TU Delft

Delft University of Technology

Document Version
Final published version

Licence
Dutch Copyright Act (Article 25fa)

Citation (APA)
Liu, J., Shahabishalghouni, E., & Santina, C. D. (2026). Neural Linear Oscillator Networks and their Application to Soft Robots Model Learning and Control. *IEEE Robotics and Automation Letters*, *11*(3), 3326-3333. https://doi.org/10.1109/LRA.2026.3656721

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Neural Linear Oscillator Networks and Their Application to Soft Robots Model Learning and Control

Jingyue Liu, *Graduate Student Member, IEEE*, Ebrahim Shahabishalghouni, and Cosimo Della Santina

***Abstract*—Recent advances in machine learning have begun to embed oscillatory network principles within neural architectures, aiming to enhance computational efficiency and robustness in time-series regression. Building on these developments, we take a step toward applying such principles to the learning of physical dynamics. We introduce Neural Linear Oscillator Networks (nLON): a vision-based framework that extracts compact latent representations of complex motion directly from image sequences. A convolutional autoencoder encodes position and velocity into a low-dimensional manifold, whose temporal evolution is governed by coupled linear mechanical oscillators driven by a linear combination of the inputs. This strong structural prior not only promotes sample efficiency and interpretability but also guarantees that the learned model remains a mechanical, Wiener-type system. From this formulation, we derive closed-loop controllers that ensure stable regulation. We focus on soft robots—systems whose nonlinear, continuous, and high-dimensional dynamics make them both a challenging and ideal testbed for our approach. Using tentacle robots in high-fidelity simulations and real-world experiments, we validate that our framework delivers accurate long-horizon predictions and consistently surpasses state-of-the-art baselines, achieving superior structural fidelity and final-step accuracy. Finally, we leverage the learned dynamics for model-based control, demonstrating in simulation that the resulting scheme achieves robust and reliable tracking.**

***Index Terms*—Modeling, control, and learning for soft robots, soft sensors and actuators, representation learning.**

I. Introduction

MODELING nonlinear dynamical systems from high-dimensional sensory data remains a central challenge in robotics and the physical sciences [1]. Soft robotic systems exemplify this difficulty: their compliance and continuum structure yield an effectively infinite-dimensional configuration space, and their nonlinear dynamics resist description through traditional rigid-body models [2]. These challenges have driven growing interest in data-driven modeling of soft robots [3], [4], particularly vision-based methods [5], [6], which offer dense, global measurements rich enough to capture such complex behavior.

In the broader context of learning dynamics and world models in machine learning, latent-variable models—and in particular autoencoder-based approaches have emerged as powerful tools [7] to connect high-dimensional observations with the underlying low-dimensional structure of physical behavior. These models compress high-dimensional sensory inputs (e.g., RGB sequences) into compact latent representations that retain the essential system state. When structured to respect physical principles, such latent states can support system identification, prediction, and model-based control [7], [8], [9]. Even with effective latent encodings, however, modeling the temporal evolution of complex dynamics remains a significant hurdle—limiting the application of these techniques to systems such as soft robots. Indeed, almost no validation of these advanced techniques in real-world soft robots exists. Here, we explore the possibility of using a new class of recurrent neural architectures in this context: *neural oscillator networks*.

Harmonic oscillators have recently gained traction as replacements for recurrent neurons, improving robustness and generalization while significantly reducing network size [10], [11], [12]. We contend that, when applied to learning dynamics, such architectures can combine the generic modeling power of oscillators with the benefits of physical priors—provided they are structured to inherit the fact that oscillators are, by nature, mechanical systems. A substantial body of work has shown that embedding physical priors into neural networks [13] can significantly enhance learning efficiency and model fidelity. We therefore anticipate that introducing these priors explicitly, via mechanical oscillator structures, will yield similar benefits.

Early efforts to learn dynamics with neural oscillators have not fully exploited this potential, though they have shown promising results. Kapoor et al. [14] used neural oscillators to predict solutions of parametric PDEs; Chandra et al. [15] designed architectures for modeling hysteresis; and Ceni et al. [16] applied them to chaotic dynamics within a reservoir computing framework. However, these works primarily address autonomous systems. Here, "autonomous" means evolution without commanded actuation; these models therefore do not capture how control inputs drive the system. Consequently, prior oscillator-based models in their work do not tackle the input-driven dynamics required in controlled physical systems. To our knowledge, the only study explicitly leveraging neural oscillators as physical priors is [17], which demonstrated learning of dynamics from pixels, including a simple soft-robot example. Yet this approach

Received 12 September 2025; accepted 7 January 2026. Date of publication 21 January 2026; date of current version 2 February 2026. This article was recommended for publication by Associate Editor F. Renda and Editor B. Mazzolai upon evaluation of the reviewers' comments. This work was supported by the European Union's Horizon Europe Program, Project EMERGE, under Grant Agreement 101070918. *(Corresponding author: Jingyue Liu.)*

Jingyue Liu and Ebrahim Shahabishalghouni are with the Department of Cognitive Robotics, Delft University of Technology, 2628 Delft, The Netherlands (e-mail: j.liu-14@tudelft.nl; e.shahabishalghouni@tudelft.nl).

Cosimo Della Santina is with the , Department of Cognitive Robotics, Delft University of Technology, 2628 Delft, The Netherlands, and also with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Oberpfaffenhofen, Germany (e-mail: c.dellasantina@tudelft.nl).

Digital Object Identifier 10.1109/LRA.2026.3656721

2377-3766 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: TU Delft Library. Downloaded on February 04,2026 at 10:43:57 UTC from IEEE Xplore. Restrictions apply.

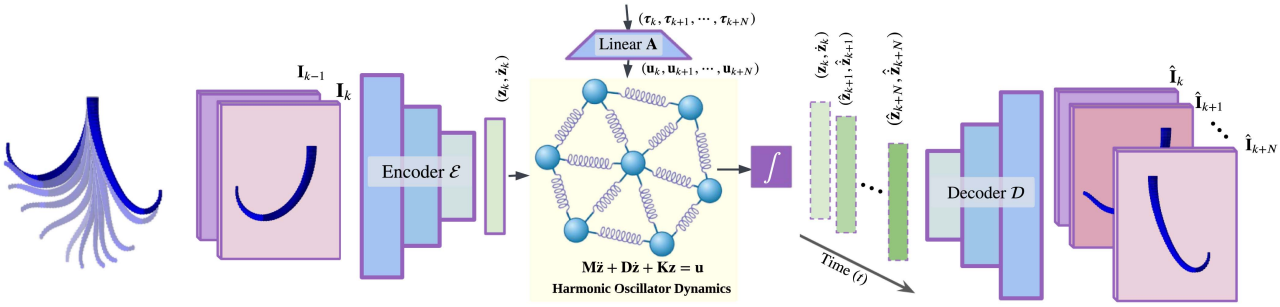


Fig. 1. Overview of the proposed modeling framework. Sequential images $(\mathbf{I}_{k-1}, \mathbf{I}_k)$ are processed by an encoder to extract latent space representations $(\mathbf{z}_k, \dot{\mathbf{z}}_k)$ representing position and velocity in a low-dimensional space $\mathbf{z} \in \mathbb{R}^l$. Latent dynamics, modeled as a harmonic oscillator $\mathbf{M}\ddot{\mathbf{z}} + \mathbf{D}\dot{\mathbf{z}} + \mathbf{K}\mathbf{z} = \mathbf{u}$, where a linear layer maps inputs $(\boldsymbol{\tau}_k, \boldsymbol{\tau}_{k+1}, \ldots, \boldsymbol{\tau}_{k+N})$ to control inputs $(\mathbf{u}_k, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_{k+N})$. The dynamics are integrated over time to predict future states $\{(\hat{\mathbf{z}}_{k+1}, \hat{\dot{\mathbf{z}}}_{k+1}), \cdots (\hat{\mathbf{z}}_{k+N}, \hat{\dot{\mathbf{z}}}_{k+N})\}$, which are decoded back into images $\{\hat{\mathbf{I}}_k, \hat{\mathbf{I}}_{k+1}, \ldots, \hat{\mathbf{I}}_{k+N}\}$ to complete the sequence reconstruction and dynamics prediction.

constrains oscillator interconnections to a simple topology, increasing expressiveness through nonlinearities at the cost of reduced physical plausibility. The resulting models capture relatively simple dynamics, trained entirely in simulation and from low-resolution images.

In this work, we introduce *Neural Linear Oscillator Networks* (nLON): a neural architecture in which latent dynamics are explicitly constrained to behave as a *mechanical* interconnection of linear oscillators. Expressivity arises from allowing arbitrary interconnection topologies rather than nonlinearities, ensuring that for any choice of parameters the model retains the mathematical structure of a Wiener-type [18] dynamical system, as this structure preserves certain interpretability, enables well-posed linear control design in latent space, and separates observation nonlinearities from the underlying dynamics model. While applicable to a broad range of mechanical systems, we focus here on learning the dynamics of soft robots directly from visual data. Furthermore, we show that nLON models can be combined with first-principles control methods to regulate shape and close the loop in pixel space—without explicit trajectory optimization or additional learning.

Our key contributions are:

- *Physics-aligned latent prior:* We introduce *Neural Linear Oscillator Networks (nLON)*, a coupled *linear* mechanical–oscillator latent model (Wiener-type) with arbitrary interconnection topology, capturing the oscillatory structure of soft-robot motion with interpretable states.
- *End-to-end vision learning:* A convolutional autoencoder maps images to a compact position–velocity latent (3–4 dims), whose dynamics follow a control-driven linear-oscillator ODE, enabling sample-efficient, stable rollouts directly from pixels; nLON achieves better long-term prediction than ConvLSTM and Koopman.
- *Latent-space control:* Two LQR-based control designs are presented that can regulate motion on the learned manifold, one of which directly closes the loop in pixel space, demonstrating robust performance across two actuation types in simulated soft robots.

II. Neural Linear Oscillator Networks for Learning Complex Dynamics

Fig. 1 provides an overview of the proposed nLON architecture. A neural encoder–decoder maps two sequential images to a compact latent representation. These latent variables form a low-dimensional abstraction of the system state, interpreted as the positions and velocities of masses in a network of linear mechanical oscillators interconnected by springs and dampers. Since the encoder and decoder are static nonlinearities and the latent dynamics are linear, the model naturally realizes a Wiener-type dynamical system. The oscillator network governs the temporal evolution of the latent states, while control inputs are applied to all masses via a linear encoder. The entire architecture is trained end-to-end, jointly learning the neural weights and the physical parameters of the oscillator network. The following sections detail each component of this framework.

A. Convolutional Autoencoder Design

We employ a convolutional autoencoder (CAE) [19] to encode sequential grayscale image data ($\mathbf{I} \in \mathbb{R}^{\text{width}\times\text{height}}$) into a lower-dimensional and compact latent space representation $\mathbf{z}$ and reconstruct them. The autoencoder consists of the encoder $\mathcal{E}(\cdot)$ in Fig. 2(a), which extracts compact features, and the decoder $\mathcal{D}(\cdot)$ in Fig. 2(b), which reconstructs the input images.

The encoding process can be described mathematically as

$$(\mathbf{z}_k, \dot{\mathbf{z}}_k) = \mathcal{E}(\mathbf{I}_{k-1}, \mathbf{I}_k; \boldsymbol{\theta}_\mathcal{E}) \tag{1}$$

where $\mathcal{E}$ represents the encoder parameterized by $\boldsymbol{\theta}_\mathcal{E}$, and $\mathbf{I}_{k-1}$ and $\mathbf{I}_k$ are two consecutive input images with a fixed interval $\mathrm{d}t$. The encoder outputs the latent vector $(\mathbf{z}_k, \dot{\mathbf{z}}_k)$, where $\mathbf{z}_k$ represents the position inferred from $\mathbf{I}_k$ and $\dot{\mathbf{z}}_k$ represents the velocity, derived from the difference between $\mathbf{I}_{k-1}$ and $\mathbf{I}_k$.

Conversely, the decoder can reconstruct the input image $\mathbf{I}_k$ from the latent representation by

$$\hat{\mathbf{I}}_k = \mathcal{D}(\mathbf{z}_k; \boldsymbol{\theta}_\mathcal{D}), \tag{2}$$

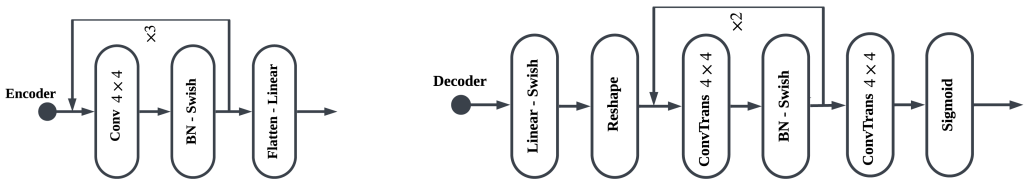
where $\mathcal{D}$ is parameterized by $\boldsymbol{\theta}_\mathcal{D}$, and $\hat{\mathbf{I}}_k$ is the reconstructed image corresponding to the input $\mathbf{I}_k$.

To enhance the training efficiency and overall performance, the CAE integrates the Swish activation function [20] with Batch Normalization (BN). Swish has demonstrated superior effectiveness in deep learning tasks, particularly when combined with BN [21].

B. Latent Dynamics Model Learning

We encode two successive frames into a low-dimensional latent state $(\mathbf{z}, \dot{\mathbf{z}}) \in \mathbb{R}^l \times \mathbb{R}^l$, with $l \ll (\text{width} \times \text{height})$. We interpret $\mathbf{z}$ as generalized coordinates of a coupled mechanical

Authorized licensed use limited to: TU Delft Library. Downloaded on February 04,2026 at 10:43:57 UTC from IEEE Xplore. Restrictions apply.



(a) Encoder Structure (b) Decoder Structure

Fig. 2. Schematic illustration of the encoder-decoder architecture used for transforming sequential images into compact latent representations and reconstructing them back into images.

system and model its evolution as a controlled coupled harmonic oscillators:

$$\mathbf{M}\,\ddot{\mathbf{z}} + \mathbf{D}\,\dot{\mathbf{z}} + \mathbf{K}\,\mathbf{z} = \mathbf{A}\,\boldsymbol{\tau}, \tag{3}$$

where $\mathbf{M} \succ 0$, $\mathbf{D} \succeq 0$, $\mathbf{K} \succ 0$, and $\mathbf{A} \in \mathbb{R}^{l\times m}$ maps actuator inputs $\boldsymbol{\tau} \in \mathbb{R}^m$ to latent generalized forces. To guarantee physical structure while keeping training stable, we parameterize:

- $\mathbf{M} = \mathrm{diag}(m_i)$ with $m_i = w_{i,m}^2 + \varepsilon$, uses a small constant $\varepsilon > 0$ as a positive-definiteness regularizer to prevent singularity when $w_{i,m} \simeq 0$ during training,
- $\mathbf{D} = \mathrm{diag}(d_i)$ with $d_i = \mathrm{Sigmoid}(w_{i,d})$ enforces nonnegative and bounded damping values in (0,1), preventing unstable negative damping and excessively large dissipation,
- $\mathbf{K} = \mathbf{L}\,\mathbf{L}^\top$ with lower-triangular $\mathbf{L}$ and softplus on $\mathrm{diag}(\mathbf{L})$ to ensure $\mathbf{K} \succ 0$ by Cholesky decomposition.

All parameters (encoder/decoder and $\mathbf{M}, \mathbf{D}, \mathbf{K}, \mathbf{A}$) are learned end-to-end. This structure biases the model toward passive, well-conditioned dynamics, enabling direct LQR design in the latent space.

C. Neural ODE Integration

We evolve the latent state with a Neural Ordinary Differential Equation (Neural ODE) solver [22]. Starting from an initial latent state $(\mathbf{z}_i, \dot{\mathbf{z}}_i)$ and a control sequence, we call a differentiable integrator (e.g., `odeint`) [1] to propagate the state to the next time stamp and repeat over the horizon. This supports variable integration steps and allows backpropagation through the solver. In practice, it yields stable long-horizon rollouts and enables conditioning of the predictions on arbitrary input sequences.

D. Loss Design

We train endtoend with a normalized composite loss that balances image fidelity, dynamical consistency, and simple physical priors:

$$\mathcal{L}_{\text{total}} = \mathrm{w}_1\mathcal{L}_{\text{recon}} + \mathrm{w}_2\mathcal{L}_{\text{latent}} + \mathrm{w}_3\mathcal{L}_{\text{vel}} + (\mathrm{w}_4\mathcal{L}_{\text{eq}}) \tag{4}$$

with

$$\begin{aligned} \mathcal{L}_{\text{recon}} &= \textstyle\sum_{i=0}^{2} \mathrm{BCE}(\mathbf{I}_{k+i}, \hat{\mathbf{I}}_{k+i}), \\ \mathcal{L}_{\text{latent}} &= \textstyle\sum_{i=1}^{2} \mathrm{MSE}((\mathbf{z}_{k+i}, \dot{\mathbf{z}}_{k+i}), (\hat{\mathbf{z}}_{k+i}, \hat{\dot{\mathbf{z}}}_{k+i})), \\ \mathcal{L}_{\text{vel}} &= \|\dot{\mathbf{z}}_{ii}\|_2 + \left\|\mathbf{z}_{ii} - \mathbf{z}_{(i-1)i}\right\|_2, \\ \mathcal{L}_{\text{eq}} &= \|\mathbf{z}_e - \hat{\mathbf{z}}_e\|_2, \end{aligned} \tag{5}$$

- $\mathcal{L}_{\text{recon}}$(image fidelity): binary cross-entropy on the current and next two frames decoded from the latent rollout;
- $\mathcal{L}_{\text{latent}}$ (dynamics consistency): mean squared error between the one–twostep integrated latent state and the target latent by re-encoding the real future image pairs $\mathcal{E}(\mathbf{I}_{k+i-1}, \mathbf{I}_{k+i})$;
- $\mathcal{L}_{\text{vel}}$ (temporal stillness prior): when two consecutive frames are identical, an ℓ_2 penalty is applied on the latent velocity to enforce zero motion and promote disentanglement between position and velocity;
- $\mathcal{L}_{\text{eq}}$ (optional equilibrium bias): encourages the predicted latent equilibrium to match the rest-state image $\mathbf{z}_{\text{e}} = \mathcal{E}(\mathbf{I}_{\text{e}}, \mathbf{I}_{\text{e}})$; this term is used only when such an equilibrium image exists.

We choose w_1, w_2, w_3, and w_4 as the hyperparameters that control their relative contributions. The loss weights are chosen according to the relative scale of each component to ensure balanced gradients and stable training, following common practice in multi-objective learning.

E. Baseline Models Setup

We compare nLON to two representative sequence models trained under the same data splits and rollout protocol.

a) ConvLSTM (Torque-conditioned): A compact ConvLSTM [23], [24] operates at a lowresolution bottleneck. A convolutional encoder maps each frame to features, and two context frames initialize the recurrent state. During rollout, a small MLP converts the current torque input into FiLMstyle parameters [25] that modulate the encoder features of the previous prediction before each ConvLSTM update; a decoder then produces the next frame. After the twoframe bootstrap, all frames are generated autoregressively using only the torque sequence and the last prediction, so the information budget matches nLON.

b) Autoencoder + Koopman (control-affine latent dynamics): A convolutional autoencoder encodes a twoframe snippet into a $2l$dimensional latent. The latent state then evolves with a linear, controlaffine update plus a bias term [26], [27]; only the subvector ($\in \mathbb{R}^l$) is decoded back to images, like we do for the position vector in nLON. Training uses the same reconstruction loss on decoded frames, a latent loss that enforces twostep linear prediction in the latent space, and a small stability regularizer on the transition map (e.g., shrink toward identity or bound the spectral radius). For fair comparison, we perform a threeframe rollout.

III. Model-Based Control With a Learned Model

In this section, we illustrate a control architecture that exploits the strong structural priors of nLON—specifically, that the learned model is a Wiener mechanical system for any choice of weights. The overall architecture, shown in Fig. 3, demonstrates

[1] We use an adaptive ODE solver with moderate tolerances and, for ablations, a fixed-step Runge–Kutta scheme. Both are compatible with end-to-end training.

Authorized licensed use limited to: TU Delft Library. Downloaded on February 04,2026 at 10:43:57 UTC from IEEE Xplore. Restrictions apply.

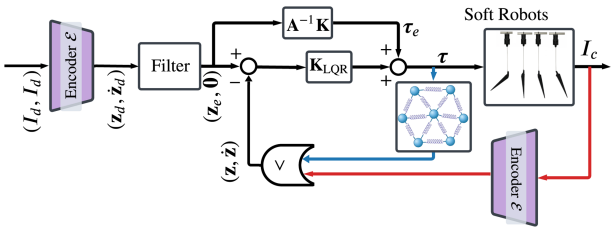


Fig. 3. Latent-space control architecture. The desired image $\mathbf{I}_d$ is encoded into latent trajectories $(\mathbf{z}_d, \dot{\mathbf{z}}_d)$, filtered to form the reference state $(\mathbf{z}_e, \mathbf{0})$. and used by a latent-space LQR controller to generate feedforward torque. Feedback is computed via either the blue path from LHOs, or the red path from re-encoded real observations $\mathbf{I}_c$.

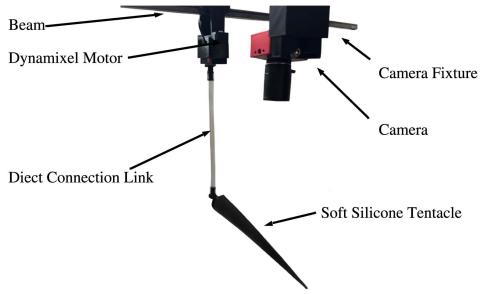


Fig. 4. Experimental setup of the planar soft robotic arm system used for validation purposes in this work. A silicone tentacle is directly connected to a Dynamixel motor through a rigid link.

how these controllers leverage the learned latent dynamics to regulate the behavior of the original system.

A. Equilibria in Latent Space

The latent system is generally underactuated (non–full-rank $\mathbf{A}$), so not every target latent is exactly reachable as an equilibrium. Given a desired latent $\mathbf{z}_{\mathrm{d}}$ (from encoding a target image $\mathbf{I}_{\mathrm{d}}$), we choose a feedforward input $\boldsymbol{\tau}_{\mathrm{e}}$ whose induced equilibrium $\mathbf{z}_{\mathrm{e}}$ is nearest to $\mathbf{z}_{\mathrm{d}}$ as

$$\min_{\boldsymbol{\tau}} |\mathbf{K}^{-1}\mathbf{A}\boldsymbol{\tau}_{\mathrm{e}} - \mathbf{z}_{\mathrm{d}}|_2. \tag{6}$$

We can solve a least-squares problem for $\boldsymbol{\tau}_{\mathrm{e}}$, and for cable actuation, we enforce $\boldsymbol{\tau}_{\mathrm{e}} \geq \mathbf{0}$ via NNLS. We then set $\mathbf{z}_{\mathrm{e}} = \mathbf{K}^{-1}\mathbf{A}$.

B. LQR-Based Latent Space Control

We develop a state-feedback controller based on the Linear Quadratic Regulator (LQR) framework, utilizing the learned harmonic oscillators (LHOs) dynamics to stabilize the system around a desired equilibrium. The full control strategy is summarized in Fig. 3, which combines feedforward control based on a desired image trajectory with feedback control computed from either model-based predictions or real robot observations. The feedforward path computes a nominal control input with the learned matrices $\boldsymbol{\tau}_{\mathrm{e}} = \mathbf{A}^{-1}\mathbf{K}\mathbf{z}_{\mathrm{e}}$.

We consider two control strategies, a predictive feedforward action generated via an internal model (IM) and a direct feedback from pixel space (FB), as illustrated in Fig. 3:

- *Predictive Internal Model (Blue Path):* propagate the latent state with Eq. 3 to estimate $(\mathbf{z}, \dot{\mathbf{z}})$ and close the loop purely in latent space
- *Pixel-level Feedback (Red Path):* re-encode the current camera frame $\mathbf{I}_c$ to refresh $(\mathbf{z}, \dot{\mathbf{z}})$. It introduces real-time correction into the loop to compensate for model mismatch and accumulate sensor feedback.

The latent-space dynamics are modeled as a second-order continuous-time system with the learned matrices $\mathbf{M}$, $\mathbf{D}$, $\mathbf{K}$, and $\mathbf{A}$ as

$$\begin{bmatrix}\dot{\mathbf{z}}\\ \ddot{\mathbf{z}}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D}\end{bmatrix}\begin{bmatrix}\mathbf{z}\\ \dot{\mathbf{z}}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{M}^{-1}\mathbf{A}\end{bmatrix}\boldsymbol{\tau}. \tag{7}$$

We compute a stabilizing LQR gain $\mathbf{K}_{\mathrm{LQR}}$ by solving the continuous-time Algebraic Riccati Equation [28] with suitable designer weights $\mathbf{Q} \succeq 0$ for state and $\mathbf{R} \succ 0$ for input. The resulting control input $\boldsymbol{\tau}$ is designed as a linear feedback law:

$$\boldsymbol{\tau} = -\mathbf{K}_{\mathrm{LQR}}\left(\begin{bmatrix}\mathbf{z} & \dot{\mathbf{z}}\end{bmatrix} - \begin{bmatrix}\mathbf{z}_{\mathrm{e}} & \mathbf{0}\end{bmatrix}\right)^{\top} + \boldsymbol{\tau}_{\mathrm{e}}. \tag{8}$$

IV. Results

We evaluate the proposed framework on a planar soft-tentacle robot, assessing the accuracy of latent-space dynamics prediction and the effectiveness of the learned-model-based controller.

A. Experiment Setup

Our experiments focus on a planar single-segment soft arm.

1) Simulation: Two soft robotic arms were simulated using the Cosserat-rod solver Elastica [29]:

- *Direct Actuation (DA):* a base-mounted motor applies torque;
- *Cable Actuation (CA):* two tendons drive the arm [30].

Both simulations include gravity ($9.81\ \mathrm{m/s^2}$) and external loading. Key parameters are listed in Table I.

2) Experiment Environments: The real setup (Fig. 4) consists of a silicone tentacle (Dragon Skin 30) directly coupled to a Dynamixel motor through a lightweight rigid link. The base frame and camera mount are 3-D printed for rigidity and repeatability. The camera provides grayscale video input for training the latent model.

3) Latent Space Dimension: Although a larger latent dimension l can encode richer features, overly large values may result in overfitting. For the single-segment planar arm considered here, we found that $l = 4$ (DA) and $l = 3$ (CA) work best, in line with intuition from first-principle approximations such as PCC and PCS.

4) Data Generation and Training Protocols: Trajectories were generated with constant, sinusoidal, and cosine actuation profiles of varying amplitude and phase. All simulated images were converted to grayscale, resized, and cropped to 64×64 px. Hardware frames underwent denoising and background removal through thresholding and morphological filtering, followed by cropping (top=25, left=0, height=235, width=340), brightness enhancement ($\times 3.0$), and resizing to 64×92 px. Independent nLON models were trained for each setup (l=4 for DA, l=3 for CA) using the hyperparameters listed in Table. II. All models are trained in a fully supervised manner on frame-pair data with actuation inputs, using end-to-end optimization of the encoder–decoder and latent ODE until the validation loss plateaus on a single RTX 3090 GPU.

B. Latent Dynamics Prediction

1) Simulation: We assess open-loop prediction on unseen trajectories to evaluate how well the learned latent dynamics

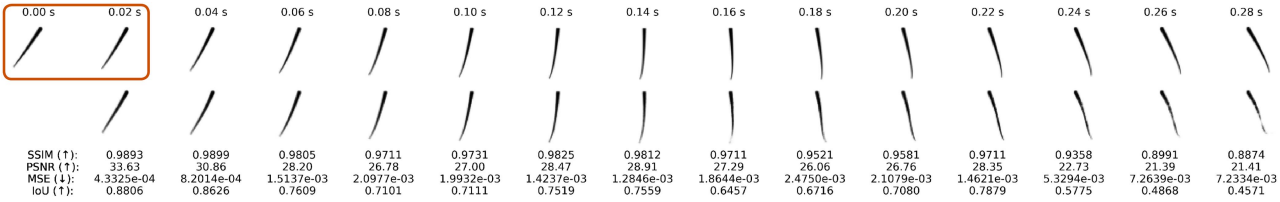
Authorized licensed use limited to: TU Delft Library. Downloaded on February 04,2026 at 10:43:57 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SIMULATION PARAMETERS FOR SOFT ROBOTIC ARM DYNAMICS

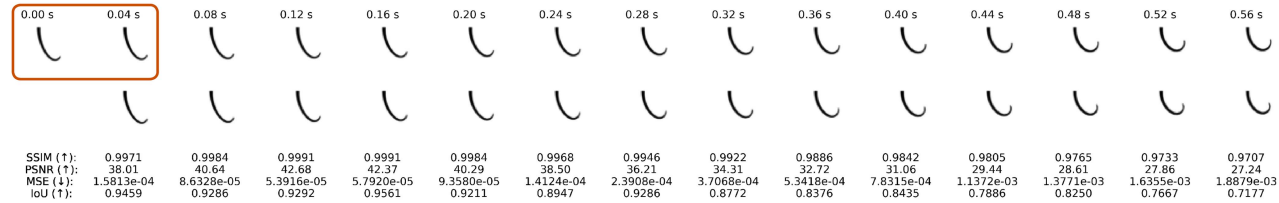
	Length [m]	Radius [cm]	# elem	Density [kg/m^3]	Young's Modulus KPa	Poisson Ratio	Damping Coefficient	Actuation Range	Time Step [s]	Snapshot [s]
DA	0.25	1.1 to 0.2	30	1080.0	940.0	0.5	0.02	-0.025 to 0.025 [Nm]	0.0002	0.01 × 2
CA	0.38	1.6 to 0.48	30	1050.0	90.0	0.36	0.02	0.2 to 12.5 [N]	0.0001	0.01 × 4

TABLE II
DATASET COMPOSITION AND TRAINING HYPERPARAMETERS FOR SIMULATION AND HARDWARE SETUPS

Setup	#Traj. (Test)	Duration (s)	Total Frames (Train/Val)	Split (Train:Val)	Batch	Epochs	learning rate	weight decay
SIM-DA	56 (4)	3.0	12,982 / 1,443	9:1	128	1,500	2×10^{-4}	1×10^{-4}
SIM-CA	73 (3)	2.0–4.0	16,839 / 1,871	9:1	1,024	2,000	5×10^{-4}	6×10^{-5}
HW-DA	25 (2)	10.0	16,839 / 1,871	9:1	512	7,000	3×10^{-3}	5×10^{-5}



(a) DA open-loop prediction performance



(b) CA open-loop prediction performance

Fig. 5. Visualization of the learned DA and CA models' open-loop prediction performances on test samples. The red blocks represent the input frames provided to the network. Rows 1 show the ground truth images, while rows 2 display the model's predicted frames for the corresponding time steps.

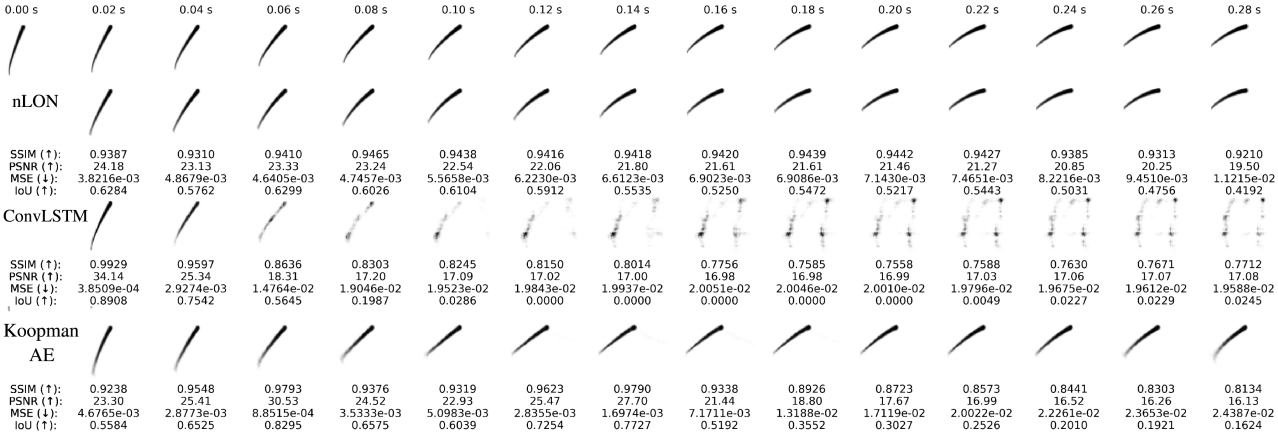


Fig. 6. Comparison of open-loop predictions over 14 time steps for the simulation DA case. The first row shows the ground truth sequence, and the subsequent rows show predictions from nLON, ConvLSTM, and Koopman AE, respectively.

generalize without feedback correction. As shown in Fig. 5, the predicted frames (Row 2) for both DA and CA models closely match the ground truth (Row 1), even over long prediction horizons. SSIM remains above 0.94 and PSNR above 20 dB across all time steps, confirming high-quality prediction and reconstruction. MSE is low—about 10^{-3} for DA and 10^{-4} for CA. These results demonstrate the generalization ability of the models to generate plausible long-horizon predictions in the latent space.

Fig. 6 presents a comparison of open-loop prediction performance across models. nLON sustains high structural fidelity and low error throughout the prediction horizon, with particularly strong performance in the later frames. In contrast, ConvLSTM predictions degrade rapidly, losing structural detail after only a

Authorized licensed use limited to: TU Delft Library. Downloaded on February 04,2026 at 10:43:57 UTC from IEEE Xplore. Restrictions apply.

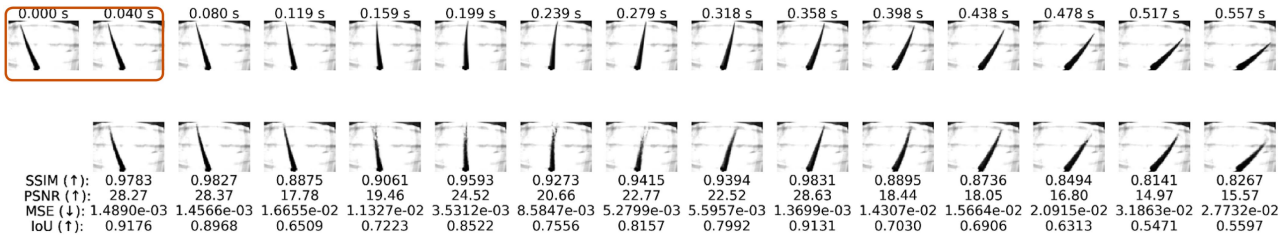
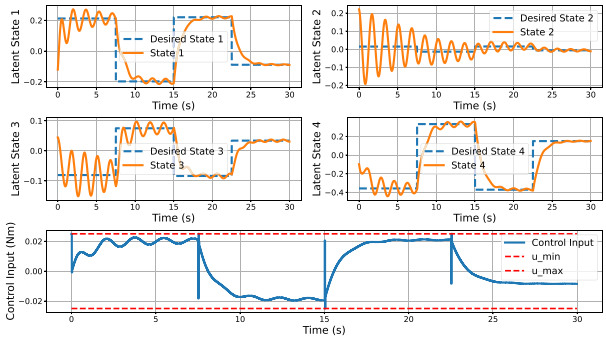
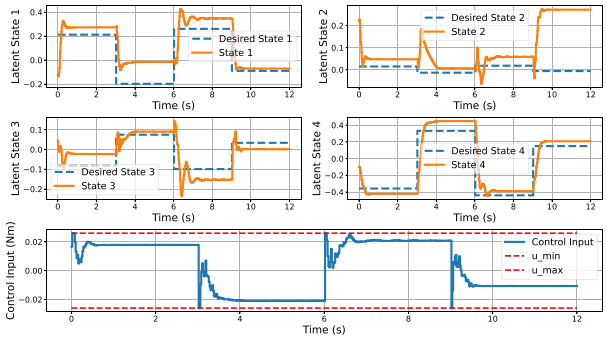


Fig. 7. DA: Visualization of the model's open-loop prediction performance on a test sample for the hardware experiment.

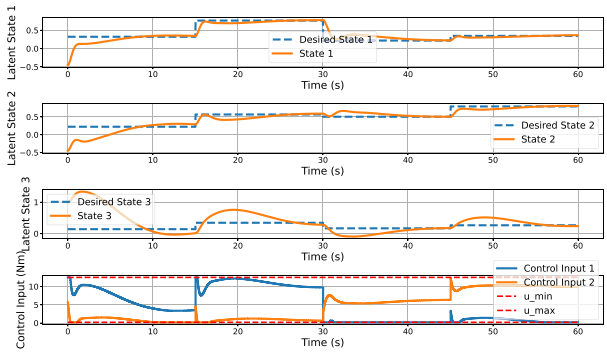


(a) DA with IM strategy.

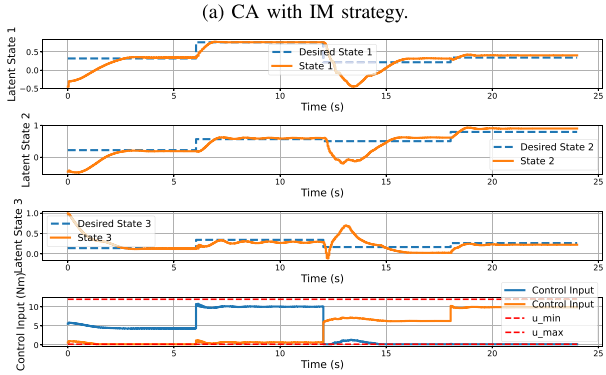


(b) DA with PF strategy.

Fig. 8. Latent-space tracking under two control strategies. (a) Internal-model (IM) strategy in the direct actuation (DA) case: trajectories track desired references with stable performance, and control inputs remain within torque bounds. (b) Pixel-level feedback (PF) in the direct actuation (DA) case: trajectories follow desired references with small offsets from autoencoder errors; control inputs stay within torque limits.



(a) CA with IM strategy.



(b) CA with PF strategy.

Fig. 9. Latent-space tracking under two control strategies. (a) Internal-model (IM) strategy in the cable actuation (CA) case: trajectories follow the references with overshoots in Latent State 3; control inputs remain within limits. (b) Pixel-level feedback (PF) strategy in the cable actuation (CA) case: trajectories closely follow the references with smaller static offsets than in DA; control inputs stay within limits.

few steps, while Koopman AE exhibits a more gradual decline, with drift becoming evident over time.

2) Hardware Experiments: We further validate the DA model on real-world hardware data. Fig. 7 shows two representative test sequences, where the predicted image sequences remain visually consistent with the ground truth. On hardware sequences, the DA model captures the overall deformation and motion trajectory of the tentacle despite increased noise and more complex backgrounds. Compared to simulation, the SSIM and PSNR values are slightly lower. This reflects the challenge of domain shift and higher variability in real-world images. Notably, the prediction does not drift or collapse over the 15-step horizon, showing that the latent dynamics are robust enough to generalize beyond synthetic training.

3) Quantitative Metrics & Trends: We summarize the quantitative results for all scenarios: simulation DA (SIM DA), simulation CA (SIM CA), and hardware DA (HW DA) in Table III. Across all three cases, nLON achieves the highest SSIM, MSE, and IoU on both the rollout average and the final step, indicating superior preservation of structure over long horizons. On SIM DA, the Koopman AE attains slightly higher average PSNR with its linear-control bias—yet nLON delivers better final-step quality. On HW DA, ConvLSTM is competitive in PSNR, suggesting improved perceptual fidelity and robustness in the real system. These experiments on two simulated soft robots and a hardware prototype confirm that the learned linear-oscillator latent model captures the underlying dynamics with high fidelity. In contrast, prior learning-based approaches that focus solely on visual reconstruction or latent regression often accumulate drift over multi-step horizons. All metrics show an overall degradation as the prediction horizon increases. The small local improvements observed at intermediate frames (e.g., 0.10–0.14 s in Fig. 5(a)) are due to image–space factors. Pixel-based metrics are sensitive to slight spatial shifts, while being comparatively tolerant to mild blur, so a sharp but

Authorized licensed use limited to: TU Delft Library. Downloaded on February 04,2026 at 10:43:57 UTC from IEEE Xplore. Restrictions apply.

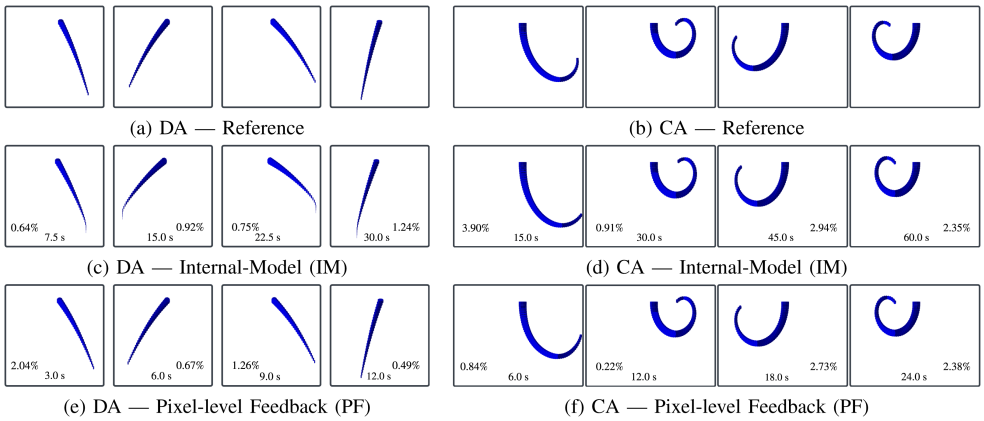


Fig. 10. Reference tracking on the soft robot with the final backbone RMS tracking error is reported as a percentage of the image height. The figure juxtaposes direct actuation (DA, left column) and cable actuation (CA, right column). Each column shows: (a) reference shapes; (b) achieved shapes with the Internal-Model (IM) controller; (c) achieved shapes with pixel-level Feedback (PF).

TABLE III
OPEN-LOOP PREDICTION PERFORMANCE ACROSS: SIMULATION DA AND CA, HARDWARE DA. METRICS ARE REPORTED AVERAGED OVER THE FULL ROLLOUT (14 STEPS) AND AT THE FINAL PREDICTION STEP. "LATENT DIM." DENOTES THE DIMENSION OF THE LEARNED LATENT REPRESENTATION FOR NLON AND KOOPMAN AE; FOR CONVLSTM, IT IS GIVEN AS $h \times w$, THE LATENT FEATURE MAP RESOLUTION

Case	Method	Latent Dim.	Average over rollout				Final step				Trainable parameter
			SSIM↑	PSNR↑	MSE↓	IoU↑	SSIM↑	PSNR↑	MSE↓	IoU↑	
SIM DA	nLON (ours)	4	**0.895±0.029**	21.0±1.12	**0.011±0.003**	**0.428±0.080**	**0.825±0.041**	**17.8±1.42**	**0.017±0.006**	**0.176±0.135**	**436,703**
	ConvLSTM	4×4	0.840±0.035	19.4±0.291	0.015±0.001	0.189±0.021	0.793±0.064	17.5±0.297	0.018±0.001	0.013±0.019	2,220,785
	Koopman AE	4	0.871±0.036	**21.2±0.998**	0.014±0.003	0.396±0.060	0.776±0.043	16.2±0.988	0.025±0.005	0.094±0.355	436,841
SIM CA	nLON (ours)	3	**0.988±0.008**	**33.91±2.68**	**0.001±0.001**	**0.778±0.105**	**0.974±0.017**	**28.4±3.29**	**0.002±0.002**	**0.778±0.105**	**412,121**
	ConvLSTM	4×4	0.947±0.008	28.6±0.651	0.005±0.001	0.675±0.050	0.884±0.010	19.5±0.633	0.011±0.002	0.345±0.065	2,220,817
	Koopman AE	3	0.955±0.006	28.2±0.721	0.004±0.001	0.702±0.031	0.891±0.018	20.1±0.914	0.010±0.002	0.415±0.070	412,211
HW DA	nLON (ours)	4	**0.904±0.047**	20.1±3.17	**0.018±0.011**	**0.664±0.114**	**0.859±0.077**	16.7±6.03	**0.0326±0.019**	**0.510±0.185**	**476,447**
	ConvLSTM	4×4	0.899±0.047	**20.3±2.88**	0.019±0.010	0.655±0.111	0.837±0.079	**17.5±7.40**	0.0329±0.019	0.497±0.209	2,220,785
	Koopman AE	4	0.872±0.032	18.9±1.92	0.020±0.007	0.616±0.083	0.814±0.048	14.7±1.52	0.036±0.010	0.418±0.061	476,777

slightly misaligned prediction can yield a worse score than a blurrier yet better-aligned one. Moreover, postures close to the initial condition occur more frequently in the training data and tend to be reconstructed with higher accuracy. These factors can temporarily reduce pixel error despite the gradual accumulation of geometric drift.

C. Performance of Control Based on the Learned Model

Closed-loop tracking experiments assess the control performance using the learned latent dynamics with internal-model (IM) and pixel-feedback (PF) strategies.

1) Direct Actuation: IM and PF Controllers: The tracking results in Fig. 8(a) illustrate the effectiveness of the proposed IM control in latent space for the DA case. The controller guides the system to track a sequence of desired equilibrium latent states over a 20-second horizon. The control torque is totally computed in latent space and then directly applied to the DA system in the pyelastica simulation. Fig. 10(c) shows the state of the soft robot's motion at several time instances. The close visual agreement between reference and actual configurations confirms that the controller successfully drives the system to track the desired motion, despite some small mismatches that can be found due to the reconstruction loss and simplification of the model.

We apply the PF control loop using real-time visual observations of the soft robot. As shown in Fig. 8(b), the latent states can generally follow the trends of the provided desired references. However, in contrast to the Model-Corrected Feedback controller, this approach exhibits persistent static errors. These static offsets likely result from reconstruction inaccuracies in the autoencoder, which introduce discrepancies between the observed visual state and its latent encoding.

However, despite these latent-level inaccuracies, the closed-loop control performance on the physical robot remains strong. Fig. 10(e) shows that the robot shapes achieved under PF control more closely match the reference trajectories. This indicates that the static errors in latent space do not significantly affect the end-effector behavior, and are primarily attributable to encoder imperfections.

2) Cable Actuation: IM and PF Controllers: Fig. 9(a) shows that the IM controller enables the latent states to reach their desired references, although Latent State 3 exhibits noticeable overshoots. The IM control torque is also applied directly to the pyelastic CA robot, and the results are shown in Fig. 10(d).

Fig. 9(b) shows the latent tracking performance using the PF strategy. Compared to the DA case, the CA model under PF exhibits reduced static errors across all latent dimensions. This improvement likely stems from the stronger reconstruction accuracy (smaller MSE shown in the prediction) of the CA model,

Authorized licensed use limited to: TU Delft Library. Downloaded on February 04,2026 at 10:43:57 UTC from IEEE Xplore. Restrictions apply.

which reduces the discrepancy between visual observations and latent encodings.

Fig. 10(f) shows that the shapes achieved on the robot using PF also align closely with the reference shapes.

V. Conclusion and Future Work

We introduced Neural Linear Oscillator Networks (nLON), a neural architecture cast in a vision-based framework for modeling and control of soft robotic tentacles. Images are encoded into compact position–velocity latents. Unlike methods such as ConvLSTM or Koopman AEs that rely on visual reconstruction or unconstrained latent regression, nLON embeds a linear-oscillator prior directly in the latent space. This preserves passivity and enforces bounded, temporally consistent evolution, yielding markedly more stable multi-step predictions and enabling effective feedback control. Although demonstrated on a simple prototype, the nLON formulation is not limited to planar or single-view setups. Because the latent dynamics are independent of the observation dimensionality, the framework extends naturally to multi-segment and spatial soft robots. Moreover, multi-view or proprioceptive fusion modules can replace the current single-view encoder to mitigate sensitivity to occlusions and lighting. Beyond the modeling, we introduced two feedback-control schemes —including a pixel-space strategy—and evaluated them in simulation. The current controller does not account for physical constraints, and the convolutional encoder adds noticeable latency on hardware. Future work will investigate lightweight perceptual encoders, extend nLON to multi-segment and fully spatial soft robots as well as rigid-body systems, and develop constraint-aware closed-loop control strategies that combine multi-view perception or proprioceptive feedback to improve robustness in more challenging environments.

References

[1] O. Limoyo, B. Chan, F. Marić, B. Wagstaff, A. R. Mahmood, and J. Kelly, "Heteroscedastic uncertainty for robust generative latent dynamics," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6654–6661, Oct. 2020.

[2] 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 Syst. Mag.*, vol. 43, no. 3, pp. 30–65, Jun. 2023.

[3] Z. Chen et al., "Data-driven methods applied to soft robot modeling and control: A review," *IEEE Trans. Automat. Sci. Eng.*, vol. 22, pp. 2241–2256, 2025.

[4] L. Han, K. Peng, W. Chen, and Z. Liu, "A data-driven koopman modeling framework with application to soft robots," *Int. J. Control, Automat. Syst.*, vol. 23, no. 1, pp. 249–261, 2025.

[5] T. Coleman, R. Babuška, J. Kober, and C. Della Santina, "An empirical investigation on variational autoencoder-based dynamic modeling of deformable objects from RGB data," in *Proc. 32nd Mediterranean Conf. Control Automat.*, 2024, pp. 921–928.

[6] S. L. Li et al., "Controlling diverse robots by inferring Jacobian fields with deep networks," *Nature*, vol. 643, pp. 89–95, 2025.

[7] O. Limoyo, T. Ablett, and J. Kelly, "Learning sequential latent variable models from multimodal time series data," in *Proc. Int. Conf. Intell. Auton. Syst.*, 2022, pp. 511–528.

[8] X. He, Y. Choi, W. D. Fries, J. L. Belof, and J.-S. Chen, "gLaSDI: Parametric physics-informed greedy latent space dynamics identification," *J. Comput. Phys.*, vol. 489, 2023, Art. no. 112267.

[9] F. Regazzoni, S. Pagani, M. Salvador, L. Dede', and A. Quarteroni, "Learning the intrinsic dynamics of spatio-temporal processes through latent dynamics networks," *Nature Commun.*, vol. 15, no. 1, 2024, Art. no. 1834.

[10] T. K. Rusch, B. Chamberlain, J. Rowbottom, S. Mishra, and M. Bronstein, "Graph-coupled oscillator networks," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 18888–18909.

[11] T. Nguyen, H. Honda, T. Sano, V. Nguyen, S. Nakamura, and T. M. Nguyen, "From coupled oscillators to graph neural networks: Reducing over-smoothing via a Kuramoto model-based approach," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2024, pp. 2710–2718.

[12] S. Lanthaler, T. K. Rusch, and S. Mishra, "Neural oscillators are universal," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 36, pp. 46786–46806.

[13] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics–informed neural networks: Where we are and what's next," *J. Sci. Comput.*, vol. 92, no. 3, 2022, Art. no. 88.

[14] T. Kapoor, A. Chandra, D. Tartakovsky, H. Wang, A. Núñez, and R. Dollevoet, "Neural oscillators for generalizing parametric PDEs," in *Proc. Symbiosis Deep Learn. Differ. Equ. III*, 2023.

[15] A. Chandra et al., "Neural oscillators for magnetic hysteresis modeling," 2023, *arXiv:2308.12002*.

[16] A. Ceni et al., "Random oscillators network for time series processing," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2024, pp. 4807–4815.

[17] M. Stölzle and C. Della Santina, "Input-to-state stable coupled oscillator networks for closed-form model-based control in latent space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 37, pp. 82010–82059.

[18] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Saddle River, NJ, USA: Prentice Hall, 2002.

[19] Y. Zhang, "A better autoencoder for image: Convolutional autoencoder," in *Proc. ICONIP17-DCEC*, pp. 11–21, 2018. Accessed: Mar. 23, 2017. [Online]. Available: http://users.cecs.anu.edu.au/Tom.Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58.pdf

[20] P. Ramachandran, B. Zoph, and Q. V. Le, "Swish: A self-gated activation function," 2017, *arXiv:1710.05941*.

[21] A. Vahdat and J. Kautz, "Nvae: A deep hierarchical variational autoencoder," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 19667–19679.

[22] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, pp. 6572–6583, 2018.

[23] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 802–810, 2015.

[24] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3560–3569.

[25] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32.

[26] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Commun.*, vol. 9, no. 1, 2018, Art. no. 4950.

[27] E. Yeung, S. Kundu, and N. Hodas, "Learning deep neural network representations for Koopman operators of nonlinear dynamical systems," in *Proc. Amer. Control Conf.*, 2019, pp. 4832–4839.

[28] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*. Oxford, U.K.: Clarendon Press, 1995.

[29] M. Gazzola, L. Dudte, A. McCormick, and L. Mahadevan, "Forward and inverse problems in the mechanics of soft filaments," *Roy. Soc. Open Sci.*, vol. 5, no. 6, 2018, Art. no. 171628, doi: 10.1098/rsos.171628.

[30] H.-S. Chang, U. Halder, C.-H. Shih, N. Naughton, M. Gazzola, and P. G. Mehta, "Energy-shaping control of a muscular octopus arm moving in three dimensions," *Proc. Roy. Soc. A*, vol. 479, no. 2270, 2023, Art. no. 20220593.

Authorized licensed use limited to: TU Delft Library. Downloaded on February 04,2026 at 10:43:57 UTC from IEEE Xplore. Restrictions apply.