

Direct Data-Driven Predictive Control for a Three-Dimensional Cable-Driven Soft Robotic Arm

Cheng Ouyang¹, Moeen Ul Islam¹, Dong Chen¹, Kaixiang Zhang², Zhaojian Li², and Xiaobo Tan³

Abstract—Soft robots offer significant advantages in safety and adaptability, yet achieving precise and dynamic control remains a major challenge due to their inherently complex and nonlinear dynamics. Recently, Data-enabled Predictive Control (DeePC) has emerged as a promising model-free method that bypasses explicit system identification by directly leveraging input–output data. While DeePC has shown success in other domains, its application to soft robots remains underexplored, particularly for 3D soft robotic systems. This paper addresses this gap by developing and experimentally validating an effective DeePC framework on a 3D cable-driven soft arm. Specifically, we design and fabricate a soft robotic arm with a thick tubing backbone for stability, a dense silicone body with large cavities for strength and flexibility, and rigid endcaps for secure termination. Using this platform, we implement DeePC with singular value decomposition (SVD)-based dimension reduction for two key control tasks: fixed-point regulation and trajectory tracking in 3D space. Comparative experiments with a baseline model-based controller demonstrate DeePC’s superior accuracy and adaptability, highlighting its potential as a practical solution for dynamic control of soft robots.

Index Terms—Data-enabled predictive control, soft robotics, trajectory tracking

I. INTRODUCTION

Soft robots, enabled by compliant materials and novel morphologies, have attracted increasing research interest across diverse applications, such as fruit harvesting [1], medical procedures [2], and beyond [3]. Compared to conventional rigid robots, they offer safer human–robot interaction [4], [5], greater adaptability to unstructured environments [6], and theoretically unlimited degrees of freedom for versatile manipulation [7].

However, this promise comes with profound control challenges. The very sources of soft robots’ advantages, deformable materials and continuum structures, also lead to extreme complexity in modeling and control [8]. A primary difficulty lies in their highly nonlinear dynamics: materials such as silicone exhibit nonlinear stress–strain behavior, and under pneumatic, electrical, or tendon actuation, their responses often involve strong nonlinearity, hysteresis, and creep [9]. Another challenge is the infinite-dimensional state space. Unlike rigid robots, whose configurations can be

described by a finite set of joint angles and velocities, a continuously deformable body theoretically requires specifying the displacement of every point, resulting in an infinite-dimensional system. Collectively, these factors make soft robot control particularly challenging [10].

To address the complex dynamics of soft robots, researchers have proposed numerous model-based control strategies; however, these methods often struggle to balance accuracy and computational complexity [9]. For instance, the Piecewise Constant Curvature (PCC) model is the most widely used simplification [11]. It assumes each section bends as a circular arc with constant curvature, reducing the configuration to only a few parameters. Yet, PCC’s validity depends on this assumption, which is frequently violated in practice due to gravity, environmental interactions, and non-uniform actuation or materials, resulting in significant model–plant mismatch and degraded performance [12]. The finite element method (FEM) offers a more accurate alternative, capable of capturing complex geometries, nonlinear material properties, and contact interactions between soft robots and their environments. However, this accuracy comes at the cost of high computational demands [13]. To address this, Navez et al. proposed a condensed FEM model that combines the efficiency of machine learning with the physical interpretability of FEM-based mechanics [14]. While promising, it relies on quasi-static assumptions, requires extensive FEM-generated training data, and shows limited generalization to highly nonlinear or dynamic scenarios.

Given the limitations of first-principles modeling, data-driven methods provide a promising alternative for soft robot control. Instead of deriving exact physical models, these approaches learn input–output mappings or latent dynamics directly from experimental data. Reinforcement learning (RL) offers an appealing model-free control strategy for systems that are difficult to model; however, its use is hindered by low sample efficiency, the simulation-to-reality gap, and safety concerns associated with exploration [15], [16]. Koopman operator theory provides a distinct, data-driven approach. Unlike RL, it is not model-free; rather, it identifies an explicit linear model from data that represents the dynamics of the underlying nonlinear system. Bruder et al. [17] demonstrated a Koopman-based method that enabled real-time trajectory tracking of soft robots with high computational efficiency. However, its effectiveness relies heavily on the choice of basis functions, and for more complex systems, the model dimension can become computationally intractable.

In contrast to RL’s fully model-free nature and Koopman operator theory’s explicit modeling, data-enabled predictive

¹Cheng Ouyang, Moeen Ul Islam, and Dong Chen are with the Department of Agricultural & Biological Engineering, Mississippi State University, MS, USA. Emails: {co603, mul136, dc2528}@msstate.edu

²Kaixiang Zhang and Zhaojian Li are with the Department of Mechanical Engineering, Michigan State University, MI, USA. Emails: {zhangk64, lizhaoj1}@msu.edu

³Xiaobo Tan is with the Department of Electrical and Computer Engineering, Michigan State University, MI, USA. Email: xbtan@egr.msu.edu

control (DeePC) adopts a nonparametric, model-implicit approach by leveraging the Hankel matrix of past input–output data as its predictive model [18]. DeePC has been successfully applied in several domains, including large-scale damping control in power systems [19], building automation [20], quadrotor trajectory tracking [21], and power-converter control [22]. Its robustness, ability to handle nonlinearity and uncertainty, and nonparametric nature align closely with the challenges of soft robot control [23], yet related applications remain at an early stage. Wang et al. were the first to apply DeePC to soft robots [24], but their validation was limited to a planar, pneumatically actuated system restricted to one plane of motion. Extending DeePC to three-dimensional soft arms and validating it in high-precision 3D dynamic tasks thus remains a key open problem. To address this gap, we build on our previous work on a modular cable-driven soft robotic arm, for which we developed a nonlinear kinematic model and demonstrated open-loop control [25]. This study advances that foundation by (i) optimizing the robotic arm design, (ii) introducing a DeePC framework with singular value decomposition (SVD)-based dimension reduction, systematically implemented and experimentally validated on the developed soft robotic arm platform for two control tasks: fixed-point regulation and dynamic trajectory tracking in 3D space, and (iii) open-sourcing the arm designs and control codes to facilitate future research¹.

II. HARDWARE DESIGN

This section presents the hardware design of the soft robot, including its structural configuration, fabrication process, and control modules.

A. Robot design and fabrication

To apply DeePC, we initially tested our previous robot design in [25]. However, through repeated testing and several design iterations, we found that practical applications require greater stiffness and load-bearing capacity while preserving the inherent flexibility of soft robots. To address this, the final design employed a thicker flexible backbone and denser silicone, providing the necessary structural strength while maintaining compliance. The resulting soft robotic arm consists of two end caps, a flexible backbone, and three helix-like spiral structures that incorporate Kevlar threads for cable routing. The detailed structure and sequential fabrication process are shown in Fig. 1a–1c, while the final cured soft robot is presented in Fig. 1d. The end caps (Fig. 1a) were 3D-printed using PETG, with anchor points designed to prevent detachment of the silicone after casting, and further shaped to support the mounting and stacking of multiple arm segments. A flexible backbone was formed using clear Masterklear soft PVC tubing (McMaster-Carr) with an outer diameter of 8 mm and an inner diameter of 3 mm. The molds for silicone casting (Fig. 1c) were also fabricated via 3D printing, and Ecoflex 00-50 silicone was used as the elastomer material. High-strength Kevlar threads served dual

purposes: as actuation cables and as reinforcement in the helix-like spiral structures.

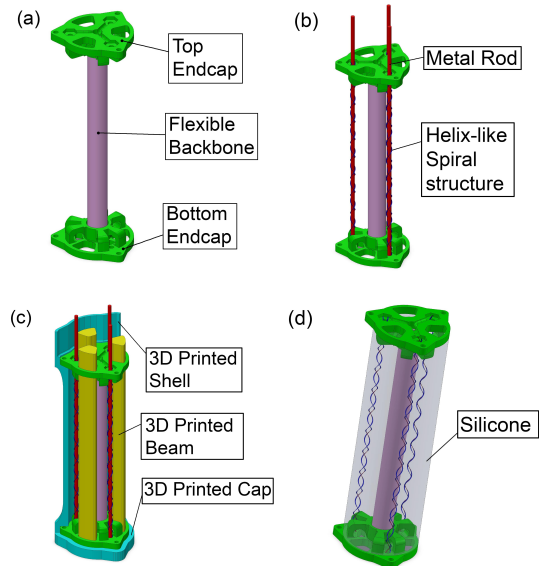


Fig. 1: Fabrication process of the soft robot: (a–c) sequential setup steps prior to silicone casting, and (d) the completed soft robot after curing. Each panel is labeled to indicate the corresponding components.

B. Actuation and control system

The soft robot arm is cable-driven, with cable lengths precisely regulated by three stepper motors. Each actuation cable is driven by a 3D-printed pulley attached directly to a stepper motor. The three stepper motors are mounted on a 3D-printed frame at 120° separation. The frame also incorporates dedicated cable guides that route each actuation cable to its corresponding stepper motor. This design prevents tangling and ensures the cables remain properly seated within the pulleys. The overall mechanical structure of the system is shown in Fig. 2a.

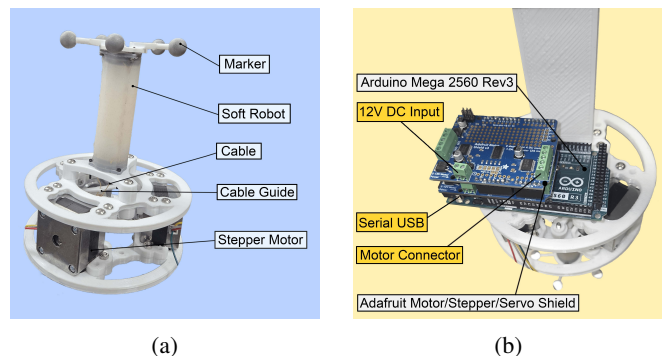


Fig. 2: Hardware setup of the soft robot, illustrating (a) the actuation system and (b) the low-level control system.

For motor control, an Arduino Mega 2560 Rev3 serves as the low-level microcontroller. It drives the stepper motors via a motor driver shield (Adafruit Motor Shield V2) and operates in open-loop mode without encoder feedback. The complete control board is shown in Fig. 2b. The low-level

¹Codes and designs: <https://github.com/chengoy30/DeePC-for-Soft-Robot-Control>

controller receives commands from a high-level controller (a laptop) through serial communication, while the high-level controller computes the optimal control inputs and generates the corresponding commands.

III. DEEPC FOR SOFT ROBOT CONTROL

A. Data-enabled Predictive Control (DeePC)

We begin with an overview of DeePC [18]. By linearizing the system dynamics around a nominal operating point, the behavior of a soft robot can be approximated by a discrete-time linear time-invariant (LTI) system:

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

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 the system matrices, and $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$ denote the state, input, and output, respectively. This formulation is applicable to soft robotic systems whose locally linearized LTI models are controllable and observable.

The core idea of DeePC is to bypass explicit system identification by directly utilizing offline input–output data rather than a parametric model [18]. This approach is grounded in behavioral systems theory, which represents the system’s dynamics using a block Hankel matrix [26]. For the specific case of our soft robotic arm, we begin by conducting an offline data collection experiment. A predefined sequence of control inputs $u^d = [u^d(0)^\top, u^d(1)^\top, \dots, u^d(T-1)^\top]^\top \in \mathbb{R}^{mT}$, which denotes the incremental lengths of the three actuation cables, is applied to the robotic arm. Simultaneously, the resulting trajectory of the end-effector is recorded as the output sequence $y^d = [y^d(0)^\top, y^d(1)^\top, \dots, y^d(T-1)^\top]^\top \in \mathbb{R}^{pT}$. This collected input–output dataset is then structured into a block Hankel matrix of depth $L \in \mathbb{Z}$. This data matrix serves as a non-parametric model of the robot’s dynamics and is defined as:

$$\begin{bmatrix} \mathcal{H}_L(u^d) \\ \mathcal{H}_L(y^d) \end{bmatrix} = \begin{bmatrix} u^d(0) & u^d(1) & \cdots & u^d(T-L) \\ \vdots & \vdots & \ddots & \vdots \\ u^d(L-1) & u^d(L) & \cdots & u^d(T-1) \\ y^d(0) & y^d(1) & \cdots & y^d(T-L) \\ \vdots & \vdots & \ddots & \vdots \\ y^d(L-1) & y^d(L) & \cdots & y^d(T-1) \end{bmatrix}. \quad (2)$$

Lemma 1 (Fundamental Lemma [26]): Consider the controllable LTI system in (1), and suppose that the input sequence u^d is persistently exciting of order $n+L$. Then, any length- L input–output trajectory $(u_{[0,L-1]}, y_{[0,L-1]})$ of (1) can be represented if and only if there exists a real vector $g \in \mathbb{R}^{T-L+1}$ such that

$$\begin{bmatrix} u_{[0,L-1]} \\ y_{[0,L-1]} \end{bmatrix} = \begin{bmatrix} \mathcal{H}_L(u^d) \\ \mathcal{H}_L(y^d) \end{bmatrix} g. \quad (3)$$

A prerequisite for invoking the lemma is that the input sequence used for offline data collection must be persistently exciting (PE). An input u^d is said to be PE of order L if its corresponding Hankel matrix $\mathcal{H}_L(u^d)$ has full row rank.

To satisfy this condition in practice, a rich variety of input signals were applied to the soft robotic arm during the offline phase, ensuring that the arm explored a comprehensive portion of its workspace and that the resulting data matrix fully captured its dynamics.

According to the lemma, for any controllable soft robotic arm system driven by a PE input u^d , any feasible motion trajectory of length L can be accurately represented, and any future motion that the robot can achieve can be represented as a linear combination of the behaviors it has exhibited in the past. Building on this principle, DeePC partitions the Hankel matrices into past and future components, which are then used to formulate a constrained optimal control problem. Specifically, let T_{ini} , $N \in \mathbb{Z}$, and $L = T_{\text{ini}} + N$. The following block matrices are defined:

$$\begin{bmatrix} U_p \\ U_f \end{bmatrix} := \mathcal{H}_L(u^d), \quad \begin{bmatrix} Y_p \\ Y_f \end{bmatrix} := \mathcal{H}_L(y^d), \quad (4)$$

where (U_p, Y_p) represent the past control inputs and corresponding motion trajectories of the soft robotic arm, comprising the first T_{ini} block rows of $\mathcal{H}_L(u^d)$ and $\mathcal{H}_L(y^d)$, respectively. These data are used to connect the model’s predictions to the most recently observed behavior of the robotic arm. (U_f, Y_f) denote the future control inputs and their associated motion trajectories of the soft arm, encompassing the last N block rows of $\mathcal{H}_L(u^d)$ and $\mathcal{H}_L(y^d)$, thereby forming a predictive basis for all feasible system evolutions. This partitioning provides the basis for casting DeePC as a finite-horizon, data-driven optimal control problem without requiring explicit system identification.

To achieve precise trajectory tracking of the 3D soft robotic arm, the control task is formulated as a receding-horizon optimization problem within the DeePC framework. With an initial window length T_{ini} and prediction horizon N , the resulting optimization problem can be formulated as:

$$\begin{aligned} \min_{g, u, y} \quad & \|y - y_r\|_Q^2 + \|u\|_R^2 \\ \text{subject to} \quad & \begin{bmatrix} U_p \\ U_f \\ Y_p \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ u \\ y_{\text{ini}} \\ y \end{bmatrix}, \\ & u \in \mathcal{U}, y \in \mathcal{Y}, \end{aligned} \quad (5)$$

where y_r is the reference trajectory, Q, R are weighting matrices, $u_{\text{ini}} = [u(t-T_{\text{ini}})^\top, u(t-T_{\text{ini}}+1)^\top, \dots, u(t-1)^\top]^\top$ and $y_{\text{ini}} = [y(t-T_{\text{ini}})^\top, y(t-T_{\text{ini}}+1)^\top, \dots, y(t-1)^\top]^\top$ are the input and output sequences within a past time horizon of length T_{ini} , $u = [u(t)^\top, u(t+1)^\top, \dots, u(t+N-1)^\top]^\top$ and $y = [y(t)^\top, y(t+1)^\top, \dots, y(t+N-1)^\top]^\top$ are the input and output sequences within a prediction horizon of length N , and \mathcal{U}, \mathcal{Y} represent the admissible sets of inputs and outputs.

B. Robust DeePC via Regularization

Although DeePC and MPC are equivalent under ideal conditions (i.e., a controllable LTI system with a persistently exciting input) [18], this equivalence rarely holds for the real-world soft robotic arm. The ideal assumptions are challenged

by multiple factors: (i) measurement noise from sensors and external disturbances (e.g., airflow, vibrations), and (ii) the highly nonlinear dynamics inherent to soft materials (e.g., viscoelasticity, hysteresis), which cannot be fully captured by a finite dataset. These imperfections mean that a newly observed trajectory $(u_{\text{ini}}, y_{\text{ini}})$ may not be perfectly linearly represented in the historical database (U_p, Y_p) . This mismatch makes the equality constraint in the optimization problem impossible to satisfy, leading to infeasibility. To restore feasibility and improve robustness against these real-world effects, the standard DeePC formulation is often augmented with regularization. Common approaches include:

Slack Variables: Introduce an output slack variable σ_y to relax the requirement of perfectly matching past trajectories. It allows for a small deviation σ_y between the soft robotic arm's actual historical trajectory and the best trajectory represented by the data. The deviation is penalized in the cost function, and assigning a sufficiently large weight λ_y ensures that σ_y becomes non-zero only when the original constraints are infeasible.

Regularization on g : Add a penalty term, such as $\lambda_g \|g\|_2^2$, to the cost function. This discourages overfitting to noisy data and improves the generalization and stability of the solution. Physically, this prevents the controller from reacting aggressively to sensor noise, resulting in smoother control actions and reduced jitter in the soft robotic arm's end-effector motion.

The resulting regularized DeePC problem for the soft robotic arm can thus be formulated as:

$$\begin{aligned} \min_{g, u, y, \sigma_y} \quad & \|y - y_r\|_Q^2 + \|u\|_R^2 + \lambda_y \|\sigma_y\|_2^2 + \lambda_g \|g\|_2^2 \\ \text{subject to} \quad & \begin{bmatrix} U_p \\ U_f \\ Y_p \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ u \\ y_{\text{ini}} \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sigma_y \\ 0 \end{bmatrix}, \\ & u \in \mathcal{U}, y \in \mathcal{Y}. \end{aligned} \quad (6)$$

Dimension Reduction: The complexity of soft robots necessitates extensive offline data acquisition to capture their dynamics, resulting in very large Hankel matrices for the DeePC controller. Consequently, the dimension of the optimization variable g becomes extremely large, making the optimization problem computationally intractable for real-time control loops that require millisecond-level response times. To make DeePC practical for real-time implementation, an effective data-driven dimensionality reduction technique is essential. SVD can be applied to reduce the dimension of the Hankel matrix in (6), and thus improve the computational efficiency of the DeePC problem [27].

To implement this, the concatenated Hankel matrix of the collected input–output data is factorized using SVD. Specifically, the decomposition is given by:

$$\begin{bmatrix} \mathcal{H}_L(u^{\text{d}}) \\ \mathcal{H}_L(y^{\text{d}}) \end{bmatrix} = \underbrace{\begin{bmatrix} W_1 & W_2 \end{bmatrix}}_W \underbrace{\begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}}_\Sigma \underbrace{\begin{bmatrix} V_1 & V_2 \end{bmatrix}}_{V^T}. \quad (7)$$

In (7), $W \in \mathbb{R}^{q_1 \times q_1}$ with $q_1 = (m+p)L$ and $V \in \mathbb{R}^{q_2 \times q_2}$ with

$q_2 = T - L + 1$ are orthogonal matrices. $\Sigma \in \mathbb{R}^{q_1 \times q_2}$ is a rectangular diagonal matrix with the singular values arranged in descending order. $\Sigma_1 \in \mathbb{R}^{r \times r}$ contains the top r non-zero singular values, and $W_1, W_2, \Sigma_2, V_1,$ and V_2 are defined with compatible dimension.

By truncating this decomposition to the top r singular values, which capture the system's principal dynamics, a new condensed data matrix $\tilde{\mathcal{H}}_L \in \mathbb{R}^{q_1 \times r}$ is obtained:

$$\tilde{\mathcal{H}}_L = \mathcal{H}_L V_1 = W_1 \Sigma_1. \quad (8)$$

Consequently, the original DeePC optimization problem can be reformulated using the compressed data matrix and a reduced optimization variable $\tilde{g} \in \mathbb{R}^r$. The resulting regularized SVD-DeePC problem for the soft arm is given by:

$$\begin{aligned} \min_{\tilde{g}, u, y, \sigma_y} \quad & \|y - y_r\|_Q^2 + \|u\|_R^2 + \lambda_y \|\sigma_y\|_2^2 + \lambda_g \|\tilde{g}\|_2^2 \\ \text{subject to} \quad & \tilde{\mathcal{H}}_L \tilde{g} = \begin{bmatrix} u_{\text{ini}} \\ u \\ y_{\text{ini}} \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sigma_y \\ 0 \end{bmatrix}, \quad u \in \mathcal{U}, y \in \mathcal{Y}. \end{aligned} \quad (9)$$

This formulation provides a significant computational advantage: the dimension of the optimization variable is reduced from the dataset length $(T - L + 1)$ to the approximation rank r . Such a reduction makes the approach more suitable for real-time control of the soft robotic arm [24]. The choice of r can be guided by inspecting the decay of the singular values or determined based on control performance.

IV. EXPERIMENTS

This section presents an experimental study aimed at evaluating the performance of DeePC on a soft robotic arm in two representative 3D tasks: (i) fixed-point reaching and (ii) trajectory tracking.

A. Setup and data collection

We validated the proposed method on a self-developed cable-driven soft robotic arm. The experimental setup is shown in Fig. 3. The arm is actuated by three cables, with a total effective length of 90 mm. End-effector positions were tracked in real time using a 12-camera OptiTrack FLEX13 motion capture system. Data collection and control were executed on a host PC, with the DeePC framework implemented in MATLAB. Optimal control commands were transmitted via serial communication to a low-level controller, which drove the stepper motors to adjust cable lengths.

To construct a nonparametric representation, input–output data were collected. Here, u^{d} denotes the incremental length of the three actuation cables, while $y^{\text{d}} = (p_x, p_y, p_z)$ represents the end-effector position in task space. During data acquisition, ramp-and-hold excitations were applied to the low-level controller, and the corresponding end-effector positions were continuously recorded. The resulting dataset was then arranged into a Hankel matrix as described in Section III. When sufficiently rich, this data supports an accurate prediction of system behavior.

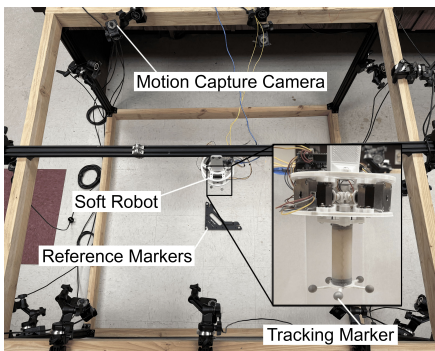


Fig. 3: Experimental setup with motion capture cameras and a test frame.

B. Fixed-point reaching control

To evaluate DeePC’s ability to accurately drive the soft arm to specified target positions, we conducted a multi-stage reference tracking experiment. The bending angle ϕ_b and orientation angle γ_g were commanded to follow stepwise reference trajectories across three stages: (I) $\phi_b = 20^\circ$, $\gamma_g = 0^\circ$; (II) $\phi_b = 40^\circ$, $\gamma_g = 60^\circ$; (III) $\phi_b = 60^\circ$, $\gamma_g = 120^\circ$. Here, ϕ_b and γ_g are computed from the tip position using the constant curvature model [28].

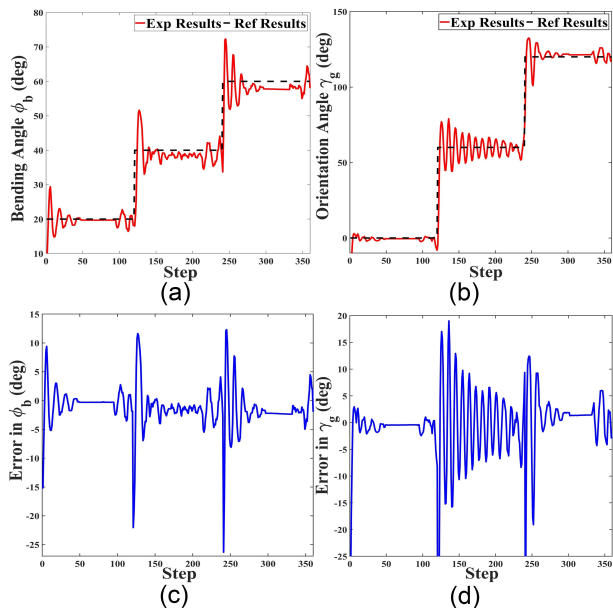


Fig. 4: Experimental results of DeePC control. (a–b) Tracking performance of bending angle ϕ_b and orientation angle γ_g . (c–d) Corresponding tracking errors.

As shown in Fig. 4, DeePC successfully drove the arm to track both the bending angle ϕ_b and orientation angle γ_g . The bending angle converged rapidly in each stage with small steady-state error, exhibiting only limited overshoot and short oscillations before stabilization. For the orientation angle, noticeable oscillations were observed around 60° , but their amplitude decayed gradually and converged to the reference. In contrast, the 0° and 120° stages displayed relatively mild transients and fast settling. Overall, the results demonstrate DeePC’s effectiveness in achieving accurate target reaching

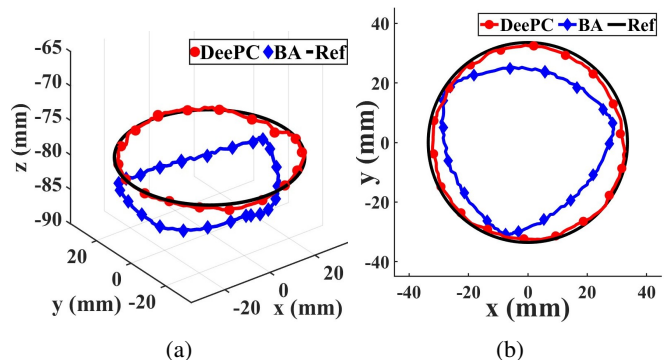


Fig. 5: Trajectory tracking control results. (a) Comparison of 3D tracking performance. (b) Comparison of 2D tracking performance in the x - y plane.

and stage switching without requiring an explicit model. Further improvements may be obtained by tuning the weighting and regularization parameters to suppress residual overshoot and oscillations.

C. Trajectory tracking control

In this experiment, the arm was tasked to track a circular trajectory in task space. The reference trajectory was defined as a circle of radius r , discretized into a sequence of waypoints. For the baseline, control inputs were computed directly from the geometric model. For DeePC, the control inputs were constrained to $u \in [0, 90]$, with $T_{\text{ini}} = 20$ and $N = 30$. The weighting matrices were set to $Q = 10 \cdot I_{3 \times 3}$ and $R = 2 \times 10^{-3} \cdot I_{3 \times 3}$, while the regularization parameters were chosen as $\lambda_g = 300$ and $\lambda_y = 1,000$.

For comparison, we implemented a baseline model-based controller derived from the constant-curvature assumption [25]. In this framework, the relationship between the backbone curvature of the arm and the cable length is expressed through idealized geometric relations. Specifically, let $\kappa_{c,i}$ and $R_{c,i}$ denote the curvature and radius of curvature of the i -th cable, respectively, and let l_i denote its length. For the i -th cable in a section:

$$\frac{1}{\kappa_b} = \frac{1}{\kappa_{c,i}} + d_i \quad (10)$$

$$l_i = R_{c,i} \cdot \phi_b = \frac{\kappa_b}{\kappa_{c,i}} L \quad (11)$$

where d_i represents the distance between the incident point of the i -th cable and the neutral plane, κ_b is the backbone curvature, and L is the segment length.

As shown in Fig. 5, DeePC achieved precise tracking with minimal deviation from the reference trajectory, both in 3D space (Fig. 5a) and in the x - y projection (Fig. 5b). The DeePC trajectory closely overlapped the reference circle, maintaining a consistent radius and smooth curvature. In contrast, the baseline approach produced a distorted path: in 3D, the baseline trajectory sagged below the desired plane, while in the x - y projection the circle was compressed into an irregular, triangular-like shape. These distortions reflect the limitations of constant-curvature geometric assumptions, which cannot capture the soft arm’s nonlinear bending and

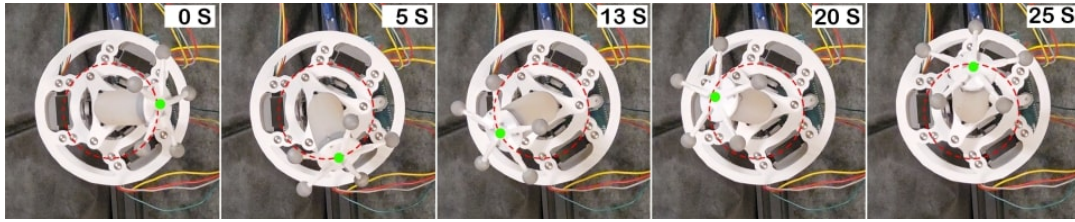


Fig. 6: Movement and bending configurations of the soft arm during circular trajectory tracking.

torsional behaviors. The physical bending configurations of the soft arm during circular trajectory tracking are illustrated in Fig. 6. The sequential snapshots show that DeePC successfully generated smooth, coordinated cable motions, enabling the arm to trace the desired circular path in real time.

V. CONCLUSION

This paper addressed the challenge of high-performance, three-dimensional dynamic control for soft robots. We designed, implemented, and experimentally validated a data-enabled predictive control (DeePC) framework on an optimized cable-driven soft arm platform. The experiments confirmed the effectiveness of DeePC for precise 3D soft robot control, advancing beyond prior work limited to planar systems or open-loop control, and demonstrated its ability to handle the complex, nonlinear dynamics of soft robots without requiring explicit first-principles models. Future work will focus on evaluating performance under large external disturbances and varying payloads. Another important direction is extending this framework from single-segment to multi-segment soft robotic arms, enabling more dexterous and versatile manipulation in 3D space. Furthermore, applying the method to advanced tasks, such as fruit handling and object manipulation, represents a promising step toward expanding the practical capabilities of soft robotic systems.

REFERENCES

- [1] X. Wang, H. Kang, H. Zhou, W. Au, M. Y. Wang, and C. Chen, "Development and evaluation of a robust soft robotic gripper for apple harvesting," *Computers and Electronics in Agriculture*, vol. 204, p. 107552, 2023.
- [2] M. Cianchetti, C. Laschi, A. Menciassi, and P. Dario, "Biomedical applications of soft robotics," *Nature Reviews Materials*, vol. 3, no. 6, pp. 143–153, 2018.
- [3] C. Lee, M. Kim, Y. J. Kim, N. Hong, S. Ryu, H. J. Kim, and S. Kim, "Soft robot review," *International Journal of Control, Automation and Systems*, vol. 15, no. 1, pp. 3–15, 2017.
- [4] G. Bao, H. Fang, L. Chen, Y. Wan, F. Xu, Q. Yang, and L. Zhang, "Soft robotics: Academic insights and perspectives through bibliometric analysis," *Soft robotics*, vol. 5, no. 3, pp. 229–241, 2018.
- [5] A. Das and M. Nabi, "A review on soft robotics: Modeling, control and applications in human-robot interaction," in *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE, 2019, pp. 306–311.
- [6] Z. Chen, D. Wu, Q. Guan, D. Hardman, F. Renda, J. Hughes, T. G. Thuruthel, C. Della Santina, B. Mazzolai, H. Zhao *et al.*, "A survey on soft robot adaptability: Implementations, applications, and prospects [survey]," *IEEE Robotics & Automation Magazine*, 2025.
- [7] Z. Chen and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
- [8] J. Wang and A. Chortos, "Control strategies for soft robot systems," *Advanced Intelligent Systems*, vol. 4, no. 5, p. 2100165, 2022.
- [9] C. Della Santina, C. Duriez, and D. Rus, "Model-based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Systems Magazine*, vol. 43, no. 3, pp. 30–65, 2023.
- [10] T. Zheng and J. Burgner-Kahrs, "Estimating dynamic soft continuum robot states from boundaries," *arXiv preprint arXiv:2505.04491*, 2025.
- [11] R. K. Katzschmann, C. Della Santina, Y. Tshimitsu, 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 *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019, pp. 454–461.
- [12] M. Roshanfar, S. Taki, A. Sayadi, R. Cecere, J. Dargahi, and A. Hooshiar, "Hyperelastic modeling and validation of hybrid-actuated soft robot with pressure-stiffening," *Micromachines*, vol. 14, no. 5, p. 900, 2023.
- [13] J. Chenevier, D. González, J. V. Aguado, F. Chinesta, and E. Cueto, "Reduced-order modeling of soft robots," *PloS one*, vol. 13, no. 2, p. e0192052, 2018.
- [14] T. Navez, E. Ménager, P. Chaillou, O. Goury, A. Kruszewski, and C. Duriez, "Modeling, embedded control and design of soft robots using a learned condensed fem model," *IEEE Transactions on Robotics*, 2025.
- [15] X. Qi, D. Chen, Z. Li, and X. Tan, "Back-stepping experience replay with application to model-free reinforcement learning for a soft snake robot," *IEEE Robotics and Automation Letters*, 2024.
- [16] S. Garg, M. Goharimanesh, S. Sajjadi, and F. Janabi-Sharifi, "Autonomous control of soft robots using safe reinforcement learning and covariance matrix adaptation," *Engineering Applications of Artificial Intelligence*, vol. 153, p. 110791, 2025.
- [17] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Data-driven control of soft robots using koopman operator theory," *IEEE transactions on robotics*, vol. 37, no. 3, pp. 948–961, 2020.
- [18] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the deep," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 307–312.
- [19] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Decentralized data-enabled predictive control for power system oscillation damping," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 3, pp. 1065–1077, 2021.
- [20] V. Chinde, Y. Lin, and M. J. Ellis, "Data-enabled predictive control for building hvac systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 8, 2022.
- [21] E. Elokda, J. Coulson, P. N. Beuchat, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for quadcopters," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 8916–8936, 2021.
- [22] 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.
- [23] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, "Robust data-enabled predictive control: Tractable formulations and performance guarantees," *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 3163–3170, 2023.
- [24] H. Wang, K. Zhang, K. Lee, Y. Mei, K. Zhu, V. Srivastava, J. Sheng, and Z. Li, "Mechanical design and data-enabled predictive control of a planar soft robot," *IEEE Robotics and Automation Letters*, 2024.
- [25] X. Qi, Y. Mei, D. Chen, Z. Li, and X. Tan, "Design and nonlinear modeling of a modular cable-driven soft robotic arm," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 4, pp. 3083–3091, 2024.
- [26] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.
- [27] K. Zhang, Y. Zheng, C. Shang, and Z. Li, "Dimension reduction for efficient data-enabled predictive control," *IEEE Control Systems Letters*, vol. 7, pp. 3277–3282, 2023.
- [28] B. Hasanshahi, L. Cao, K.-Y. Song, and W. Zhang, "Design of soft robots: A review of methods and future opportunities for research," *Machines*, vol. 12, no. 8, p. 527, 2024.