationship [8], [23]. However, the PCC model exhibits certain inherent limitations or flaws. The constant curvature assumption may not adapt well to real-world robotic scenarios, considering factors such as gravity effects, tip payload, or material characteristics. Additionally, imperfections in the manufacturing process could also cause non-homogeneity, making the accuracy of the constant curvature assumption suffer. Therefore, in this paper, we aim to leverage recent advancements in **d**ata-**e**nabl**e**d **p**redictive **c**ontrol (DeePC) [12] to achieve optimal control of the soft robot without relying on an explicit system model. Different from PCC approaches, DeePC directly uses input/output measurements to learn the system behavior, eliminating the need for the intricate transformation between the configuration space and task space required by PCC approaches.

B. Data-Enabled Predictive Control

We now provide an overview of DeePC. For more details, please refer to [12]. Through linearization around the operating point, the dynamic model of the soft robot can be approximated by the following discrete-time linear time-invariant (LTI) system (we will handle the inherent nonlineariy in the sequel):

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{2}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$ are system matrices, and $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$ are, respectively, the state, control input, and output at time $t \in \mathbb{Z}$. We purposefully abstract notation above to highlight the fact that the problem statement can be applied to different soft robotic systems with nonlinear dynamics whose linearization about the operating point is a controllable and observable LTI system. The DeePC algorithm utilizes pre-collected input/output data to describe the system behavior based on Willems's fundamental lemma [13], and the following definition of persistent excitation is needed.

Definition 1: Let $L, T \in \mathbb{Z}$ and $L \le T$. The Hankel matrix of depth L for the signal sequence $\omega_{[0,T-1]} := [\omega^\top(0), \omega^\top(1), \ldots, \omega^\top(T-1)]^\top$ is defined as

$$\mathcal{H}_L(\omega_{[0,T-1]}) := \begin{bmatrix} \omega(0) & \omega(1) & \cdots & \omega(T-L) \\ \omega(1) & \omega(2) & \cdots & \omega(T-L+1) \\ \vdots & \vdots & \ddots & \vdots \\ \omega(L-1) & \omega(L) & \cdots & \omega(T-1) \end{bmatrix}. \tag{3}$$

The sequence $\omega_{[0,T-1]}$ is said to be *persistently exciting of order* L if $\mathcal{H}_L(\omega_{[0,T-1]})$ has full row rank.

The DeePC algorithm begins by collecting an input/output data sequence of length T from the system:

$$\begin{aligned} u^{\mathrm{d}}_{[0,T-1]} &:= \left[u^{\mathrm{d}}(0)^{\mathsf{T}}, u^{\mathrm{d}}(1)^{\mathsf{T}}, \ldots, u^{\mathrm{d}}(T-1)^{\mathsf{T}}\right]^{\mathsf{T}}, \\ y^{\mathrm{d}}_{[0,T-1]} &:= \left[y^{\mathrm{d}}(0)^{\mathsf{T}}, y^{\mathrm{d}}(1)^{\mathsf{T}}, \ldots, y^{\mathrm{d}}(T-1)^{\mathsf{T}}\right]^{\mathsf{T}}. \end{aligned} \tag{4}$$

Then, the Hankel matrices $\mathcal{H}_L(u^{\mathrm{d}}_{[0,T-1]})$ and $\mathcal{H}_L(y^{\mathrm{d}}_{[0,T-1]})$ are given by

$$\begin{bmatrix} \mathcal{H}_L(u^{\mathrm{d}}_{[0,T-1]}) \\ \mathcal{H}_L(y^{\mathrm{d}}_{[0,T-1]}) \end{bmatrix} := \left[\begin{array}{cccc} u^{\mathrm{d}}(0) & u^{\mathrm{d}}(1) & \ldots & u^{\mathrm{d}}(T-L) \\ \vdots & \vdots & \ddots & \vdots \\ u^{\mathrm{d}}(L-1) & u^{\mathrm{d}}(L) & \ldots & u^{\mathrm{d}}(T-1) \\ \hline y^{\mathrm{d}}(0) & y^{\mathrm{d}}(1) & \ldots & y^{\mathrm{d}}(T-L) \\ \vdots & \vdots & \ddots & \vdots \\ y^{\mathrm{d}}(L-1) & y^{\mathrm{d}}(L) & \ldots & y^{\mathrm{d}}(T-1) \end{array}\right]. \tag{5}$$

The Willems' fundamental lemma aims to represent (2) via the input/output data $u^{\mathrm{d}}_{[0,T-1]}$ and $y^{\mathrm{d}}_{[0,T-1]}$.

Lemma 1 (Fundamental Lemma [13]): Consider a controllable LTI system (2) and assume that the input sequence $u^{\mathrm{d}}_{[0,T-1]}$ is persistently exciting of order $n+L$. Then, any length-L sequence $(u_{[0,L-1]}, y_{[0,L-1]})$ is an input/output trajectory of (2) if and only if we have

$$\begin{bmatrix} u_{[0,L-1]} \\ y_{[0,L-1]} \end{bmatrix} = \begin{bmatrix} \mathcal{H}_L(u^{\mathrm{d}}_{[0,T-1]}) \\ \mathcal{H}_L(y^{\mathrm{d}}_{[0,T-1]}) \end{bmatrix} g \tag{6}$$

for some real vector $g \in \mathbb{R}^{(T-L+1)}$.

Lemma 1 reveals that if the pre-collected input/output data is sufficiently long and rich, then all valid length-L trajectories of (2) can be generated via its corresponding Hankel matrices. This lemma demonstrates a non-parametric representation of system behavior, which is the key to formulate DeePC. We now introduce the DeePC formulation. Let $T_{\text{ini}}, N \in \mathbb{Z}$ be the time length of "past data" and "future data", respectively, and $T_{\text{ini}} + N = L$. The Hankel matrices $\mathcal{H}_L(u^{\mathrm{d}}_{[0,T-1]})$ and $\mathcal{H}_L(y^{\mathrm{d}}_{[0,T-1]})$ are divided into two parts:

$$\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix} := \mathcal{H}_L(u^{\mathrm{d}}_{[0,T-1]}), \quad \begin{bmatrix} Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} = \mathcal{H}_L(y^{\mathrm{d}}_{[0,T-1]}), \tag{7}$$

where U_{p} and U_{f} consist of the first T_{ini} block rows (i.e., the "past" data section) and the last N block rows (i.e., the "future" data section) of $\mathcal{H}_L(u^{\mathrm{d}}_{[0,T-1]})$, respectively (similarly for Y_{p} and Y_{f}). Let $u_{\text{ini}} = u_{[t-T_{\text{ini}},t-1]}$ be the control input sequence within a past time horizon of length T_{ini}, and $u = u_{[t,t+N-1]}$ be the control input sequence within a prediction horizon of length N (similarly for y_{ini} and y). At each time step t, the DeePC employs Lemma 1 to predict future system behavior and solves the following constrained optimization problem:

$$\begin{aligned} \min_{g,u,y} \quad & \|y - y_r\|_Q^2 + \|u\|_R^2 \\ \text{subject to} \quad & \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ u \\ y_{\text{ini}} \\ y \end{bmatrix}, u \in \mathcal{U}, y \in \mathcal{Y}, \end{aligned} \tag{8}$$

where $y_r = \left[y_r^{\mathsf{T}}(t), y_r^{\mathsf{T}}(t+1), \ldots, y_r^{\mathsf{T}}(t+N-1)\right]^{\mathsf{T}}$ is a desired trajectory, Q, R are weighting matrices, $\|y - y_r\|_Q^2 := (y - y_r)^\top Q (y - y_r)$, $\|u\|_R^2 := u^\top R u$, and $\mathcal{U}$, $\mathcal{Y}$ represent the input and output constraints.

Note that the formulation (8) can achieve satisfactory performance if noise-free data is collected from the deterministic LTI system (2). As discussed in [12], [24], in the presence of output measurement noise or system nonlinearities, slack variables and regularization design can be introduced to extend the DeePC algorithm. Motivated by this, a slack variable, denoted as σ_y, is introduced to ensure the feasibility of the equality constraint, and the regularized version of DeePC can be formulated as

$$\begin{aligned} \min_{g,u,y,\sigma_y} \quad & \|y - y_r\|_Q^2 + \|u\|_R^2 + \lambda_y \|\sigma_y\|_2^2 + \lambda_g \|g\|_2^2 \\ \text{subject to} \quad & \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ u \\ y_{\text{ini}} \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sigma_y \\ 0 \end{bmatrix}, u \in \mathcal{U}, y \in \mathcal{Y}. \end{aligned} \tag{9}$$

In (9), the slack variable σ_y is subjected to a weighted quadratic norm penalty function. The weight coefficient $\lambda_y > 0$ can be chosen sufficiently large to ensure that $\sigma_y \neq 0$ only when the equality constraint is infeasible [12], [15]. Moreover, a quadratic norm penalty is applied to g with a weight coefficient $\lambda_g > 0$. Different choices of regularization for g can be incorporated into the DeePC framework, each with unique characteristics. For instance, the one-norm regularizer [12] can serve as a surrogate for low-rank pre-processing, while the projection-based regularizer [25] is consistent without biasing the solution obtained for perfect data. In this paper, quadratic regularization, i.e., $\lambda_g \|g\|_2^2$, is implemented as it leads to robust and optimal solutions with regards to bounded disturbance on the input/output data [26], and it renders the cost function quadratic. For a more detailed discussion about regularizers in DeePC, interested readers can refer to [27].

The formulation (9) is applicable to nonlinear and nondeterministic soft robotic systems. The application of DeePC for the soft robot is detailed in Algorithm 1. Initially, when the algorithm starts, u_{ini} and y_{ini} are empty. u_{ini} is then set as 0, while y_{ini} is filled with real-time position data. As the algorithm progresses, when $k > T_{\text{ini}}$, DeePC optimization begins. The optimal g is solved based on the formulation in (9), and the iteration window moves until the end.

C. Dimension Reduction for DeePC

Note that in Lemma 1, ensuring the persistent excitation of the sequence $u^{\mathrm{d}}_{[0,T-1]}$ requires that column number of the Hankel

Algorithm 1: DeePC Algorithm for Soft Robot.

```
**Input:** iteration number $t_{\text{end}}$, pre-collected pressure
input $u^{\text{d}}_{[0,T-1]}$ and soft robot position output $y^{\text{d}}_{[0,T-1]}$.
Construct Hankel matrices $U_{\text{p}}$, $U_{\text{f}}$, $Y_{\text{p}}$, $Y_{\text{f}}$.
For $t < T_{\text{ini}}$, initialize $u_{\text{ini}}$ with 0 and $y_{\text{ini}}$ with
real-time position measurements.
**while** $T_{\text{ini}} \leq t \leq t_{\text{end}}$ **do**
    Solve the optimization problem(9) for $g$ and obtain
    optimal pressure control $u = U_{\text{f}} g$.
    Send the first step optimal pressure control $u(1)$ to
    low-level pneumatic controller.
    Measure the soft robot position, and update $u_{\text{ini}}$ and
    $y_{\text{ini}}$ to the $T_{\text{ini}}$ most recent input/output
    measurements.
    Set $t$ to $t + 1$.
**end while**
```

matrix $\mathcal{H}_{n+L}(u^{\text{d}}_{[0,T-1]})$ to be no less than its row number. This condition implies that the number of data points T must satisfy $T - (n+L) + 1 \geq m(n+L)$, i.e., $T \geq (m+1)(n+L) - 1$. Thus, the dimension of g in (9) is lower bounded as

$$T - L + 1 \geq mL + (m+1)n. \tag{10}$$

This inequality indicates that a large T results in a high dimension for the optimization variable g in (9). In practical scenarios, it becomes necessary to collect a sufficiently extensive dataset to satisfy $T \geq (m+1)(n+L) - 1$. However, this choice often leads to a very large value of T, making the optimization problem (9) large-scale and nontrivial to solve efficiently.

Considering the aforementioned issues, we leverage our recently proposed singular value decomposition (SVD) based approach [18] to reduce the dimension of optimization variables in DeePC, thereby enhancing the computation efficiency. Note that SVD techniques are widely used in system identification methods but for different purposes. In system identification, SVD is applied to identify an explicit parametric model, which can then be incorporated into model-based control frameworks. In contrast, our SVD-based approach focuses on extracting the core features from the original Hankel matrix (i.e., the collected data) to create a new data matrix with a reduced column dimension. This downsized data matrix is then introduced into DeePC, leading to a substantial reduction in the dimension of optimization variables without compromising control performance.

After collecting the input/output data sequences $u^{\text{d}}_{[0,T-1]}$, $y^{\text{d}}_{[0,T-1]}$ and constructing the Hankel matrices in (5), we apply the SVD technique to compute a new data library. Specifically, the SVD of the Hankel matrix (5) has the following form:

$$\begin{bmatrix} \mathcal{H}_L(u^{\text{d}}_{[0,T-1]}) \\ \mathcal{H}_L(y^{\text{d}}_{[0,T-1]}) \end{bmatrix} = \underbrace{\begin{bmatrix} W_1 & W_2 \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} V_1 & V_2 \end{bmatrix}^{\mathsf{T}}}_{V^{\mathsf{T}}}. \tag{11}$$

In (11), $W \in \mathbb{R}^{q_1 \times q_1}$ and $V \in \mathbb{R}^{q_2 \times q_2}$ are orthogonal matrices, where $q_1 = (m+p)L$ and $q_2 = T - L + 1$. $\Sigma \in \mathbb{R}^{q_1 \times q_2}$ is a rectangular diagonal matrix with non-negative singular values on the diagonal. The singular values in Σ are arranged in descending order. Let $\Sigma_1 \in \mathbb{R}^{r \times r}$ $(r \leq \min\{q_1, q_2\})$ contain the top r non-zero singular values. Then, Σ_2, W_1, W_2, V_1, and V_2 are defined with appropriate dimensions. The new data matrix $\bar{\mathcal{H}}_L \in \mathbb{R}^{q_1 \times r}$ is constructed via

$$\bar{\mathcal{H}}_L = \mathcal{H}_L V_1 = W_1 \Sigma_1. \tag{12}$$

Given $\bar{\mathcal{H}}_L$, the DeePC problem in (9) can be replaced by

$$\begin{aligned} \min_{\bar{g}, u, y, \sigma_y} \quad & \|y - y_r\|_Q^2 + \|u\|_R^2 + \lambda_y \|\sigma_y\|_2^2 + \lambda_g \|\bar{g}\|_2^2 \\ \text{subject to} \quad & \bar{\mathcal{H}}_L \bar{g} = \begin{bmatrix} u_{\text{ini}} \\ u \\ y_{\text{ini}} \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sigma_y \\ 0 \end{bmatrix}, u \in \mathcal{U}, y \in \mathcal{Y}. \end{aligned} \tag{13}$$

If Σ_1 contains all non-zero singular values (i.e., r is equal to the rank of the Hankel matrix (5)), then the new data matrix $\bar{\mathcal{H}}_L$ has the same range space as the Hankel matrix (5). For systems beyond deterministic LTI with noises, the Hankel matrix (5) is typically full rank, and selecting r equal to the rank of the Hankel matrix (5) is unnecessary. In such cases, r can be selected based on the distribution of singular values. An appropriate selection of r can be guided by identifying the turning point in the singular value distribution, which indicates the transition from singular vectors representing the principal patterns to those that are insignificant [18]. If the turning point is not clear or applicable, the minimum dimension can be determined experimentally by gradually reducing the size of r until the control performance becomes unsatisfactory.

In (9), the optimization variable g has a dimension of $T - L + 1$, whereas in (13), the dimension of the optimization variable $\bar{g}$ has been reduced to r. In practical applications, T needs to be sufficiently large to ensure that the collected data is rich enough for the non-parametric representation. The dimension reduction scheme leads to a much condensed data matrix $\bar{\mathcal{H}}_L$, which can significantly improve the computation efficiency of DeePC.

IV. Experiments

In this section, the experimental study evaluating the performance of DeePC on the soft robot system is presented. A video demonstration of the experimental test for DeePC is available at https://youtu.be/yTN6vKkndDg.

A. Setup and Data Collection

The experimental setup is shown in Fig. 4. Specifically, the soft robot hangs upside down within the workspace of the Qualisys motion capture system. The Qualisys motion capture system consists of eight cameras, which can provide accurate positions of the soft robot. The position information is transmitted from the Qualysis server to the DeePC host PC through an Ethernet interface. The DeePC algorithm, implemented in Matlab, operates in real-time on the host PC, and the optimal pressure command is sent to a lower-level controller via the publisher-subscriber communication protocol of the robotic operating system. The low-level pneumatic control board employs a PID controller

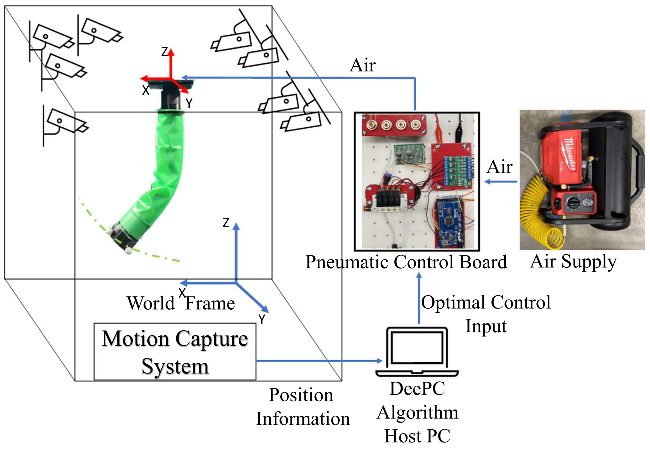

Fig. 4. The experimental setup for the soft robot control.

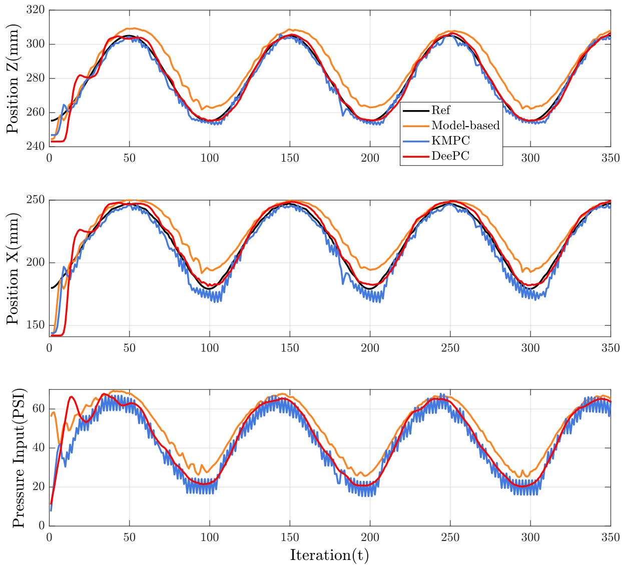

Fig. 5. The tracking results comparison between DeePC, KMPC, and the model-based control method.

to achieve the desired pressure by regulating the duty cycle of the fast-switching values. The air from the portable tank passes through the control board and finally enters the soft robot. For this experiment, only one chamber of the soft robot is actuated.

In order to construct the non-parametric representation of the soft robot, input/output data, i.e., u^{d} and y^{d}, needs to be collected. For the considered soft robot, the control input u^{d} is defined as the pressure in the air chamber, and the output y^{d} is defined as the robot's coordinates (p_x, p_z) in task space. During data collection, random and open-loop step inputs are commanded to the low-level pressure controller. The soft robot is then actuated, and the corresponding coordinates are continuously recorded. Subsequently, the collected data set is organized into the Hankel matrix shown in (5). If the collected input/output data is sufficiently rich, then the constructed Hankel matrix can be used to achieve accurate predictions of system behavior, thereby mitigating biased results in DeePC. To evaluate the performance of the controller, sinusoidal trajectory tracking is performed, where the reference is given in both Z and X directions. The reference trajectories for the Z and X directions are shown as follows:

$$\begin{cases} Z_{\text{ref}} &= 25\sin(0.02\pi t) + 280, \\ X_{\text{ref}} &= f(Z_{\text{ref}}). \end{cases} \tag{14}$$

Since the soft robot is an underactuated system, its movements along the Z and X directions are coupled, preventing the independent selection of Z_{ref} and X_{ref}. In (14), the interpolation function f is introduced to approximate the coupling dynamics between the Z and X directions, ensuring that the reference trajectories are feasible for the soft robot. We use pre-collected system output trajectories and MATLAB function `interp1` to determine the interpolation function f. Then, for a given Z_{ref}, X_{ref} can be computed with $f(Z_{\text{ref}})$.

B. Benchmark Approaches

In this section, we present two benchmark approaches for comparison with our method.

The first one is a model-based control approach that relies on the PCC assumption. After obtaining the kinematic relationship between θ and the end-effector position from (1), the dynamic equation of the soft manipulator can be derived using the Euler-Lagrange formula [8], [23]. After the derivation of the Lagrangian formulation, the equation of motion can be put into the following form:

$$M(q)\ddot{q} + V(q,\dot{q}) + D(q)\dot{q} + G(q) + Kq = A(q)\tau. \tag{15}$$

The definitions of the parameters in (15) can be found in [7], [23]. Experimental data is recorded, and then system identification is performed to estimate best model parameters [7], [28]. A model-based control with feedback linearization is used for this application, and the control law is designed as follows:

$$\tau = M(q)\Bigg(\ddot{q}_{\mathrm{d}} + k_d(\dot{q}_{\mathrm{d}} - \dot{q}) + k_p(q_{\mathrm{d}} - q) \\ + k_i \int (q_{\mathrm{d}} - \dot{q})\, dt\Bigg) + h. \tag{16}$$

The term h is the sum of all the nonlinear terms from the Lagrange formulation, the details of the rest of the term can be found in [29].

The second benchmark approach is data-driven and relies on the Koopman operator. This approach builds a linear model using the Koopman operator, which is subsequently integrated into the MPC framework. Extended dynamic mode decomposition is exploited to approximate the Koopman operator and thereby determining a linear model in a lifted state space that approximates the nonlinear system. For more details on this approach, please refer to Algorithms 1 and 3 in [30]. In the following, we refer to this approach as KMPC. The prediction horizon for KMPC is selected to be the same as the value of $N = 45$ in DeePC.

C. Proposed Approach: DeePC

The DeePC algorithm is implemented on our test setup. The task space limits for the end effector of the soft robot are

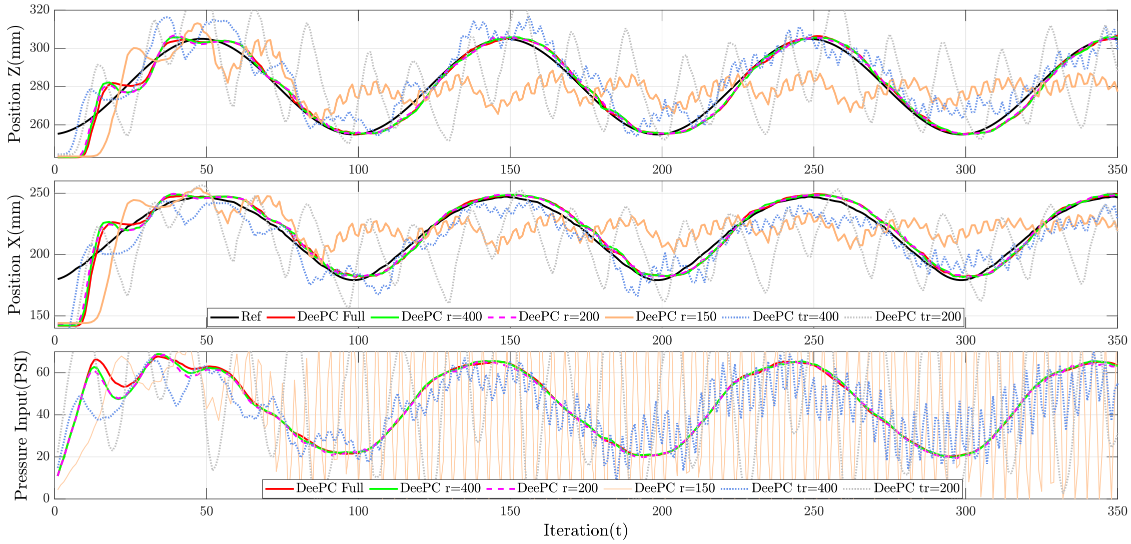

Fig. 6. The tracking results for full-size DeePC, truncated DeePC, and dimension-reduced DeePC.

$p_z \in [240, 330]$ mm and $p_x \in [140, 270]$ mm. The constraint for control input is $u \in [0, 90]$ PSI. The parameters for DeePC formulation in (9) and the dimension-reduced DeePC in (13) are chosen to be the same: $T_{\text{ini}} = 30$, $N = 45$, $Q = 0.1 \cdot I_{2\times 2}$, $R = 1 \times 10^{-5}$, $\lambda_g = 300$, $\lambda_y = 1000$. We summarize our general approach for parameter selection. For a detailed tuning procedure and analysis, please refer to [15]. The parameters N and T_{ini} directly affect the Hankel matrix size and thus influence computational time. Therefore, they are selected with a balance in mind between speed and performance. The matrices Q and R, representing state cost and control cost respectively, can be chosen similarly to how they are in the LQR/MPC controllers. The parameter λ_y, which weights the softened initial condition constraint, should be selected as large as possible without causing numerical issues. λ_g introduces robustness to the system in the presence of noisy data. If λ_g is too small, it can impact the tracking performance, due to prediction errors introduced by the system's nonlinearity. Conversely, if λ_g is too large, it increases robustness but also introduces conservativeness, which can degrade performance [26].

The tracking performance of DeePC, KMPC, and the model-based control method is illustrated in Fig. 5. The DeePC method consistently maintains commendable tracking performance. Additionally, the optimal pressure input commands exhibit smooth transitions, effectively reducing jittering behaviors in the pneumatic soft robot. The KMPC also tracks the trajectory relatively well, but its control input fluctuates more than DeePC, causing vibrations in the system output. The model-based control method shows more tracking errors and oscillations, as the derived model (15) cannot fully capture the actual system dynamics.

To demonstrate the effectiveness of the SVD based dimension reduction method introduced in Section III-C, we run the DeePC algorithm with three different data matrices: the full-size Hankel matrix (5) (denoted by $\mathcal{H}_L$), its direct truncation matrix $\mathcal{H}_{L,[1:tr]}$ (i.e., the first tr columns of $\mathcal{H}_L$ are extracted to construct the truncation matrix), and the new data matrix $\bar{\mathcal{H}}_L$ constructed with the SVD based method. The truncation matrix $\mathcal{H}_{L,[1:tr]}$ is evaluated at $tr = 400, 200$, and the data matrix $\bar{\mathcal{H}}_L$ is evaluated at $r = 400, 200, 150$. The results, shown in Fig. 6, indicate that the full-size Hankel matrix $\mathcal{H}_L$ and the data matrix $\bar{\mathcal{H}}_L$ with $r = 400, 200$ can be well incorporated into the DeePC algorithm to exhibit similar and satisfactory tracking performance. In comparison, DeePC with truncated Hankel matrix ($tr = 400, 200$) fails to track the reference. These results reveal that with an appropriate selection of r, the data matrix $\bar{\mathcal{H}}_L$ retains the essential range space of the original Hankel matrix, ensuring that DeePC performance is not compromised. At $r = 150$, as indicated by the light orange line, DeePC fails to maintain tracking accuracy, leading us to experimentally establish the minimum dimension at $r = 200$. In addition, the computation time of DeePC for different values of r is illustrated in Fig. 7, showing a nearly linear trend concerning the size of r. For real-time implementation, the computational time of DeePC must be shorter than the communication interval between the DeePC host PC and the low-level pneumatic control board. In our experiment, the communication frequency is 8 Hz, and we observe that a matrix size of 800 results in a computation time just below 0.12 seconds per iteration. Thus, the maximum column size is set as 800.

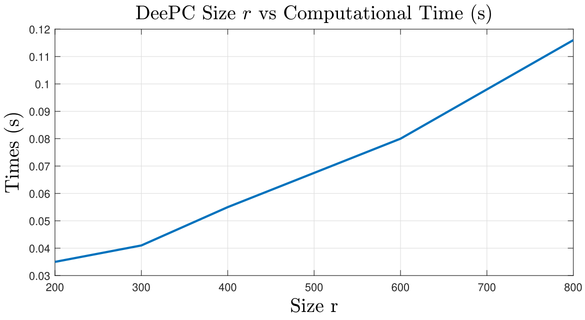

Fig. 7. DeePC size r presents a nearly linear relationship with computational time.

TABLE I
AVERAGE EUCLIDEAN DISTANCE ERROR(MM) COMPARISON BETWEEN DIFFERENT METHODS

	Dimension of $\mathcal{H}_L/\bar{\mathcal{H}}_L$	Euclidean Distance Error (mm)
Model-Based Control (16)	/	10.75
KMPC	/	4.77
Full-Size DeePC	225×800	4.29
Dimension-Reduced DeePC	225×600	4.49
	225×400	4.38
	225×300	4.47
	225×200	**4.20**

Finally, the average Euclidean distance errors of different approaches are calculated, and the results are summarized in Table I. Both KMPC and DeePC achieve much better tracking accuracy compared to the model-based control method. Different from KMPC that involves the approximating the Koopman operator, DeePC can conveniently obtain the non-parametric model through direct data collection. In addition, Table I shows that the SVD based method can effectively improve the computation efficiency of DeePC, while retaining satisfactory tracking accuracy.

V. CONCLUSION

This letter introduced a DeePC approach for dynamic control of planar soft robots. We first presented the system setup and then formulated the DeePC approach for addressing the soft robot control problem. Additionally, we incorporated a dimension reduction scheme into the DeePC framework to achieve faster computational times. The experimental study focused on dynamic tracking, where performance metrics were thoroughly analyzed. Furthermore, we compared the DeePC approach with a model-based approach and another data-driven control method. The experimental study demonstrated the validity and effectiveness of the DeePC approach for planar soft robots. Notably, this experiment focuses solely on the planar robot. In our future work, we plan to explore robots with more degrees of freedom and present real-world application cases.

REFERENCES

[1] M. S. Xavier et al., "Soft pneumatic actuators: A review of design, fabrication, modeling, sensing, control and applications," *IEEE Access*, vol. 10, pp. 59442–59485, 2022.

[2] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, "Soft robots modeling: A structured overview," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1728–1748, Jun. 2023.

[3] J. Wang and A. Chortos, "Control strategies for soft robot systems," *Adv. Intell. Syst.*, vol. 4, no. 5, 2022, Art. no. 2100165.

[4] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 3982–3987.

[5] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, 2010.

[6] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *J. Robotic Syst.*, vol. 20, no. 2, pp. 45–63, 2003.

[7] R. K. Katzschmann, C. D. Santina, Y. Toshimitsu, A. Bicchi, and D. Rus, "Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model," in *Proc. 2nd IEEE Int. Conf. Soft Robot.*, 2019, pp. 454–461.

[8] V. Falkenhahn, T. Mahl, A. Hildebrandt, R. Neumann, and O. Sawodny, "Dynamic modeling of constant curvature continuum robots using the Euler-Lagrange formalism," in *PRoc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2428–2433.

[9] T. Behr, T. P. Pusch, M. Siegfarth, D. Hüsener, T. Mörschel, and L. Karstensen, "Deep reinforcement learning for the navigation of neurovascular catheters," *Curr. Directions Biomed. Eng.*, vol. 5, no. 1, pp. 5–8, 2019.

[10] D. Bruder, B. Gillespie, C. D. Remy, and R. Vasudevan, "Modeling and control of soft robots using the Koopman operator and model predictive control," in *Proc. Robot. Sci. Syst.*, FreiburgimBreisgau, Germany, Jun. 2019, doi: 10.15607/RSS.2019.XV.060.

[11] J. Wang et al., "An improved Koopman-MPC framework for data-driven modeling and control of soft actuators," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 616–623, Feb. 2023.

[12] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *Proc. 18th Eur. Control Conf.*, 2019, pp. 307–312.

[13] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. D. Moor, "A note on persistency of excitation," *Syst. Control Lett.*, vol. 54, no. 4, pp. 325–329, 2005.

[14] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Decentralized data-enabled predictive control for power system oscillation damping," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 3, pp. 1065–1077, May 2022.

[15] E. Elokda, J. Coulson, P. N. Beuchat, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for quadcopters," *Int. J. Robust Nonlinear Control*, vol. 31, no. 18, pp. 8916–8936, Dec. 2021.

[16] J. Wang, Y. Zheng, K. Li, and Q. Xu, "DeeP-LCC: Data-enabled predictive leading cruise control in mixed traffic flow," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 6, pp. 2760–2776, Nov. 2023.

[17] K. Zhang, K. Chen, Z. Li, J. Chen, and Y. Zheng, "Privacy-preserving data-enabled predictive leading cruise control in mixed traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 5, pp. 3467–3482, May 2023.

[18] K. Zhang, Y. Zheng, C. Shang, and Z. Li, "Dimension reduction for efficient data-enabled predictive control," *IEEE Control Syst. Lett.*, vol. 7, pp. 3277–3282, 2023.

[19] T. Liu, T. Abrar, and J. Realmuto, "Modular and reconfigurable body mounted soft robots," in *Proc. IEEE 7th Int. Conf. Soft Robot.*, 2024, pp. 145–150.

[20] K. Lee, K. Bayarsaikhan, G. Aguilar, J. Realmuto, and J. Sheng, "Design and characterization of soft fabric omnidirectional bending actuators," *Actuators*, vol. 13, no. 3, 2024, Art. no. 112.

[21] D. P. Holland, E. J. Park, P. Polygerinos, G. J. Bennett, and C. J. Walsh, "The soft robotics toolkit: Shared resources for research and design," *Soft Robot.*, vol. 1, no. 3, pp. 224–230, 2014.

[22] C. D. Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: Trajectory tracking and interaction with the environment," *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 490–513, 2020.

[23] Y. Mei, P. Fairchild, V. Srivastava, C. Cao, and X. Tan, "Simultaneous motion and stiffness control for soft pneumatic manipulators based on a Lagrangian-based dynamic model," in *Proc. Amer. Control Conf.*, 2023, pp. 145–152.

[24] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1702–1717, Apr. 2021.

[25] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging direct and indirect data-driven control formulations via regularizations and relaxations," *IEEE Trans. Autom. Control*, vol. 68, no. 2, pp. 883–897, Feb. 2023.

[26] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, "Quadratic regularization of data-enabled predictive control: Theory and application to power converter experiments," *IFAC-PapersOnLine*, vol. 54, no. 7, pp. 192–197, 2021.

[27] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annu. Rev. Control*, vol. 52, pp. 42–64, 2021.

[28] C. D. Santina, R. K. Katzschmann, A. Biechi, and D. Rus, "Dynamic control of soft robots interacting with the environment," in *Proc. IEEE Int. Conf. Soft Robot.*, 2018, pp. 46–53.

[29] M. Azizkhani, A. L. Gunderman, I. S. Godage, and Y. Chen, "Dynamic control of soft robotic arm: An experimental study," *IEEE Robot. Automat. Lett.*, vol. 8, no. 4, pp. 1897–1904, Apr. 2023.

[30] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Data-driven control of soft robots using Koopman operator theory," *IEEE Trans. Robot.*, vol. 37, no. 3, pp. 948–961, Jun. 2021.